EMC® Avamar® 6.1
Management Console Command Line Interface (MCCLI)

Programmer Guide

EMC²

CONTENTS

TABLES

	Title	Page

PREFACE

As part of an effort to improve its product lines, EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your EMC representative if a product does not function properly or does not function as described in this document.

Note: This document was accurate at publication time. New versions of this document might be released on the EMC online support website. Check the EMC online support website to ensure that you are using the latest version of this document.

Purpose

This document describes how to install, configure, and use the Avamar Management Console Command Line Interface (MCCLI) Java client software application.

Audience

This document is intended for system administrators who are responsible for installing software and maintaining servers and clients on a network. This document assumes that the reader is familiar with the Avamar Administrator graphical management console as documented in the *EMC Avamar Administration Guide*, and does not generally repeat information in that document.

Revision history

The following table presents the revision history of this document.

Table 1 Revision history (page 1 of 2)

Revision	Date	Description
05	May 2, 2013	Updated the following topics to note REPLICATION domain limitation: • "client add" on page 57 • "client backup-dataset" on page 65 • "client backup-target" on page 68 • "client delete" on page 70 • "client edit" on page 71 • "client invite" on page 75 • "client move" on page 77 • "client retire" on page 79
04	April 17, 2013	Updated the following topics: • "Version compatibility" on page 18. • "schedule add" on page 161 and "schedule edit" on page 165 to deprecate --hours in favor of --time

Table 1 Revision history (page 2 of 2)

Revision	Date	Description
03	July 31, 2012	Updated "Where to get help" on page 13.
A02	June 15, 2012	Updated "client add" on page 57.
A01	April 25, 2012	First release of Avamar 6.1.

Related documentation

The following EMC publications provide additional information:

◆ *EMC Avamar Administration Guide*

◆ *EMC Avamar Operational Best Practices*

◆ *EMC Avamar and Data Domain Integration Guide*

◆ *EMC Avamar Release Notes*

Conventions used in this document

EMC uses the following conventions for special notices:

⚠**DANGER**

DANGER indicates a hazardous situation which, if not avoided, will result in death or serious injury.

⚠**WARNING**

WARNING indicates a hazardous situation which, if not avoided, could result in death or serious injury.

⚠**CAUTION**

CAUTION, used with the safety alert symbol, indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE

NOTICE is used to address practices not related to personal injury.

Note: A note presents information that is important, but not hazard-related.

IMPORTANT

An important notice contains information essential to software or hardware operation.

Typographical conventions

EMC uses the following type style conventions in this document:

Normal	Used in running (nonprocedural) text for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • Names of resources, attributes, pools, Boolean expressions, buttons, DQL statements, keywords, clauses, environment variables, functions, and utilities • URLs, pathnames, filenames, folder names, computer names, links, groups, service keys, file systems, and notifications
Bold	Used in running (nonprocedural) text for names of commands, daemons, options, programs, processes, services, applications, utilities, kernels, notifications, system calls, and man pages Used in procedures for: • Names of interface elements, such as names of windows, dialog boxes, buttons, fields, and menus • What the user specifically selects, clicks, presses, or types
Italic	Used in all text (including procedures) for: • Full titles of publications referenced in text • Emphasis, for example, a new term • Variables
`Courier`	Used for: • System output, such as an error message or script • URLs, complete paths, filenames, prompts, and syntax when shown outside of running text
`Courier bold`	Used for specific user input, such as commands
`Courier italic`	Used in procedures for: • Variables on the command line • User input variables
‹ ›	Angle brackets enclose parameter or variable values supplied by the user
[ ]	Square brackets enclose optional values
|	Vertical bar indicates alternate selections — the bar means "or"
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

Where to get help

The Avamar support page provides access to licensing information, product documentation, advisories, and downloads, as well as how-to and troubleshooting information. This information may enable you to resolve a product issue before you contact EMC Customer Service.

To access the Avamar support page:

1. Go to https://support.EMC.com/products.

2. Type a product name in the **Find a Product** box.

3. Select the product from the list that appears.

4. Click the arrow next to the **Find a Product** box.

5. (Optional) Add the product to the **My Products** list by clicking **Add to my products** in the top right corner of the **Support by Product** page.

Documentation

The Avamar product documentation provides a comprehensive set of feature overview, operational task, and technical reference information. Review the following documents in addition to product administration and user guides:

◆ Release notes provide an overview of new features and known limitations for a release.

◆ Technical notes provide technical details about specific product features, including step-by-step tasks, where necessary.

◆ White papers provide an in-depth technical perspective of a product or products as applied to critical business issues or requirements.

Knowledgebase

The EMC Knowledgebase contains applicable solutions that you can search for either by solution number (for example, esgxxxxxx) or by keyword.

To search the EMC Knowledgebase:

1. Click the **Search** link at the top of the page.

2. Type either the solution number or keywords in the search box.

3. (Optional) Limit the search to specific products by typing a product name in the **Scope by product** box and then selecting the product from the list that appears.

4. Select **Knowledgebase** from the **Scope by resource** list.

5. (Optional) Specify advanced options by clicking **Advanced options** and specifying values in the available fields.

6. Click the search button.

Live chat

To engage EMC Customer Service by using live interactive chat, click Join Live Chat on the Service Center panel of the Avamar support page.

Service Requests

For in-depth help from EMC Customer Service, submit a service request by clicking Create Service Requests on the Service Center panel of the Avamar support page.

Note: To open a service request, you must have a valid support agreement. Contact your EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

To review an open service request, click the Service Center link on the Service Center panel, and then click View and manage service requests.

Facilitating support

EMC recommends that you enable ConnectEMC and Email Home on all Avamar systems:

◆ ConnectEMC automatically generates service requests for high priority events.

◆ Email Home emails configuration, capacity, and general system information to EMC Customer Service.

Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to:

BSGDocumentation@emc.com

Please include the following information:

- ◆ Product name and version

- ◆ Document name, part number, and revision (for example, A01)

- ◆ Page numbers

- ◆ Other details that will help us address the documentation issue

CHAPTER 1
Introduction

The EMC® Avamar® Management Console Command Line Interface (MCCLI) is a Java software application that provides command line access to Avamar Administrator functions. The following topics introduce system requirements and capabilities and limitations of the MCCLI:

System requirements

To connect to the Avamar server using MCCLI, you must have:

◆ A valid Avamar Administrator ID and password

◆ Network access to an operational Avamar server

Additionally, the computer running MCCLI software must meet the minimum requirements listed in the following table.

Table 2 **Minimum requirements for MCCLI software**

Requirement	Minimum
Operating system	Red Hat Enterprise Linux Release 4 (64-bit)
RAM	256 MB
Hard drive space	60 MB permanent hard drive space
Network interface	10BaseT or higher, configured with the latest drivers for the platform
Java Runtime Environment	JRE 1.6

Capabilities and limitations

The following topics discuss the capabilities and limitations of the MCCLI.

Version compatibility

The MCCLI software must be the same version as the server MCS..

Hierarchical management

Hierarchical management is not supported in this release of MCCLI. The root domain is assumed for all user IDs included on the command line. This means that a user must have an account in the root domain to use the MCCLI, and domain administrators cannot log on.

Relative path filenames

The use of filenames containing relative paths is not supported. Filenames are specified as part of the client bulk validate and load commands.

Client bulk validate/load file format

The **client load-bulk** command, discussed on page 76, and **client validate-bulk** command, discussed on page 76, currently only support clients definition input files in XML format. You cannot use a CSV file with these commands.

Delayed updates for MCCLI scheduler changes

When you use the MCCLI to suspend or resume the scheduler, the change takes effect immediately. However, the change may not appear in Avamar Administrator for approximately 15 seconds.

Deleting clients with backups pending

As with the Avamar Administrator, you cannot delete a client until it is idle without any backups in progress or it is in the wait queue.

Overall performance

MCCLI commands typically require one to two seconds to complete. This elapsed time is primarily because the Java Virtual Machine (JVM) must be started to run any MCCLI command.

Maximum number of clients

As with Avamar Administrator, the MCCLI is suitable for systems with a maximum of 5,000 clients.

Java stack trace

The **mccli** command might display a Java stack trace if the command fails. This is due to unhandled exceptions.

Generic mccli command failed message

Some errors and failure conditions are not associated with specific event codes. As a result, when these errors and failure conditions occur, the generic event code for an **mccli** command failed appears. For example, missing arguments or failed connection from the MCCLI to the MCS generates the same 23998 event (**mccli** command failed).

CHAPTER 2
Installation and Configuration

This chapter describes how to configure the Avamar Management Console Command Line Interface (MCCLI). To properly install and configure the MCCLI, perform the following tasks in order:

Install the Java Runtime Environment

The MCCLI requires Java Runtime Environment (JRE) 1.6 Update 12 or later. You can download the JRE directly from the Avamar server.

To install the Java Runtime Environment:

1. Log in to the computer where the software will be installed as root.

2. Open a web browser and go to the following URL:

    `http://AVAMARSERVER`

    where AVAMARSERVER is the Avamar server network hostname (as defined in DNS) or IP address.

    You are automatically redirected to the Avamar secure web server.

    Depending on browser security settings, a security alert dialog box might appear.

3. If a security alert dialog box appears, click **Yes** or **OK** to allow redirection to the Avamar secure web server.

    The Secure Log On page appears.

4. Scroll down until the **Documents and Downloads** hyperlink appears.

5. Click **Documents and Downloads**.

    The Downloads and Documentation page appears.

6. Click **Downloads › Red Hat Enterprise Linux 3** and download **jre-6u12-linux-amd64.rpm** to a temporary install directory on the computer, such as /tmp.

7. Run the installation program from a command shell by typing:

    ```
    cd /tmp
    rpm -ivh jre-6u12-linux-amd64.rpm
    ```

8. Follow the onscreen instructions to complete the JRE installation.

Install the MCCLI RPM package

The MCCLI RPM package contains all necessary files for the MCCLI Java client application.

To install the MCCLI RPM package:

1. Obtain the dpnmccli-VERSION.sles11_64.x86_64.rpm package, where VERSION is the version of the RPM package to install.

2. Open a command shell and log in to the system as root.

3. Copy the RPM to a temporary folder.

4. Type:

   `rpm -ivh dpnmccli-VERSION.sles11_64.x86_64.rpm`

   where VERSION is the version of the RPM package to install.

   The following information appears in the command shell:

   ```
   rpm -ivh dpnmccli-VERSION.sles11_64.x86_64.rpm
   Preparing... ####################################### [100%]
   1:dpnmccli ######################################### [100%]
   Please run /usr/local/avamar/bin/avsetup_mccli to configure MCLI
   ```

5. Follow the instructions in the command shell to complete the installation.

Run avsetup_mccli

The **avsetup_mccli** utility configures the MCCLI.

By default the MCCLI RPM installs essential files to the locations in the following table.

Table 3 MCCLI essential files

Location	Essential files
/usr/local/avamar/bin	Binary and executables
/usr/local/avamar/doc	Documentation and report templates
/usr/local/avamar/lib	Resource libraries and initial configuration files

The RPM documentation provides instructions on installing the MCCLI application to a folder other than the default /usr/local/avamar location.

The default paths in the following table are used by the MCCLI during command invocation.

Table 4 MCCLI default paths

Default path	Description
/usr/java/jre1.6.0_12	Location of JRE installation
/usr/local/avamar	Location of MCCLI installation (this is also known as $AVAMAR_ROOT)
~/.avamardata/var	Location of user command invocation data and logs (this is also known as $USER_ROOT)

You can modify these path assignments at any time by rerunning the **avsetup_mccli** utility.

Additionally, **avsetup_mccli** also prompts you to specify values for all **mccli** global options, as discussed in "Global options" on page 34. When you specify these values during the interactive **avsetup_mccli** session, it automatically updates the mcclimcs.xml default options file, which is discussed in "mcclimcs.xml default options file" on page 25.

To use **avsetup_mccli** to configure the MCCLI:

1. Open a command shell and log in to the system as root.

2. Launch **avsetup_mccli** by typing:

   ```
   avsetup_mccli
   ```

   The following information appears in the command shell:

   ```
   setting linux default
   Enter the location of your JRE 1.5 installation
   [/usr/java/jre1.6.0_12]:
   ```

3. Type the JRE installation folder and press **Enter**.

   The following information appears in the command shell:

   ```
   Enter the root folder of your Avamar installation
   [/usr/local/avamar]:
   ```

4. Type the top-level (root) installation folder for the Avamar software and press **Enter**.

   The following information appears in the command shell:

   ```
   Enter the user data folder of your Avamar installation
   [~/.avamardata/var]:
   ```

5. Type the folder where user data is stored and press **Enter**.

   The following information appears in the command shell:

   ```
   Configuring default local mcsprofile in
   /usr/local/avamar/lib/mcclimcs.xml
   Enter default mcs host name (mcsaddr) [avamar-1.example.com]:
   ```

6. Type the Avamar server IP address or hostname as defined in corporate DNS and press **Enter**.

   The following information appears in the command shell:

   ```
   Enter default mcs port number on avamar-1.example.com (mcsport)
   [7778]:
   ```

7. Press **Enter** to accept the default data port assignment.

   The following information appears in the command shell:

   ```
   Enter default userid on avamar-1.example.com (mcsuserid) [MCUser]:
   ```

8. Type an existing Avamar administrative user account and press **Enter**.

   The following information appears in the command shell:

   ```
   Enter password for MCUser (mcspasswd) [MCUser1]:
   ```

9. Type the Avamar administrative user account password and press **Enter**.

    The following information appears in the command shell:

    ```
    Avamar CLI has been configured correctly
    Type mccli command to use it
    ```

Additional configuration

This topic describes the preferences and options files used by the MCCLI application.

mccli.xml preferences file

The mccli.xml preferences file contains the parameters that you can edit for the MCCLI application.

The default version of mccli.xml is located in $AVAMAR_ROOT/lib. Each time the MCCLI application is run, $USER_ROOT/.avamardata/var/mc/cli_data/prefs is examined to determine if a working copy of mccli.xml is present. If mccli.xml is not present in $USER_ROOT/.avamardata/var/mc/cli_data/prefs, then the default copy of mccli.xml is copied to that location from $AVAMAR_ROOT/lib.

When any MCCLI command is invoked, $USER_ROOT/.avamardata/var/mc/cli_data/prefs/mccli.xml is read, and those settings are used for that command session.

The mccli.xml file contains the two parameters in the following table.

Table 5 mccli.xml file parameters

Parameter	Description
event_monitor_display_limit	Sets the maximum number of events to retrieve
mcs_config_file	Stores the location of the mcclimcs.xml default options file

mcclimcs.xml default options file

The mcclimcs.xml is an XML file that stores custom **mccli** command parameters and profile settings that are used when you invoke any **mccli** command.

Default command parameters

The mcclimcs.xml preferences file can be used to set a default value for any **mccli** command parameter. Any default values set in this file are used unless another value is explicitly supplied on the command line. Additionally, these default values are global, meaning that they are used by all profiles.

Profiles

Each profile is an element in the XML document and is distinguishable by the mcsprofile attribute, which identifies the name of the profile. Each profile contains a list of default options to use with the MCS specified for that profile.

You can designate one profile as the default profile. This default profile is used if no MCS information is specified on the command line global options, which are discussed in "Global options" on page 34. Otherwise, the profile name of the MCS is all that is required on the command line, and the remainder of the options are read from the configuration file.

One or all of the options can be specified on the command line to override entries in the mcclimcs.xml file.

> **NOTICE**

If the server hostname or data port assignment are changed for any reason (for example, after running the **resite** utility), or the user account name or password used to run **mccli** commands is changed for any reason, you must manually update the corresponding settings in the mcclimcs.xml preferences file to account for those changes.

Behavior

The default version of mcclimcs.xml is located in $AVAMAR_ROOT/lib. Each time the MCCLI application is run, $USER_ROOT/.avamardata/var/mc/cli_data/prefs is examined to determine if a working copy of mcclimcs.xml is present.

If $USER_ROOT/.avamardata/var/mc/cli_data/prefs/mcclimcs.xml is not present, then the default copy of mcclimcs.xml is copied to that location from $AVAMAR_ROOT/lib.

The following is a listing of the default mcclimcs.xml default options file:

```
<!-- Administrator Server Profiles -->

   <!-- default : name of default Administrator Server profile -->
   <!-- mcsprofile : Administrator Server profile name -->
   <!-- mcsaddr : network name or IP of Administrator Server node -->
   <!-- mcsport : port to contact Administrator Server -->
   <!-- mcsuserid : account on Administrator Server -->
   <!-- mcspasswd : password -->

   <MCSConfig default="local">
   <Defaults>
      <Commands>
         <!-- Add Resource, Command, Options, Option nodes to match
-->
         <!-- the hierarchy in the mcclisyntax.xml file -->
         <!-- use the Value attribute to specify the default value -->
         <!--
         <Resource Name="ResourceName">
            <Command Name="CommandName">
               <Options>
                  <Option Name="OptionName" Value="OptionValue" />
               </Options>
            </Command>
         </Resource>
         -->
         <Resource Name="activity">
            <Command Name="show">
               <Options>
                  <Option Name="active" Value="true" />
               </Options>
            </Command>
         </Resource>
         <Resource Name="client">
            <Command Name="add">
               <Options>
                  <Option Name="enabled" Value="true" />
                  <Option Name="pageport" Value="29123" />
               </Options>
            </Command>
         </Resource>
      </Commands>
   </Defaults>
   <MCS
      mcsprofile="local"
      mcsaddr="avamar-1.example.com"
      mcsport="7778"
      mcsuserid="root"
      mcspasswd="MyPassword"
   />
   <!-- add more profiles if needed here -->
   <!-- and set default to select default -->
   </MCSConfig>
```

Practical examples

Consider the following activity resource setting:

```
<Resource Name="activity">
   <Command Name="show">
      <Options>
         <Option Name="active" Value="true" />
      </Options>
   </Command>
   </Resource>
```

This setting constrains the **mccli activity show** command, which is discussed on page 37, to only show active jobs, as if the **--active=true** option was supplied on the command line.

Consider the following client resource settings:

```
<Resource Name="client">
   <Command Name="add">
      <Options>
         <Option Name="enabled" Value="true" />
         <Option Name="pageport" Value="29123" />
      </Options>
   </Command>
   </Resource>
```

These settings affect the **mccli client add** command, which is discussed on page 57, so that any new client is enabled and its page data port is set to 29123 as if the **--enabled=true** and **--pageport=29123** options were supplied on the command line.

Remote clients must run mccli as root

If you have installed MCCLI on a remote client, all mccli commands invoked using the operatng system root privileges.

Furthermore, if additional security is required, users should remove any passwords stored in mcclimcs.xml and instead supply a **--mcspasswd=PASSWORD** option with each **mccli** command.

CHAPTER 3
Command Reference

The following topics describe all commands and functions provided by the Avamar Management Console Command Line Interface (MCCLI) application:

General programming notes

The **mccli** is a shell script wrapper that invokes the MCCLI Java application. It automatically sets various environment arguments that are required to invoke a Java application, thereby simplifying use of the MCCLI java application.

Command line elements

Each **mccli** command comprises:

◆ A resource, discussed in "Resources" on page 33.

◆ A command, discussed in "Commands" on page 34.

◆ Global options for connecting to the MCS, discussed in "Global options" on page 34.

◆ Command options, which are specific to the resource and command combination used on a specific **mccli** command line.

◆ Display options, discussed in "Display options" on page 34, that control how the output is formatted.

You must specify these command line elements in this order or the **mccli** command fails.

Data types

The following tables lists the data types used by **mccli** commands and discusses how to correctly specify these kinds of values on the **mccli** command line.

Table 6 Data types for **mccli** commands

Data type	Description
BOOLEAN	BOOLEAN values always express a binary condition: true or false. Valid values are the case-insensitive words true or false.
INTEGER	INTEGER values are always whole numbers.
STRING	STRING values contain plain text. **Note:** If STRING contains spaces, it must be enclosed in either single or double quotes. Some **mccli** STRING values accept regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.

Default values

If a command line argument has a default value, it is shown in parentheses. For example, the following BOOLEAN value defaults to the false condition:

```
--verbose=BOOLEAN(false)
```

Similarly, the following STRING value defaults to the root domain denoted by the slash character (/):

```
--domain=STRING(/)
```

Optional and required arguments

Command synopses use the following convention to convey whether a particular argument is required for that command:

◆ Optional arguments are enclosed by square brackets.

◆ Required arguments are not.

For example, consider the following command line synopsis:

```
mccli client activate [GLOBAL-OPTIONS] [--domain=STRING(/)]
    --name=STRING
```

In the context of this command, **mccli client activate**, the presence of square brackets indicates that --domain=STRING is an optional argument, while the absence of square brackets indicates that --name=STRING is a required argument.

Pattern matching

Some **mccli** STRING values accept regular expression (regex) pattern matching operators, also known as wildcards, with the following capabilities and limitations:

Table 7 Pattern matching operator capabilities and limitations

Operator	Description
Asterisk (*)	Matches zero or more occurrences of any character until the first folder delimiter character (for example, slash on UNIX platforms and backslash on Windows platforms) is encountered. This effectively limits the pattern matching to a single client folder. For example, /usr/* matches the contents of /usr but not the contents of /usr/bin or /usr/local.
Double asterisk (**)	Matches zero or more occurrences of any character. This correlates to conventional single asterisk regex behavior. For example, /usr/** matches the entire /usr folder structure, no matter how many subfolder levels are encountered.
Question mark (?)	Matches one occurrence of any character. Conventional regex behavior.
Plus sign (+)	Unlike regex, the plus sign is not processed as a glob operator; the plus sign only matches a single occurrence of the plus sign.
Forward slash (/)	Patterns beginning with forward slash (/) are assumed to be absolute path designations for a single folder. Recursive processing of subfolders is disabled, and that folder name is not matched anywhere else.
[range of values]	Characters enclosed in square brackets and separated by a single hyphen (-) are interpreted as a range of values. This is conventional regex behavior. For example: • [0-9] matches any single numeric character • [a-z] matches any single lowercase alpha character
Pound sign (#)	In most cases, you can define multiple matching patterns in a text file and pass that file into the utility. This is generally easier than specifying multiple matches directly on the command line. However, when using a text file to pass in pattern matches, the pound sign (#) is interpreted as a comment if it appears at the beginning of a matching pattern. This causes that entire pattern matching entry to be ignored.

Output description

The following topics discuss the messages that are returned when an **mccli** command completes either successfully or unsuccessfully.

Successful completion

When an **mccli** command completes successfully, the following message is returned:

```
0,23000,CLI command completed successfully.
```

The first line is divided into three separate comma-delimited elements:

◆ The first element is the return code, which is zero because the command successfully completed.

◆ The second element is the event code, 23000.

◆ The third element is the event code short description, "CLI command completed successfully".

Depending on the command issued, additional information might be displayed on subsequent lines. For example, consider this sample output from the **mccli client show --domain=clients** command:

```
0,23000,CLI command completed successfully.
command completed successfully
Name       Domain
--------   --------
MyClient   /clients
```

Unsuccessful completion

When an **mccli** command does not complete successfully, the output message is in the same format, but the return code is one (1) and the event code describes the error condition. For example:

```
1,22288,Dataset does not exist.
```

Global command line syntax and behavior

This topic describes the proper format and syntax for all **mccli** commands.

Synopsis

```
mccli RESOURCE COMMAND [GLOBAL-OPTIONS] [COMMAND-OPTIONS]
    [DISPLAY-OPTIONS]
```

Resources

Each **mccli** command line must contain one and only one of the resources in the following table.

Table 8 Resources for the **mccli** command

Resource name	Description	Additional information
activity	Activity management	"activity" on page 36
agent	Client agent management	"agent" on page 39
backup	Backup management	"backup" on page 39
checkpoint	Checkpoint management	"checkpoint" on page 51
client	Client management	"client" on page 55
dataset	Dataset management	"dataset" on page 91
dd	Data Domain system configuration and monitoring	"dd" on page 108
domain	Domain management	"domain" on page 115
dump	Strictly reserved for EMC internal use only	"dump" on page 119
event	Event code information	"event" on page 120
group	Group management	"group" on page 128
help	Help information	"help" on page 147
mcs	Administrator server management	"mcs" on page 147
plugin	Client plug-in management	"plugin" on page 152
retention	Retention policy management	"retention" on page 153
schedule	Schedule management	"schedule" on page 161
server	Server monitoring	"server" on page 169
snapup	Snapup management resource that has been deprecated in favor of the backup management resource	"snapup" on page 175
user	User management	"user" on page 182
vcenter	vCenter management	"vcenter" on page 192
version	Version information	"version" on page 195

Commands

The list of possible commands is specific to each resource class. The individual command listings elsewhere in this chapter provide a list of commands available for each resource class.

Global options

Each **mccli** command line can contain one or more of the following options, which are global in nature, meaning that they can be used with any resource class or command:

```
[--mcsprofile=STRING] | {[--mcsaddr=STRING] [--mcspasswd=STRING]
[--mcsport=INTEGER] [--mcsuserid=STRING]}
```

Specify these global options on the command line before you specify command options.

The following table provides details on each global option.

Table 9 Global options for **mccli**

Global option	Description
--mcsprofile=STRING	Specifies the MCS profile name. If you supply this option, then settings stored in this profile are used, and other global options that you supply on the command line are ignored.
--mcsaddr=STRING	Specifies the MCS network name or IP address.
--mcspasswd=STRING	Specifies the --mcsuserid account password.
--mcsport=INTEGER	Specifies the data port used to contact MCS.
--mcsuserid=STRING	Specifies the Avamar user account that is used to run the **mccli** command.

Typically, global options are persistently stored and read from the mccli.xml preferences file, which is discussed in "mccli.xml preferences file" on page 25. You only supply the global options on an **mccli** command line in cases where the persistent settings must be temporarily overridden.

Command options

The list of possible command options is specific to each combination of resource class and command. The individual command listings elsewhere in this chapter provide a list of command options available for each resource-command combination.

Display options

Specify the following display options after the command options.

Table 10 Display options for **mccli**

Display option	Description
--normalize	Produces output in a format that better supports parsing and comparison. This option affects any command that returns a date or time, or filesystem information such as capacity or file sizes.
--xml	Formats output as XML. This is useful for parsing output as part of a script.

The following table shows examples of returned values expressed in both default and normalized formats:

Table 11 Returned values in default and normalized formats

Value	Default format	Normalized format
Absolute time	Expressed as a conventional date and timestamp with the local timezone. For example: 2011-09-12 11:00:17 PDT	Expressed as an integer representing the UNIX precision time format numerical (UTC milliseconds from the epoch). For example: 1189620017000
Elapsed time	Expressed as the number of days, hours, minutes, and seconds that have elapsed. For example: 12 days 17h:30m	Expressed as an integer representing the total number of milliseconds that have elapsed. For example: 1099814000
Capacity	Expressed in bytes, MB, GB, or TB, whichever is most correct. For example: 1.3 TB	Expressed as an integer representing the total number of bytes. For example: 1464957140992

Return codes

Each **mccli** command returns one of the following codes:

0   Command succeeded.

1   Command failed.

Global event codes

These events codes are global, which means that they can be returned for any **mccli** command. However, in the event of an error, many **mccli** commands return other event codes that more fully describe the specific error condition encountered. Those event codes are documented with each **mccli** command.

22601   Server inactive.

23000   CLI command successfully completed.

23001   Arguments required by CLI command are either missing or empty.

23993   Attempt to read or write a file has failed.

23995   Invalid option specified on the CLI.

23996   Failed to connect to the administrator server.

23997   Conflicting arguments specified on the command line.

23998   CLI command failed.

23999   Unexpected command failure.

activity

The **mccli activity** resource is used to cancel or show backup, restore, and validation activities that are currently active. Available commands are:

- "activity cancel" on page 36
- "activity get-log" on page 37
- "activity show" on page 37

activity cancel

The **mccli activity cancel** command cancels a backup, restore, or validation activity. If the activity has already completed, then the command returns an event code indicating that the activity ID is invalid, along with information indicating that the job has already completed.

Synopsis

```
mccli activity cancel [GLOBAL-OPTIONS] --id=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli activity cancel** command.

Table 12 Command options for **mccli activity cancel**

Option	Description
--id=STRING	Specifies which activity to cancel. STRING must be a valid activity ID. This argument is required. Use **mccli activity show**, discussed on page 37, to return a list of valid activity IDs, or use the activity ID returned by the backup, restore, or validation command.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli activity cancel** command.

22205	Backup cancelled via console.
23010	Invalid ID specified on the CLI.
23023	Activity already completed.

activity get-log

The **mccli activity get-log** command displays the contents of the session log file for an activity. Use the global --xml option to display escaped log file content.

Synopsis

mccli activity get-log [GLOBAL-OPTIONS] --id=STRING [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command option is available for the **mccli activity get-log** command.

Table 13 Command options for **mccli activity get-log**

Option	Description
--id=STRING	Specifies the activity ID, which is also called the session ID, for which to get log files, where STRING is an activity ID as returned by **mccli activity show**, discussed on page 37, or any MCCLI issued backup or restore job.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event code is associated with the **mccli activity get-log** command.

23010   Invalid ID specified on the CLI.

activity show

The **mccli activity show** command lists backup, restore, and validation activities with summary information, or detailed information for a specific activity. If you are viewing summary information for multiple activities, you can filter the information on an individual domain or client basis.

Synopsis

mccli activity show [GLOBAL-OPTIONS] [--active=Boolean(false)]
   [--completed=Boolean(false)] [--domain=STRING] [--id=STRING]
   --name=STRING [--queued=Boolean(false)] [--source=STRING]
   [--verbose=Boolean(false)] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli activity show** command.

Table 14 Command options for **mccli activity show**

Option	Description
`--active=Boolean(false)`	If true, then only currently running activities are returned.
`--completed=Boolean(false)`	If true, then only completed activities are returned.
`--domain=STRING`	Specifies the Avamar server domain with the client specified by the --name argument. If you supply this option and you do not supply --name, then all activities in that domain are shown.
`--id=STRING`	Specifies which activities to show. STRING must be a valid activity ID. Multiple --id options can be specified on the same command line.
`--name=STRING`	Specifies the client for which activities are shown. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
`--queued=Boolean(false)`	If true, then only queued activities are returned.
`--source=STRING`	Specifies a source (storage location), either: • avamar • dd If you specify avamar, then the command shows activities for the Avamar server. If you specify dd, then the command shows activities for all configured Data Domain systems.
`--verbose=Boolean(false)`	If true, then detailed activity information is returned. If false or not supplied, then summary information is returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

agent

The **mccli agent** resource comprises a single command, **agent show**.

agent show

The **agent show** command shows summary properties for all client agents.

Synopsis

```
mccli agent show [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

backup

The **mccli backup** resource is used to restore folders and files to clients, as well as to manage backups on the Avamar server. Available commands are:

- "backup delete" on page 39
- "backup edit" on page 41
- "backup restore" on page 42
- "backup show" on page 47
- "backup validate" on page 49

backup delete

The **mccli backup delete** command permanently deletes a backup from the server.

Synopsis

```
mccli backup delete [--created=STRING] [--domain=STRING(/)]
    [--force=Boolean(false)] [--labelNum=INTEGER] [--name=STRING]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli backup delete** command.

Table 15 Command options for **mccli backup delete**

Option	Description
`--created=STRING`	Specifies the date the backup was created. STRING must be in the format of YYYY-MM-DD, where YYYY, MM, and DD are calendar year, month, and date, respectively.
`[--domain=STRING(/)]`	Specifies the Avamar server domain that contains the client specified by the --name argument.
`--force=Boolean(false)`	By default, if you attempt to delete a backup that has more than one retention type assigned to it, then a warning is issued and the backup is not deleted. This is intended to prevent inadvertent deletion of a single backup that could remove more than one level of historical backups (daily, weekly, monthly, or yearly) from the server. If true, then the checking is disabled and the backup is deleted regardless of the number of retention types assigned to it.
`--labelNum=INTEGER`	Specifies the label number of the backup to delete.
`--name=STRING`	Specifies the client from which the backup was originally taken. This argument is required. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli backup delete** command.

22236	Client does not exist.
22552	Backup does not exist.
22556	Changed backup retention.
22558	Multiple retention tags exist.

backup edit

The **mccli backup edit** command enables you to edit the backup expiration date by one of the following methods:

◆ Directly specifying a new expiration date

◆ Adding additional days (d), weeks (w), months (m), or years to the existing backup expiration date

◆ Specifying that the backup should never expire

Synopsis

```
mccli backup edit [--created=STRING] [--domain=STRING(/)]
   [--expiration=STRING | --retention=STRING] [--force=Boolean(false)]
   [--labelNum=INTEGER] [--name=STRING]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli backup edit** command.

Table 16 Command options for **mccli backup edit** (page 1 of 2)

Option	Description
--created=STRING	Specifies the date the backup was created. STRING must be in the format of YYYY-MM-DD, where YYYY, MM, and DD are calendar year, month, and date, respectively.
[--domain=STRING(/)]	Specifies the Avamar server domain that contains the client specified by the --name argument.
--expiration=STRING	Specifies a new expiration date. STRING must be in one of the following formats: YYYY-MM-DD | +nn{D|W|M|Y} | NO_EXPIRATION where: • YYYY-MM-DD is a specific year (YYYY), month (MM), and day (DD). • +nn{D|W|M|Y is the number (nn) of additional days (d), weeks (w), months (m), or years that should be added to the existing backup expiration date. • NO_EXPIRATION specifies that the backup should never expire. The --expiration and --retention options are mutually exclusive.
--force=Boolean(false)	By default, if you attempt to edit the retention type for a backup that has more than one retention type assigned to it, a warning is issued and the retention type is not edited. If true, the checking is disabled and the backup is edited regardless of the number of retention types assigned to it.

Table 16 Command options for **mccli backup edit** (page 2 of 2)

Option	Description
`--labelNum=INTEGER`	Specifies the label number of the backup to change.
`--name=STRING`	Specifies the client from which the backup was originally taken. This argument is required. If you supply a fully qualified client name (for example, /clients/MyClient), the --domain argument is ignored.
`--retention=STRING`	Specifies the retention types to assign to the backup. STRING must be one or more of the following values: • D or daily • W or weekly • M or monthly • Y or yearly • N or none Both short form and long form retention type values are allowed and can be mixed. For example, all of the following are valid: • --retention=D,weekly • --retention=none • --retention=Daily,W,monthly The --expiration and --retention options are mutually exclusive.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli backup edit** command.

22236	Client does not exist.
22552	Backup does not exist.
22556	Changed backup retention.
22557	Failed to modify retention of a backup.
22558	Multiple retention tags exist.

backup restore

The **mccli backup restore** command restores data to a client.

This command returns an activity ID, which can be passed to **mccli activity show**, discussed on page 37, to get status for the restore activity.

This command supports any of the following restore scenarios:

◆ If restoring from normal (non-virtual) filesystem or application backup:

  • Restore entire backup to the same (non-virtual) client

  • Restore selected folders or files to the same (non-virtual) client

  • Redirected restore of selected folders or files to a different (non-virtual) client

◆ If restoring from VMware image backup:

- Restore an entire VMware image backup to the same virtual machine

- Redirected restore an entire VMware image backup to a different existing virtual machine

- Redirected restore an entire VMware image backup to the new virtual machine

- Restore selected folders or files to the same virtual machine

- Redirected restore of selected folders or files to a different virtual machine

Synopsis

```
mccli backup restore [GLOBAL-OPTIONS] [--cmd=STRING [--cmd=STRING]
    ...] [--data=STRING] [--datacenter=STRING]
    [--datastore-name=STRING] [--dest-client-domain=STRING]
    [--dest-client-name=STRING] [--dest-client-password=STRING]
    [--dest-client-username=STRING] [--dest-dir=STRING]
    [--domain=STRING(/)] [--esx-host-name=STRING] [--folder=STRING]
    --labelNum=INTEGER --name=STRING --plugin=INTEGER
    [--restore-vm-to=STRING] [--virtual-center-name=STRING]
    [--vm-configuration=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli backup restore** command.

Table 17 Command options for **mccli backup restore** (page 1 of 3)

Option	Description

--cmd=STRING	Specifies one ore more optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command.
	 For example, this is valid --cmd syntax:
	 `--cmd="verbose=5" --cmd="throttle=5"`
	
	 However, this is not valid --cmd syntax:
	 `--cmd="verbose=5 throttle=5"`

--data=STRING	Specifies a single optional target folder or file to restore. When restoring a VMWare image backup, STRING can also specify a single virtual disk (VMDK file) to restore or ALL, to restore all virtual disks. For example: • --data=All restores all virtual disks • --data=BASE-VMDK-FILENAME restores only the virtual disk defined by BASE-VMDK-FILENAME Each --data option can only specify one target folder or file to restore. Use multiple --data options to restore multiple targets.

Table 17 Command options for **mccli backup restore** (page 2 of 3)

Option	Description
`--datacenter=STRING`	When restoring a VMware image backup, specifies a fully qualified datacenter name in vCenter in the format of PATH/NAME, where PATH is the full folder location and NAME is the datacenter name. This argument is required when restoring an entire image or selected disks to the same, different existing, or new virtual machine. It is not valid when restoring individual folders or files.
`--datastore-name=STRING`	When restoring a VMware image backup to a new virtual machine, specifies the datastore name to be removed from the Avamar image proxy client. This argument is required.
`--dest-client-domain=STRING`	Specifies the Avamar server domain that contains the alternative client specified by the --dest-client-name argument.
`--dest-client-name=STRING`	When performing any redirected restore, specifies the destination client. When performing a redirected restore of specific folders or files, you must also supply --dest-dir. If you supply a fully qualified client name (for example, /clients/MyClient), then the --dest-client-domain option is ignored.
`--dest-client-password=STRING`	When restoring folders or files from a VMware image backup, specifies the destination client password.
`--dest-client-username=STRING`	When restoring folders or files from a VMware image backup, specifies the destination client username.
`--dest-dir=STRING`	When performing a redirected restore of specific folders or files, specifies the destination folder for the restored folders or files. This argument is required when performing any redirected restore of specific folders or files, and optional for all other operations.
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the client specified by the --name argument.
`--esx-host-name=STRING`	When restoring a VMware image backup to a new virtual machine, specifies a fully qualified ESX server hostname in a datacenter, in the format of PATH/NAME, where PATH is the full folder location and NAME is the hostname.
`--folder=STRING`	When restoring a VMware image backup to an existing or new virtual machine, specifies the folder path for the destination virtual machine in the datacenter.
`--labelNum=INTEGER`	Specifies the backup to use for the restore. This argument is required. Use **mccli snapup show**, discussed on page 179, to return a list of valid backup label numbers.

Table 17 Command options for **mccli backup restore** (page 3 of 3)

Option	Description
--name=STRING	Specifies the target client for the restore. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain option is ignored.
--plugin=INTEGER	Specifies the plug-in ID for the restore. This argument is required. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs. When performing a redirected restore, this argument should specify a plug-in that is compatible with the alternative destination client, not the plug-in that was originally used to perform the backup.
--restore-vm-to=STRING	When restoring a VMware image backup, specifies the type of restore operation. STRING must be one of the following: • original — restores the entire backup image to the original virtual machine • existing — restores the entire backup image to a different existing virtual machine • new — restores the entire backup image to a new virtual machine • flr — restores folders or files
--virtual-center-name=STRING	When restoring a VMware image backup to an existing or new virtual machine, specifies the vCenter name.
--vm-configuration=BOOLEAN(false)	When restoring a VMware image backup to an existing virtual machine, and this argument is true, VMware configuration files are restored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli backup restore** command.

22236   Client does not exist.

22297   Request to restore is rejected.

22312   Client restore scheduled.

23009   Invalid plugin specified on the CLI.

Examples

The following example restores an entire VMware image backup to the original virtual machine:

```
mccli backup restore --name=/Test/Test1 --labelNum=1 --plugin=3016
   --restore-vm-to=original
0,22312,client restore scheduled.
Attribute    Value
----------   ----------------------------------------------------------
client       /Test/Test1
activity-id  9131666434371809
```

The following example restores an entire VMware image backup to a different existing virtual machine:

```
mccli backup restore --name=/Test/Test1 --labelNum=1 --plugin=3016
   --restore-vm-to=existing --datacenter=Test/ESX40
   --virtual-center=Test--dest-client-name=abcdef
0,22312,client restore scheduled.
Attribute    Value
----------   ----------------------------------------------------------
client       /Test/Test1
activity-id  9131666434371809
```

The following example restores virtual disk from a VMware image backup to a new virtual machine:

```
mccli backup restore --name=/Test/Test1/ --labelNum=2 --plugin=3016
   --restore-vm-to=new --datacenter=VAAYU-DEV-WIN
   --virtual-center-name=10.31.183.228 --datastore-name=Storage1
   --esx-host-name=10.31.183.7 --dest-client-name=Test12345
   --folder=Discovered virtual machine --data=[Storage1]
   Disk-1/Disk-1.vmdk
0,22312,client restore scheduled.
Attribute    Value
----------   ----------------------------------------------------------
client       /Test/Test1/Disk-1_UohhwqYknag96PWb1WrEnA
activity-id  9131666441684809
```

The following example restores selected folders and files to a different existing virtual machine:

```
mccli backup restore --name=/Test/Test1 --labelNum=1 --plugin=3019
   --restore-vm-to=flr --dest-client-name=Test5
   --dest-client-domain=/Test/Test1
   --dest-client-username=Administrator --data=C:\\1.log
   --dest-dir=C:\\ --dest-client-password=abcdefgh
```

backup show

The **mccli backup show** command returns all backups currently stored on the Avamar server or a Data Domain system for a client.

Synopsis

```
mccli backup show [GLOBAL-OPTIONS] [--after=STRING --before=STRING]
   [--dir=STRING(/)] [--domain=STRING(/)] [--labelNum=INTEGER]
   --name=STRING [--recursive=BOOLEAN(false)] [--retention=STRING]
   [--verbose=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli backup show** command.

Table 18 Command options for **mccli backup show** (page 1 of 2)

Option	Description
`--after=STRING`	Used with --before to specify a range of backups dates to return. STRING must be in the format of YYYY-MM-DD, where YYYY is the four-digit year, MM is the two-digit calendar month, and DD is the two-digit day of the month.
`--before=STRING`	Used with --after to specify a range of backups dates to return. STRING must be in the format of YYYY-MM-DD, where YYYY is the four-digit year, MM is the two-digit calendar month, and DD is the two-digit day of the month.
`--dir=STRING(/)`	Specifies the initial folder of the backup from which to begin listing folders and files.
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the client specified by the --name argument.
`--labelNum=INTEGER`	Lists the contents of the specified backup. Use **backup show** without supplying --labelNum to return a list of backups with integer label numbers.
`--name=STRING`	Specifies the client from which to show backups. This argument is required. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Table 18 Command options for **mccli backup show** (page 2 of 2)

Option	Description
`--recursive=BOOLEAN(false)`	If true, backup folders and files are recursively shown.
`--retention=STRING`	If supplied, then only backups with the specified retention types are returned. STRING must be one or more of the following: • D or daily • W or weekly • M or monthly • Y or yearly • N or none Both short form and long form retention type values are allowed and can be mixed. For example, all of the following are valid: --retention=D,weekly --retention=none --retention=Daily,W,monthly
`--verbose=BOOLEAN(false)`	If true, then detailed information is returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli backup show** command.

22236   Client does not exist.

22504   Failed to retrieve the backups for a client.

Examples

The following command returns a list of backups stored in the system for MyClient.example.com:

```
mccli backup show --name=clients/MyClient.example.com
0,23000,CLI command completed successfully.

Created                 LabelNum   Size
----------------------  --------   ----------
2011-12-1 15:51:30 PST  2          4767841280
2011-12-1 15:00:23 PST  1          4750878720
```

The following command returns verbose file and folder information for a specific parent folder (C:/) in a specific client backup (the second backup taken from MyClient.example.com):

```
mccli backup show --name=clients/MyClient.example.com --verbose --labelNum=2 --dir=C:/
0,23000,CLI command completed successfully.
Type Date                 Size          Owner          Group        Permissions Name
---- -------------------  ------------- -------------- ------------ ----------- ------------------
Dir  2011-12-1 08:59:27   205,526,604   Administrators unknown      drwxrwxr-x  C:/Documents/
Dir  2010-12-12 11:33:54  1,601,595,257 Administrators unknown      dr-xr-x---  C:/Program Files/
Dir  2010-12-12 08:59:55  1,841         Administrator  None         drwx------  C:/RECYCLER/
Dir  2010-01-31 15:45:30  20,480        Administrators unknown      d--x------  C:/System/
Dir  2010-01-17 15:18:22  211           Administrators unknown      drwxrwx---  C:/tmp/
Dir  2010-01-18 09:14:25  1,600,258,500 Administrators unknown      drwxrwx---  C:/WINDOWS/
File 2010-12-09 15:21:15  211           Administrators unknown      -r-xr-x---  C:/boot.ini
File 2010-12-12 08:56:49  4,128         Administrators unknown      -rwxrwx---  C:/INFCACHE.1
File 2010-08-11 15:15:00  0             Administrators unknown      -rwxrwx---  C:/IO.SYS
File 2010-08-11 15:15:00  0             Administrators unknown      -rwxrwx---  C:/MSDOS.SYS
File 2010-08-04 03:00:00  47,564        Administrators unknown      -r-xr-x---  C:/NTDETECT.COM
File 2010-08-04 03:00:00  250,032       Administrators unknown      -r-xr-x---  C:/ntldr
```

backup validate

The **mccli backup validate** command initiates a validation of a backup.

This command returns an activity ID, which can be passed to **mccli activity show**, discussed on page 37, to get status for this validation activity.

Synopsis

```
mccli backup validate [GLOBAL-OPTIONS] [--cmd=STRING [--cmd=STRING]
   ...] [--dest-client-domain=STRING --dest-client-name=STRING]
   [--domain=STRING(/)] --labelNum=INTEGER --name=STRING
   --plugin=INTEGER [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli backup validate** command.

Table 19 Command options for **mccli backup validate**

Option	Description

`--cmd=STRING`	Specifies one ore more optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command.
	For example, this is valid --cmd syntax:
	`--cmd="verbose=5" --cmd="throttle=5"`
	
	However, this is not valid --cmd syntax:
	`--cmd="verbose=5 throttle=5"`

`--dest-client-domain=STRING`	Used with --dest-client-name to validate a replicated backup (that is, a backup for a client in the /REPLICATE domain) on an alternate client. Specifies the Avamar server domain that contains the client specified by the --dest-client-name argument.
`--dest-client-name=STRING`	Used with --dest-client-domain to validate a replicated backup (that is, a backup for a client in the /REPLICATE domain) on an alternate client. Specifies the alternative target client for the validation. STRING must be a valid client name. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --dest-client-domain argument is ignored. This argument returns an error if you supply anything other than a valid client name from the /REPLICATE domain.
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the client specified by the --name argument.
`--labelNum=INTEGER`	Specifies the backup to validate. This argument is required.
`--name=STRING`	Specifies the client associated with the backup to validate. STRING must be a valid client name. This argument is required. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
`--plugin=INTEGER`	Specifies the plug-in ID for the restore. This argument is required. Use **mccli plugin show**, discussed on page 152, to return a list of valid plug-in IDs.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli backup validate** command.

22236   Client does not exist.

22298   Request to validate is rejected.

22315   Client validate scheduled.

23009   Invalid plugin specified on the CLI.

checkpoint

The **mccli checkpoint** resource is used to create and manage checkpoints on the Avamar server. Available commands are:

- ◆ "checkpoint cancel-validate" on page 51
- ◆ "checkpoint create" on page 52
- ◆ "checkpoint delete" on page 52
- ◆ "checkpoint show" on page 53
- ◆ "checkpoint validate" on page 54

checkpoint cancel-validate

The **mccli checkpoint cancel-validate** command cancels an active (currently running) checkpoint validation.

Synopsis

mccli checkpoint cancel-validate [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli checkpoint cancel-validate** command.

22615   A checkpoint validation was cancelled.

22616   No checkpoint validation is running.

checkpoint create

The **mccli checkpoint create** command creates a checkpoint. An MCS flush occurs as part of the checkpoint.

Synopsis

```
mccli checkpoint create [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli checkpoint create** command.

22608   An Avamar server checkpoint was successfully created.

22609   An error occurred creating an Avamar server checkpoint.

checkpoint delete

The **mccli checkpoint delete** command permanently deletes a checkpoint from the Avamar server or a Data Domain system.

Synopsis

```
mccli checkpoint delete [GLOBAL-OPTIONS] --cptag=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command option is available for the **mccli checkpoint delete** command.

Table 20 Command options for **mccli checkpoint delete**

Option	Description
`--cptag=STRING`	Specifies the checkpoint to delete. STRING must be a valid checkpoint ID. This argument is required.
	Use **mccli checkpoint show**, discussed on page 53, to return a list of valid checkpoint IDs.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli checkpoint delete** command.

22610   An Avamar server checkpoint was successfully deleted.

22611   An Avamar server checkpoint was not successfully deleted.

22617   The specified checkpoint was not found.

checkpoint show

The **mccli checkpoint show** command lists all checkpoints with summary or detailed information.

Synopsis

```
mccli checkpoint show [GLOBAL-OPTIONS] [--verbose=BOOLEAN(false)]
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command option is available for the **mccli checkpoint show** command.

Table 21 Command options for **mccli checkpoint show**

Option	Description
`--verbose=BOOLEAN(false)`	If true, then detailed checkpoint information is returned. If false or not supplied, then summary information is returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event code is associated with the **mccli checkpoint show** command.

22531   Unexpected exception occurred.

checkpoint validate

The **mccli checkpoint validate** command validates a checkpoint.

Synopsis

```
mccli checkpoint validate [GLOBAL-OPTIONS] [--checktype=TYPE]
   --cptag=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli checkpoint validate** command.

Table 22 Command options for **mccli checkpoint validate**

Option	Description
--checktype=TYPE	Constrains checkpoint validation to one or more of the following checkpoint TYPES: • full — Validation performed all checks. • rolling — Validation was rolling HFS check.
--cptag=STRING	Specifies the checkpoint to validate. STRING must be a valid checkpoint ID. This argument is required. Use **mccli checkpoint show,** discussed on page 53, to return a list of valid checkpoint IDs.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli checkpoint validate** command.

22612   Starting to validate an Avamar server checkpoint.

22617   The specified checkpoint was not found.

client

The **mccli client** resource is used to create and manage client accounts on the Avamar server. Available commands are:

Enumerating client resources

Several **mccli client** commands require precise case-sensitive names (STRING values) for the following client resources:

- Datasets
- Retention policies
- Plug-ins

The preferred way to view a list of valid datasets, retention policies, and plug-ins is to use the following commands, respectively:

- **mccli dataset show,** discussed on page 106
- **mccli retention show,** discussed on page 158
- **mccli plugin show,** discussed on page 152

You can also view a list of valid datasets, retention policies, and plug-ins using the Avamar Administrator **Tools › Manage All Datasets, Manage All Retention Policies,** and **Manage Agents & Plug-ins** features, respectively.

client activate

The **mccli client activate** command initiates activation of a specific client by first paging that client. If the client responds, then activation is completed and a Client ID (CID) is assigned to that client.

> **NOTICE**
>
> The **mccli client activate** command has been deprecated in favor of **mccli client invite**, discussed on page 75.

Synopsis

```
mccli client activate [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client activate** command.

Table 23 Command options for **mccli client activate**

Option	Description
--domain=STRING(/)	Specifies the domain that contains the client specified by the --name argument.
--name=STRING	Specifies the client to activate. STRING must be a valid client name. This argument is required. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client activate** command.

22236	Client does not exist.
22237	Client invited to activate with server.
22263	Client registration error.
22271	Client registration error - unable to contact client on port.
22280	Client reconnect error - host name mismatch.
22282	Client reconnect error - unknown ID.
22295	Client registration error - server not available.

client add

The **mccli client add** command registers a new client with the MCS and adds it to the default group. The client need not be connected to the network. Registering a client allows you to subsequently define policies for that client, even if the client has not yet been activated.

> **NOTICE**
>
> The **mccli client add** command cannot be used to add clients to the REPLICATE domain.

Synopsis

```
mccli client add [GLOBAL-OPTIONS]
    [--changed-block-tracking=BOOLEAN(false)] [--datacenter=STRING]
    [--contact=STRING] [--dataset-domain=STRING(/)] [--dataset=STRING]
    --domain=STRING(/) [--email=STRING][--encryption=STRING(None)]
    [--folder=STRING] [--location=STRING] --name=STRING
    [--override-dataset=BOOLEAN] [--override-encryption=BOOLEAN]
    [--override-retention=BOOLEAN] [--overtime=BOOLEAN]
    [--overtime-option=STRING] [--pageable=BOOLEAN] [--pageaddr=STRING]
    [--pageport=STRING] [--phone=STRING] [--retention-domain=STRING(/)]
    [--retention=STRING] [--type=STRING(normal)]
    [--virtual-center-name=STRING] [--virtual-center-password=STRING]
    [--virtual-center-port=Integer] [--virtual-center-username=STRING]
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client add** command.

Table 24 Command options for **mccli client add** (page 1 of 4)

Option	Description
`--changed-block-tracking=BOOLEAN(false)`	If true, when adding a virtual machine client, changed block tracking is enabled.
`--contact=STRING`	Specifies contact information for the new client.
`--datacenter=STRING`	When adding a virtual machine client, specifies a fully qualified datacenter name in vCenter in the format of PATH/NAME, where PATH is the full folder location and NAME is the datacenter name. This argument is required.
`--dataset-domain=STRING(/)`	Specifies the Avamar server domain that contains the dataset specified by the --dataset argument.
`--dataset=STRING`	Specifies the dataset that the client should use when performing on-demand backups and restores, or when the group dataset is overridden.

Table 24 Command options for **mccli client add** (page 2 of 4)

Option	Description
--domain=STRING(/)	Specifies the domain for the new client.
--email=STRING	Specifies the email address for the primary contact (responsible party) for the client.
--encryption=STRING	Specifies the encryption method that the client should use when performing on-demand backups and restores, or when the group encryption method is overridden. STRING must be one of the following: • High — Strongest available encryption setting for that specific client platform. • Medium — Medium strength encryption. • None — No encryption. • AES 128-bit — This setting is deprecated in version 4.1. • Axion — This setting is deprecated in version 4.1. **Note:** The exact encryption technology and bit strength used for any given client/server connection depends on a number of factors, including the client platform and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.
--folder=STRING	When adding a virtual machine client, specifies the folder path for virtual machines in the datacenter.
--location=STRING	Specifies location information for the new client.
--name=STRING	Specifies the new client name. This argument is required. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
--override-dataset=BOOLEAN	If true, then use the client dataset instead of the group dataset for scheduled backups.
--override-encryption=BOOLEAN	If true, then use the encryption method specified by --encryption instead of the group encryption method for scheduled backups.
--override-retention=BOOLEAN	If true, then use the retention policy specified by --retention instead of the group retention policy for scheduled backups.

Table 24 Command options for **mccli client add** (page 3 of 4)

Option	Description
--overtime-option=STRING	Specifies scheduled group backup overtime behavior. STRING must be one of the following: • ALWAYS — Scheduled group backups can always run past the schedule duration setting. • NEVER — Scheduled group backups can never run past the schedule duration setting. • NEXT — Only the next scheduled group backup can run past the schedule duration setting. • NEXT_SUCCESS — Scheduled group backups can run past the schedule duration setting until a successful backup is completed. The default is NEXT_SUCCESS.
--overtime=BOOLEAN	If true, client can exceed its backup window during scheduled backups.
--pageable=BOOLEAN	If true, client can be paged for the purpose of initiating activation or picking up new backup or restore work.
--pageaddr=STRING	Specifies the IP address that the Avamar server can use to contact the client.
--pageport=STRING	Specifies the data port that the Avamar server can use to contact the client.
--phone=STRING	Specifies the voice telephone information for the primary contact (responsible party).
--retention-domain=STRING(/)	Specifies the Avamar server domain that contains the retention policy specified by the --retention argument.
--retention=STRING	Specifies the retention policy that the client should use when performing on-demand backups and restores, or when the group retention policy is overridden.
--type=STRING(normal)	Specifies the type of client to add. STRING must be one of the following: • normal — any client that is not a vCenter, proxy, or virtual machine. This is the default client type. • vcenter — vCenter client. • proxy — Avamar proxy client. • vmachine — virtual machine client.
--virtual-center-name=STRING	When adding an Avamar proxy client, specifies the vCenter name.

Table 24 Command options for **mccli client add** (page 4 of 4)

Option	Description
`--virtual-center-password=STRING`	When adding a vCenter client, specifies the vCenter root user password.
`--virtual-center-port=Integer`	When adding a vCenter client, specifies the vCenter port address.
`--virtual-center-username=STRING`	When adding a vCenter client, specifies the vCenter root username.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client add** command.

22210	Client successfully added.
22238	Client exists.
22263	Client registration error.
22288	Dataset does not exist.
22289	Retention policy does not exist.
22558	A domain or client with this name already exists.
23012	Invalid encryption method specified on the CLI.
30922	Failed to connect to vCenter.

Notes

Consider the following notes on the --overtime-option and --overtime command options:

◆ If you do not supply --overtime-option and you supply --overtime=true, then --overtime-option is set to NEXT_SUCCESS.

◆ If you do not supply --overtime-option and you supply --overtime=false, then --overtime-option is set to NEVER.

◆ If you supply both --overtime-option and --overtime, then --overtime-option takes precedence.

Examples

The following command adds a new client called MyClient:

```
mccli client add --name=MyClient
0,22210,Client added
Attribute      Value
------------   ----------------------------------------------------
action         add
domain         /
node           MyClient
clientid       79a1042d7f5158c660fb7b863281f9787f8cb942
```

The following command adds a new client called MyClient, and formats the output in XML:

```
mccli client add --name=MyClient --xml
<CLIOutput>
   <Results>
       <ReturnCode>0</ReturnCode>
       <EventCode>22210</EventCode>
       <EventSummary>Client successfully added</EventSummary>
   </Results>
   <Data>
       <Row>
           <Attribute>action</Attribute>
           <Value>add</Value>
       </Row>
       <Row>
           <Attribute>domain</Attribute>
           <Value>/</Value>
       </Row>
       <Row>
           <Attribute>node</Attribute>
           <Value>MyClient</Value>
       </Row>
       <Row>
           <Attribute>clientid</Attribute>
           <Value>70279698b5f755b1bc2f41432b3b3471048478e7</Value>
       </Row>
   </Data>
</CLIOutput>
```

The following command adds a new proxy client called MyProxyClient:

```
mccli client add --name=/clients/MyProxyClient --type=proxy
   --virtual-center-name=10.31.183.228
0,22210,Client added
Attribute                    Value
--------------------------------------------------------------------
restoreOnly                  false
AllowClientOverrideSchedule  false
pagePort                     N/A
plugins                      <Plugins/>
registeredDate               N/A
nodeName                     MyProxyClient
display-nodeName             MyProxyClient
DSOverride                   false
checkinDate                  N/A
windowsHardwareProfile       N/A
clientid                     f8888acc3c2fb8db604bd6a1abbacc5bd57b9a9d
encryptionMethod             high
pageAddress                  N/A
canPage                      true
backedUpDate                 N/A
action                       add
AllowFileSelectionOnSccSnapupstrue
agentVersion                 unKnown
pageAddrLocked               false
```

windowsID	N/A
enabled	false
isClientOs	false
DATASETID	Default:SNAPID
AllowClientAddToDataset	false
retryCnt	2
overtimeOption	NEXT_SUCCESS
RPOverride	false
modifiedDate	Thu Aug 25 04:45:21 UTC 2011
AllowSccSnapups	true
nodeAddress	N/A
timeOut	10
registered	false
POLICYID	Default:POLICYID
overrideEncryption	false
OverrideStandardSccRetpol	false
fullName	/clients/MyProxyClient

The following command adds a new virtual machine client from this vSphere 5 environment:
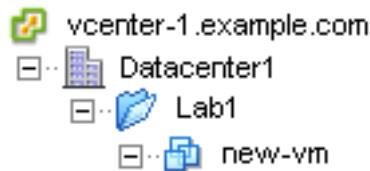

```
mccli client add --type=vmachine --name=new-vm
    --datacenter=Datacenter1
    --domain=/vcenter-1.example.com/VirtualMachines --folder=Lab1
0,22210,Client added
Attribute                   Value
-----------------------------------------------------------------------
```

Attribute	Value
restoreOnly	false
AllowClientOverrideSchedule	false
pagePort	N/A
plugins	<Plugins><Plugin Build="ALL" Description="Linux VMware Image" Version="6.1.100"/></Plugins>
registeredDate	N/A
nodeName	new-vm_UgqlbhZtNPdDKnHEpQLMOQ
display-nodeName	new-vm
DSOverride	false
checkinDate	N/A
windowsHardwareProfile	N/A
clientid	1bdcf9e5f67a0e383202bbbe307473a59131fdcc
encryptionMethod	high
pageAddress	N/A

```
canPage                         true
backedUpDate                    N/A
action                          add
AllowFileSelectionOnSccSnapupstrue
agentVersion                    unKnown
pageAddrLocked                  false
windowsID                       N/A
enabled                         true
isClientOs                      false
DATASETID                       VMWARE:SNAPID
AllowClientAddToDataset         false
retryCnt                        2
overtimeOption                  NEXT_SUCCESS
RPOverride                      false
modifiedDate                    Thu Aug 25 05:06:38 UTC 2011
AllowSccSnapups                 true
nodeAddress                     N/A
timeOut                         10
registered                      false
POLICYID                        Default:POLICYID
overrideEncryption              false
OverrideStandardSccRetpol       false
fullName                        /10.31.183.228/ACMCommunity_UgqlbhZtNPdD
                                   KnHEpQLMOQ
```

client add-datastore

The **mccli client add-datastore** command adds a new datastore to be protected by the specified Avamar image proxy client.

Synopsis

```
mccli client add-datastore [GLOBAL-OPTIONS] --datacenter=STRING
   --datastore-name=STRING [--domain=STRING(/)] --name=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client add-datastore** command.

Table 25 Command options for **mccli client add-datastore**

Option	Description
--datacenter=STRING	Specifies a fully qualified datacenter name in vCenter in the format of PATH/NAME, where PATH is the full folder location and NAME is the datacenter name. This argument is required.
--datastore-name=STRING	Specifies the datastore name to be protected by the Avamar image proxy client. This argument is required.
--domain=STRING(/)	Specifies the Avamar server domain that contains the Avamar image proxy client specified by the --name argument.
--name=STRING	Specifies the Avamar image proxy client to which the datastore will be added. STRING must be a valid Avamar image proxy client name. This argument is required. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client add-datastore** command.

22236   Client does not exist.

24003   Failed to update datastore mappings of a client.

24004   Datastore mappings of a client successfully updated.

Examples

The following example adds datastor Storage1 and assigns proxy client backupproxy225 to it:

```
mccli client add-datastore --name=/clients/backupproxy225
   --datacenter=VAAYU-DEV-WIN --datastore-name=Storage1
0,24004,Datastore mappings of a client successfully updated.
Attribute    Value
-----------  ----------------------
proxyclient  /clients/backupproxy225
```

client backup-dataset

The **mccli client backup-dataset** command initiates an on-demand backup of a single client using the specific dataset. If you do not specify a dataset on the command line, then the default dataset associated with that client is used.

This command returns an activity ID, which can be passed to **mccli activity show**, discussed on page 37, to get status for this backup activity.

> **NOTICE**
>
> The **mccli client backup-dataset** command cannot be used to back up clients in the REPLICATE domain.

Synopsis

```
mccli client backup-dataset [GLOBAL-OPTIONS] [--cmd=STRING
   [--cmd=STRING] ...] [--dataset=STRING] [--dataset-domain=STRING(/)]
   [--domain=STRING(/)] --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client backup-dataset** command.

Table 26 Command options for **mccli client backup-dataset** (page 1 of 2)

Option	Description

`--cmd=STRING`	Specifies one ore more optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command.
	 For example, this is valid --cmd syntax:
	 `--cmd="verbose=5"`
	 `--cmd="throttle=5"`
	
	 However, this is not valid --cmd syntax:
	 `--cmd="verbose=5 throttle=5"`

Table 26 Command options for **mccli client backup-dataset** (page 2 of 2)

Option	Description
--dataset=STRING	Specifies the dataset to use for the backup. If you do not supply the argument, then the default dataset associated with that client is used. Use **mccli dataset show**, discussed on page 106, to return a list of valid dataset names.
--dataset-domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --dataset argument.
--domain=STRING(/)	Specifies the Avamar server domain that contains the client specified by the --name argument.
--name=STRING	Specifies which client is backed up. STRING must be a valid client name. This argument is required. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client backup-dataset** command.

22228   A client was not backed up because it is disabled, retired, or its plug-in(s) has backups disabled.

22236   Client does not exist.

22253   Client Adhoc Backup Request Error - Exception.

22305   Client backup scheduled.

22309   Invalid plugin specified on the CLI.

23003   Invalid dataset name specified on the CLI.

client backup-group-dataset

The **mccli client backup-group-dataset** command initiates an on-demand backup of a single client using the group dataset. The client must already be a member of the group, or the backup fails.

This command initiates an on-demand client backup, not an on-demand group backup. Therefore, even if the client is a member of a group that is disabled and you specify the dataset for the group, the backup still occurs.

Synopsis

```
mccli client backup-group-dataset [GLOBAL-OPTIONS] [--cmd=STRING
   [--cmd=STRING] ...] [--domain=STRING(/)] [--group-domain=STRING(/)]
   --group-name=STRING --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client backup-group-dataset** command.

Table 27 Command options for **mccli client backup-group-dataset**

Option	Description

`--cmd=STRING`	Specifies one ore more optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command.
	For example, this is valid --cmd syntax:
	`--cmd="verbose=5" --cmd="throttle=5"`
	
	However, this is not valid --cmd syntax:
	`--cmd="verbose=5 throttle=5"`

`--domain=STRING(/)`	Specifies the Avamar server domain that contains the client specified by the --name argument.
`--group-domain=STRING(/)`	Specifies the Avamar server domain that contains the group specified by the --group-name argument.
`--group-name=STRING`	Specifies the group dataset to use for the backup. STRING must be a valid group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --group-domain argument is ignored.
`--name=STRING`	Specifies which client is backed up. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client backup-group-dataset** command.

22226   Group disabled.

22228   A client was not backed up because it is disabled, retired, or its plug-in(s) has backups disabled.

22234   Group does not exist.

22236   Client does not exist.

22241   Client is not a member of group.

22253   Client Adhoc Backup Request Error - Exception.

22305   Client backup scheduled.

client backup-target

The **mccli client backup-target** command initiates an on-demand backup of folders or files on a client.

This command returns an activity ID, which can be passed to **mccli activity show**, discussed on page 37, to get status for this backup activity.

NOTICE

The **mccli client backup-target** command cannot be used to back up folders or files belonging to clients in the REPLICATE domain.

Synopsis

```
mccli client backup-target [GLOBAL-OPTIONS] [--cmd=STRING
   [--cmd=STRING] ...] [--domain=STRING(/)] --name=STRING
   [--plugin=INTEGER] --target=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli backup-target** command.

Table 28 Command options for **mccli backup-target** (page 1 of 2)

Option	Description

`--cmd=STRING`	Specifies one ore more optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command.
	For example, this is valid --cmd syntax:
	`--cmd="verbose=5" --cmd="throttle=5"`
	
	However, this is not valid --cmd syntax:
	`--cmd="verbose=5 throttle=5"`

`--domain=STRING(/)`	Specifies the Avamar server domain that contains the client specified by the --name argument.

Table 28 Command options for **mccli backup-target** (page 2 of 2)

Option	Description
--name=STRING	Specifies which client is backed up. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
--plugin=INTEGER	Specifies the plug-in ID. Use **plugin show,** discussed on page 152, to return a list of valid plug-in IDs.
--target=STRING	Specifies the folders or files to include in the backup. At least one --target=STRING argument is required, and you can supply more than one --target=STRING on the same command line. If backing up a virtual machine, individual disks can be backed up by specifying the corresponding base VMDK file. Multiple virtual disks can be backed up with additional --target=STRING argument s. If backing up non-virtual filesystems, this argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli backup-target** command.

22228   A client was not backed up because it is disabled, retired, or its plug-in(s) has backups disabled.

22236   Client does not exist.

22253   Client Adhoc Backup Request Error - Exception.

22305   Client backup scheduled.

22309   Invalid plugin specified on the CLI.

Examples

The following example backs up a specific virtual disk:

```
mccli client backup-target --name=/10.31.183.228/Test1Windows222
   --plugin=3016 --target=[Storage2]
   Test1Windows222/Test1Windows222_1.vmdk
0,22305,client backup scheduled.
Attribute    Value
-----------  -------------------------------------------------
client       /10.31.183.228/Test1Windows222
target       [Storage2] Test1Windows222/Test1Windows222_1.vmdk
activity-id  9131407958056109
```

client delete

The **mccli client delete** command permanently deletes a client and its backups from the Avamar server.

Synopsis

```
mccli client delete [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [DISPLAY-OPTIONS]
```

> **NOTICE**

The **mccli client delete** command cannot be used to delete clients in the REPLICATE domain.

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client delete** command.

Table 29 Command options for **mccli client delete**

Option	Description
--domain=STRING(/)	Specifies the domain that contains theclient specified by the --name argument.
--name=STRING	Specifies the client to delete. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client delete** command.

22212   Client deleted.

22236   Client does not exist.

22240   Client delete failed.

client edit

The **mccli client edit** command edits the properties for a client.

There are no default settings for the **mccli client edit** command. If you enter this command but do not explicitly supply options and values on the command line, then there is no change to the client.

> **NOTICE**
>
> The **mccli client edit** command cannot be used to edit clients in the REPLICATE domain.

Synopsis

```
mccli client edit [GLOBAL-OPTIONS] [--activated=BOOLEAN]
   [--allow-cis=BOOLEAN] [--allow-cis-fileselection=BOOLEAN]
   [--changed-block-tracking=BOOLEAN(false)] [--contact=STRING]
   [--dataset=STRING] [--dataset-domain=STRING(/)] --domain=STRING
   [--email=STRING] [--enabled=BOOLEAN] [--encryption=STRING(None)]
   [--location=STRING] --name=STRING [--new-name=STRING]
   [--override-cis-retention=BOOLEAN] [--override-dataset=BOOLEAN]
   [--override-encryption=BOOLEAN] [--override-retention=BOOLEAN]
   [--overtime=BOOLEAN] [--overtime-option=STRING]
   [--page-detection=STRING(Automatic)] [--pageable=BOOLEAN]
   [--pageaddr=STRING] [--pageport=STRING] [--phone=STRING]
   [--retention=STRING] [--retention-domain=STRING(/)]
   [--timeout=STRING] [--virtual-center-name=STRING]
   [--virtual-center-password=STRING]
   [--virtual-center-port=Integer(443)]
   [--virtual-center-username=STRING] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client edit** command.

Table 30 Command options for **mccli client edit** (page 1 of 4)

Option	Description
`--activated=BOOLEAN`	If true, client is assumed to have been previously activated.
`--allow-cis-fileselection=BOOLEAN`	If true, users can make file selections when they initiate a backup of the client.
`--allow-cis=BOOLEAN`	If true, users can initiate a backup of the client.
`--changed-block-tracking=BOOLEAN(false)`	If true, when editing a virtual machine client, changed block tracking is enabled.
`--contact=STRING`	Specifies contact information for the client.
`--dataset-domain=STRING(/)`	Specifies the Avamar server domain that contains the dataset specified by the --dataset argument.

Table 30 Command options for **mccli client edit** (page 2 of 4)

Option	Description
`--dataset=STRING`	Specifies the dataset that the client should use when performing on-demand backups and restores, or when the group dataset is overridden.
`--domain=STRING`	Specifies the domain that contains the client specified by the --name argument.
`--email=STRING`	Specifies an email address for the primary contact (responsible party).
`--enabled=BOOLEAN`	If true, client is eligible to immediately participate in on-demand and group backup and restore activities. If false, then the client is disabled.
`--encryption=STRING`	Specifies the encryption method that the client should use when performing on-demand backups and restores, or when the group encryption method is overridden. STRING must be one of the following: • High — Strongest available encryption setting for that specific client platform. • Medium — Medium strength encryption. • None — No encryption. • AES 128-bit — This setting is deprecated in version 4.1. • Axion — This setting is deprecated in version 4.1. **Note:** The exact encryption technology and bit strength used for any given client/server connection depends on a number of factors, including the client platform and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.
`--location=STRING`	Specifies location information for the client.
`--name=STRING`	Specifies the client to edit. STRING must be a valid client name. This argument is required. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
`--new-name=STRING`	Specifies a new client name.
`--override-cis-retention=BOOLEAN`	If true, the retention policy is used for all user-initiated backups of the client.
`--override-dataset=BOOLEAN`	If true, use the client dataset instead of the group dataset for scheduled backups.
`--override-encryption=BOOLEAN`	If true, use the client encryption method instead of the group encryption method for scheduled backups.

Table 30 Command options for **mccli client edit** (page 3 of 4)

Option	Description
`--override-retention=BOOLEAN`	If true, use the client retention policy instead of the group retention policy for scheduled backups.
`--overtime-option=STRING`	Specifies scheduled group backup overtime behavior. STRING must be one of the following: • ALWAYS — Scheduled group backups can always run past the schedule duration setting. • NEVER — Scheduled group backups can never run past the schedule duration setting. • NEXT — Only the next scheduled group backup can run past the schedule duration setting. • NEXT_SUCCESS — Scheduled group backups can run past the schedule duration setting until a successful backup is completed. Default is NEXT_SUCCESS.
`--overtime=BOOLEAN`	If true, the client can exceed its backup window during backup activities.
`--page-detection=STRING(auto)`	Specifies client page detection mode. STRING must be one of the following: • Automatic — use automatic paging detection (that is, ignore `--pageaddr` and `--pageport` settings) • Manual — use `--pageaddr` and `--pageport` settings
`--pageable=BOOLEAN`	If true, the client can be paged for the purpose of initiating activation or picking up new backup or restore work.
`--pageaddr=STRING`	Specifies client paging IP address.
`--pageport=STRING`	Specifies client paging port. The default page port is 28002.
`--phone=STRING`	Specifies voice telephone information for the primary contact (responsible party) for the client.
`--retention-domain=STRING(/)`	Specifies the Avamar server domain that contains the retention policy specified by the --retention argument.
`--retention=STRING`	Specifies the retention policy that the client should use when performing on-demand backups and restores, or when the group retention policy is overridden.
`--timeout=STRING`	Specifies the client browse timeout in seconds.
`--virtual-center-name=STRING`	When editing an Avamar proxy client, specifies the vCenter name.

Table 30 Command options for **mccli client edit** (page 4 of 4)

Option	Description
`--virtual-center-password=STRING`	When editing a vCenter client, specifies the vCenter root user password.
`--virtual-center-port=Integer(443)`	When editing a vCenter client, specifies the vCenter port address. Default port is 443.
`--virtual-center-username=STRING`	When editing a vCenter client, specifies the vCenter root username.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client edit** command.

22211	Client was successfully updated.
22236	Client does not exist.
22239	Client update failed.
23003	Invalid dataset name specified on the CLI.
23005	Invalid retention policy name specified on the CLI.
23012	Invalid encryption method specified on the CLI.
30922	Failed to connect to vCenter.

Notes

Consider the following notes about the --overtime-option and --overtime command options:

◆ If you do not supply --overtime-option and you supply --overtime=true, then --overtime-option is set to NEXT_SUCCESS.

◆ If you do not supply --overtime-option and you supply --overtime=false, then --overtime-option is set to NEVER.

◆ If you supply both --overtime-option and --overtime, then --overtime-option takes precedence.

Examples

The following example sets the data port for vCenter-1.example.com:

```
mccli client edit --name=/vCenter-1.example.com
   --virtual-center-port=446
0,22211,Client was updated.
```

The following example renames Proxy-1 to Proxy-2, and adds a contact phone number:

```
mccli client edit --name=/clients/Proxy1
   --virtual-center-name=vCenter-1.example.com --new-name=Proxy2
   --phone=555-1212
0,22211,Client was updated.
```

The following example renames virtual machine client vm-1 to vm-2:

```
mccli client edit --name=/clients/vm-1 --new-name=clients/vm-2
0,22211,Client was updated.
```

client invite

The **mccli client invite** command initiates activation of a specific client by first paging that client. If the client responds, then activation is completed and a Client ID (CID) is assigned to the client.

> **NOTICE**
>
> The **mccli client invite** command cannot be used to invite clients to activate in the REPLICATE domain.

Synopsis

```
mccli client invite [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client invite** command.

Table 31 Command options for **mccli client invite**

Option	Description
--domain=STRING(/)	Specifies the domain that contains the client specified by the --name argument.
--name=STRING	Specifies the client to invite. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client invite** command.

22236   Client does not exist.

22237   Client invited to activate with server.

22263   Client registration error.

22271   Client registration error - unable to contact client on port.

22280	Client reconnect error - host name mismatch.
22282	Client reconnect error - unknown ID.
22295	Client registration error - server not available.

client load-bulk

The **mccli client load-bulk** command reads a clients definition input file, then registers any clients defined in that clients definition input file.

Synopsis

```
mccli client load-bulk [GLOBAL-OPTIONS] --file=STRING
    [--verbose=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client load-bulk** command.

Table 32 Command options for **mccli client load-bulk**

Option	Description
`--file=STRING`	Specifies the full path and filename of the clients definition input file. The file must be in XML format. This argument is required. The *EMC Avamar Administration Guide* provides additional information about clients definition files.
`--verbose=BOOLEAN(false)`	If true, then a list of clients and their load status is output to stdout.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client load-bulk** command.

22263	Client registration error.
22532	Client(s) successfully bulk-loaded.
23006	CLI client bulk load encountered one or more errors.

client move

The **mccli client move** command permanently moves a client and all its backups from one domain to another.

> **NOTICE**
>
> The **mccli client move** command cannot be used to move clients in or out of the REPLICATE domain.

Synopsis

```
mccli client move [GLOBAL-OPTIONS] [--domain=STRING(/)] --name=STRING
    --new-domain=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client move** command.

Table 33 Command options for **mccli client move**

Option	Description
--domain=STRING(/)	Specifies the domain that contains the client specified by the --name argument.
--name=STRING	Specifies the client to move. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
--new-domain=STRING	Specifies the new domain for the client. This argument is required.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client move** command.

22515   Client was moved.

22236   Client does not exist.

22501   Failed to move client.

23019   Client move failed - still active.

client remove-datastore

The **mccli client remove-datastore** command removes datastore protection from the specified Avamar image proxy client.

Synopsis

```
mccli client remove-datastore [GLOBAL-OPTIONS] --datacenter=STRING
    --datastore-name=STRING [--domain=STRING(/)] --name=STRING
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client remove-datastore** command.

Table 34 Command options for **mccli client remove-datastore**

Option	Description
--datacenter=STRING	Specifies a fully qualified datacenter name in vCenter in the format of PATH/NAME, where PATH is the full folder location and NAME is the datacenter name. This argument is required.
--datastore-name=STRING	Specifies the datastore name to be removed from the Avamar image proxy client. This argument is required.
--domain=STRING(/)	Specifies the Avamar server domain that contains the Avamar image proxy client specified by the --name argument.
--name=STRING	Specifies the Avamar image proxy client to which the datastore will be removed. STRING must be a valid Avamar image proxy client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client remove-datastore** command.

22236   Client does not exist.

24003   Failed to update datastore mappings of a client.

24004   Datastore mappings of a client successfully updated.

Examples

The following example removes datastore Storage1 from client backupproxy225:

```
mccli client remove-datastore --name=/clients/backupproxy225
   --datacenter=VAAYU-DEV-WIN --datastore-name=Storage1
0,24004,Datastore mappings of a client successfully updated.
Attribute    Value
-----------  ----------------------
proxyclient  /clients/backupproxy225
```

client retire

The **mccli client retire** command retires a client from active backup activities. All backups belonging to the client expire on the dates originally assigned to them.

> **NOTICE**
>
> The **mccli client retire** command cannot be used to retire clients in the REPLICATE domain.

Synopsis

```
mccli client retire [GLOBAL-OPTIONS] [--domain=STRING(/)]
   [--expiration=STRING] --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client retire** command.

Table 35 Command options for **mccli client retire**

Option	Description
--domain=STRING(/)	Specifies the domain that contains the client specified by the --name argument.
--expiration=STRING	Specifies the expiration of backups for the retired client. STRING must be one of the following: • YYYY-MM-DD — Specifies an explicit date. • +nn{D|W|M|Y} — Specifies a duration from today. (For example, +4W specifies 4 weeks from today.) • NO_EXPIRATION
--name=STRING	Specifies the client to retire. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client retire** command.

22236   Client does not exist.

22506   Failed to retire a client.

22512   Client was retired.

client show

The **mccli client show** command lists clients and their properties.

Synopsis

```
mccli client show [GLOBAL-OPTIONS] [--domain=STRING(/)]
   [--name=STRING] [--recursive=BOOLEAN] [--replicated=Boolean(false)]
   [--retired=BOOLEAN(false)] [--verbose=BOOLEAN(false)]
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client show** command.

Table 36 Command options for **mccli client show** (page 1 of 2)

Option	Description
--domain=STRING(/)	Specifies the domain that contains the client specified by the --name argument. If you supply --retired, then this argument is ignored.
--name=STRING	Specifies a client name. If not supplied, all clients with the specified domain are listed. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
--recursive=BOOLEAN(false)	If true, then the command shows all clients in the domain and any subdomains specified by the --domain=STRING argument. If false or not supplied, then the command only shows clients in the specified domain. Subdomains are not examined.

Table 36 Command options for **mccli client show** (page 2 of 2)

Option	Description
`--replicated=Boolean(false)`	If true, then clients in the REPLICATE domain are shown. If false or not supplied, then clients in the REPLICATE domain are not shown.
`--retired=BOOLEAN(false)`	If true, then retired clients are listed. If supplied, then the --domain=STRING argument is ignored.
`--verbose=BOOLEAN(false)`	If you do not supply --name and the option is false or not supplied, then only the domain and client name are returned for each client. If true, then other properties are also returned. This option is only meaningful when you do not supply the --name option. When you supply the --name option, the details for the client are always returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client show** command.

22236   Client does not exist.

22542   Domain does not exist.

Examples

The following command returns a simple list of all clients in the clients domain:

```
mccli client show --domain=clients
0,23000,CLI command completed successfully.
command completed successfully
 Name      Domain
 -------- --------
 MyClient /clients
```

The following command returns a detailed list of properties for MyClient.example.com:

```
mccli client show --name=/clients/MyClient.example.com
0,23000,CLI command completed successfully.
command completed successfully
Attribute                               Value
--------------------------------------- -------------------------------
Name                                    MyClient
Fully Qualified Name                    /clients/MyClient.example.com
Operating System                        Windows XP Professional SP 1.0
Paging                                  Yes
Page Address                            MyClient.example.com
Page Port                               28002
Paging Detection                        Automatic
Agent Version                           6.5.100-nnn
CID                                     6e6241ae236c2ab58205f413f24d3d6e0
CID Assigned                            2010-10-13 00:00:00 PDT
Activated                               Yes
```

Activated Date	2010-10-13 12:28:13 PDT
Disabled	No
Agent Last Started	2011-12-1 06:00:00 PST
Last Check-in	2011-12-1 11:00:59 PST
Override Group Dataset	No
Dataset Name	Default Dataset
Override Group Retention	No
Retention Policy	Default Retention
Allow Overtime	No
Restore Only	No
Encryption Method	High
Override Encryption	No
Allow client initiated backups	Yes
Allow file selection in client initiated backups	Yes
Override retention policy on client initiated backups	No
Contact Name	N/A
Contact Phone	N/A
Contact Email	N/A
Contact Location	N/A
Contact Notes	N/A
Member of Group	/Default Group
Plugin	Windows File System(3001)
Initial Install Date	2011-12-1 12:28:14 PDT
Last Version Registered	2011-12-1 12:28:14 PDT
Last Successful Backup	2011-12-1 22:00:41 PST

client show-datastore

The **mccli client show-datastore** command returns detailed information for a datastore protected by a specific Avamar image proxy client.

Synopsis

```
mccli client show-datastore [GLOBAL-OPTIONS] [--domain=STRING(/)]
    --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client show-datastore** command.

Table 37 Command options for **mccli client show-datastore**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the Avamar image proxy client specified by the --name argument.
--name=STRING	Specifies the Avamar image proxy client that is protecting the datastore of interest. STRING must be a valid Avamar image proxy client name. This argument is required. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client show-datastore** command.

22234	Group does not exist.
22236	Client does not exist.
24001	Failed to update proxy client mappings of a group.
24004	Proxy client mappings of a group successfully updated.

Examples

The following example shows all datastores protected by client backupproxy225:

```
mccli client show-datastore --name=/clients/backupproxy225
Name      Type  Accessible   Hosts
--------  ----  ----------   -----------
Storage1  VMFS  Yes          10.31.183.7
Storage2  VMFS  Yes          10.31.183.7
```

client show-plugins

The **mccli client show-plugins** command lists all Avamar plug-ins installed on a client. If you also specify a plug-in number, then detailed information is returned for the plug-in.

Synopsis

```
mccli client show-plugins [GLOBAL-OPTIONS] [--domain=STRING(/)]
    --name=STRING [--plugin=INTEGER] [--verbose=BOOLEAN(false)]
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client show-plugins** command.

Table 38 Command options for **mccli client show-plugins**

Option	Description
`--domain=STRING(/)`	Specifies the domain that contains the client specified by the --name argument.
`--name=STRING`	Specifies the client for which plug-ins are shown. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
`--plugin=INTEGER`	Specifies the plug-in ID. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.
`--verbose=BOOLEAN(false)`	If true, detailed plug-in information is returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client show-plugins** command.

22236   Client does not exist.

23009   Invalid plugin specified on the CLI.

client snapup-dataset

The **mccli client snapup-dataset** command initiates an on-demand backup of a single client using the specific dataset. If you do not specify a dataset on the command line, then the default dataset associated with that client is used.

> **NOTICE**
>
> This command has been deprecated in favor of **mccli client backup-dataset**, discussed on page 65. Whenever possible, you should use **mccli client backup-dataset** instead of **mccli client snapup-dataset**.

This command returns an activity ID, which can be passed to **mccli activity show**, discussed on page 37, to get status for this backup activity.

Synopsis

```
mccli client snapup-dataset [GLOBAL-OPTIONS] [--cmd=STRING
   [--cmd=STRING] ...] [--dataset=STRING] [--dataset-domain=STRING(/)]
   [--domain=STRING(/)] --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client snapup-dataset** command.

Table 39 Command options for **mccli client snapup-dataset** (page 1 of 2)

Option	Description

--cmd=STRING	Specifies one ore more optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command.
	For example, this is valid --cmd syntax:
	`--cmd="verbose=5" --cmd="throttle=5"`
	
	However, this is not valid --cmd syntax:
	`--cmd="verbose=5 throttle=5"`

--dataset=STRING	Specifies the dataset to use for the backup. If you do not supply the argument, then the default dataset associated with that client is used. Use **mccli dataset show,** discussed on page 106, to return a list of valid dataset names.
--dataset-domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --dataset argument.

Table 39 Command options for **mccli client snapup-dataset** (page 2 of 2)

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the client specified by the --name argument.
--name=STRING	Specifies which client is backed up. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client snapup-dataset** command.

22228	A client was not backed up because it is disabled, retired, or its plug-in(s) has backups disabled.
22236	Client does not exist.
22253	Client Adhoc Backup Request Error - Exception.
22305	Client backup scheduled.
22309	Invalid plugin specified on the CLI.
23003	Invalid dataset name specified on the CLI.

client snapup-group-dataset

The **mccli client snapup-group-dataset** command initiates an on-demand backup of a single client using the group dataset. The client must already be a member of the group, or the backup fails.

This command initiates an on-demand client backup, not an on-demand group backup. Therefore, even if the client is a member of a group that is disabled and you specify the dataset for the group, the backup still occurs.

> **NOTICE**

This command has been deprecated in favor of **mccli client backup-group-dataset**, discussed on page 66. Whenever possible, use **mccli client backup-group-dataset** instead of **mccli client snapup-group-dataset**.

This command returns an activity ID, which can be passed to **mccli activity show,** discussed on page 37, to get status for this backup activity.

Synopsis

```
mccli client snapup-group-dataset [GLOBAL-OPTIONS] [--cmd=STRING
   [--cmd=STRING] ...] [--domain=STRING(/)] [--group-domain=STRING(/)]
   --group-name=STRING --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client snapup-group-dataset** command.

Table 40 Command options for **mccli client snapup-group-dataset**

Option	Description

`--cmd=STRING`	Specifies one ore more optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command.
	 For example, this is valid --cmd syntax:
	 `--cmd="verbose=5" --cmd="throttle=5"`
	
	 However, this is not valid --cmd syntax:
	 `--cmd="verbose=5 throttle=5"`

`--domain=STRING(/)`	Specifies the Avamar server domain that contains the client specified by the --name argument.
`--group-domain=STRING(/)`	Specifies the Avamar server domain that contains the group specified by the --group-name argument.
`--group-name=STRING`	Specifies the group dataset to use for the backup. STRING must be a valid group name. This argument is required. Use **mccli group show**, discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --group-domain argument is ignored.
`--name=STRING`	Specifies which client is backed up. STRING must be a valid client name. This argument is required. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client backup-group-dataset** command.

22228	A client was not backed up because it is disabled, retired, or its plug-in(s) has backups disabled.
22234	Group does not exist.
22236	Client does not exist.
22241	Client is not a member of group.
22253	Client Adhoc Backup Request Error - Exception.
22305	Client backup scheduled.
23003	Invalid dataset name specified on the CLI.

client snapup-target

The **mccli client snapup-target** command initiates an on-demand backup of a single file or folder on a client.

To back up more than one file or folder, create a dataset and then use **mccli client snapup-dataset** to perform the backup.

> **NOTICE**
>
> This command has been deprecated in favor of **mccli client backup-target**, which is discussed on page 68. Whenever possible, use **mccli client backup-target** instead of **mccli client snapup-target**.

This command returns an activity ID, which can be passed to **mccli activity show**, discussed on page 37, to get status for this backup activity.

Synopsis

```
mccli client snapup-target [GLOBAL-OPTIONS] [--cmd=STRING
   [--cmd=STRING] ...] [--domain=STRING(/)] --name=STRING
   [--plugin=INTEGER] --target=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client snapup-target** command.

Table 41 Command options for **mccli client snapup-target**

Option	Description

`--cmd=STRING`	Specifies one ore more optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command.
	For example, this is valid --cmd syntax:
	`--cmd="verbose=5" --cmd="throttle=5"`
	
	However, this is not valid --cmd syntax:
	`--cmd="verbose=5 throttle=5"`

`--domain=STRING(/)`	Specifies the Avamar server domain that contains the client specified by the --name argument.
`--name=STRING`	Specifies which client is backed up. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
`--plugin=INTEGER`	Specifies the plug-in ID. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.
`--target=STRING`	Specifies the file or folder to include in backups. This argument is required. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli client snapup-target** command.

22228	A client was not backed up because it is disabled, retired, or its plug-in(s) has backups disabled.
22236	Client does not exist.
22253	Client Adhoc Backup Request Error - Exception.
22305	Client backup scheduled.
22309	Invalid plugin specified on the CLI.

client validate-bulk

The **mccli client validate-bulk** command validates a clients definition input file, which is used by the **mccli client load-bulk** command discussed on page 76. The validation process ensures that the XML data is properly formatted and that the file works with the **mccli client load-bulk** command.

Synopsis

```
mccli client validate-bulk [GLOBAL-OPTIONS] --file=STRING
   [--verbose=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli client validate-bulk** command.

Table 42 Command options for **mccli client validate-bulk**

Option	Description
--file=STRING	Specifies the clients definition input file. The file must be in XML format. This argument is required. The *EMC Avamar Administration Guide* provides additional information about clients definition files.
--verbose=BOOLEAN(false)	If true, then a list of clients and their load status is output to stdout.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event code is associated with the **mccli client validate-bulk** command.

23006   CLI client bulk load encountered one or more errors.

dataset

The **mccli dataset** resource is used to create and manage datasets on the Avamar server. Available commands are:

dataset add

The **mccli dataset add** command creates a dataset. By default, the initial dataset uses all available source data plug-ins and contains no explicit exclusion or inclusion entry entries. You can use other dataset commands to customize the dataset after it has been created, including:

Synopsis

```
mccli dataset add [GLOBAL-OPTIONS] [--alldata=BOOLEAN(true)]
    [--domain=STRING(/)] --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset add** command.

Table 43 Command options for **mccli dataset add**

Option	Description
--alldata=BOOLEAN(true)	If true, the initial dataset uses all available source data plug-ins and contains no explicit exclusion or inclusion entries. Set --alldata=false to create an empty dataset.
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--name=STRING	Specifies the name of the new dataset. This argument is required. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset add** command.

22219   Dataset created.

23008   Dataset already exists.

dataset add-exclude

The **mccli dataset add-exclude** command adds an exclusion entry to a dataset.

Synopsis

```
mccli dataset add-exclude [GLOBAL-OPTIONS] [--domain=STRING(/)]
   {--exclude=STRING [--exclude=STRING] ... | --exclude-file=FILE}
   --name=STRING --plugin=INTEGER [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset add-exclude** command.

Table 44 Command options for **mccli dataset add-exclude**

Option	Description
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
`--exclude=STRING`	Specifies the folders or files to exclude from the dataset. You must supply either --exclude=STRING or --exclude-file=FILE. Multiple --exclude=STRING arguments are allowed and must be separated by a space. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.
`--exclude-file=FILE`	Specifies the full path of a text FILE that contains exclusion entries. Each exclusion entry must conform to allowable --exclude=STRING syntax and be on a single line. You must supply either --exclude=STRING or --exclude-file=FILE.
`--name=STRING`	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show,** discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
`--plugin=INTEGER`	Specifies the plug-in ID. This argument is required. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset add-exclude** command.

23009   Invalid plugin specified on the CLI.

22220   Dataset modified.

22288   Dataset does not exist.

23022   Error parsing input file.

23023   Input file could not be found.

dataset add-include

The **mccli dataset add-include** command adds an inclusion entry to a dataset.

Synopsis

```
mccli dataset add-include [GLOBAL-OPTIONS] [--domain=STRING(/)]
    {--include=STRING [--include=STRING] ... | --include-file=FILE}
    --name=STRING --plugin=INTEGER [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset add-include** command.

Table 45 Command options for **mccli dataset add-include**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--include=STRING	Specifies the folders or files to add back to the dataset after an exclusion entry has excluded them. You must supply either --include=STRING or --include-file=FILE. Multiple --include=STRING arguments are allowed and must be separated by a space. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.
--include-file=FILE	Specifies the full path of a text FILE that contains inclusion entries. Each include entry must conform to allowable --include=STRING syntax and be on a single line. You must supply either --include=STRING or --include-file=FILE.
--name=STRING	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show,** discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
--plugin=INTEGER	Specifies the plug-in ID. This argument is required. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset add-include** command.

23009   Invalid plugin specified on the CLI.

22220   Dataset modified.

22288   Dataset does not exist.

23022   Error parsing input file.

23023   Input file could not be found.

dataset add-option

The **mccli dataset add-option** command adds a plug-in command (option) to a dataset.

Synopsis

```
mccli dataset add-option [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING {--option=STRING [--option=STRING] ...}
   --plugin=INTEGER {--value=STRING [--value=STRING] ...}
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset add-option** command.

Table 46 Command options for **mccli dataset add-option**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--name=STRING	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show,** discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
--option=STRING	Specifies the plug-in command to add to the dataset. This argument is required. Multiple --option=STRING arguments are allowed and must be separated by a space. Each client guide describes the valid plug-in commands for a dataset.
--plugin=INTEGER	Specifies the plug-in ID. This argument is required. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.
--value=STRING	Specifies a value for --option. This argument is required. Each client guide describes the valid plug-in commands for a dataset.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset add-option** command.

23009   Invalid plugin specified on the CLI.

22220   Dataset modified.

22288   Dataset does not exist.

Notes

Plug-in commands and values, specified by the --option=STRING and --value=STRING options, respectively, must occur in equal numbers. If you specify multiple --option=STRING options, then you must also supply an equal number of --value=STRING options to specify values for all plug-in commands. Furthermore, the option-value pairing is positional. In other words, the first occurrence of --option=STRING is assigned the value specified by the first occurrence of --value=STRING, the second occurrence of --option=STRING is assigned the value specified by the second occurrence of --value=STRING, and so forth.

dataset add-target

The **mccli dataset add-target** command adds a target file or folder entry to a dataset.

Synopsis

```
mccli dataset add-target [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING --plugin=INTEGER {--target=STRING
   [--target-file=FILE] ... | --target-file=FILE} [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset add-target** command.

Table 47 Command options for **mccli dataset add-target**

Option	Description
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
`--name=STRING`	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show,** discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
`--plugin=INTEGER`	Specifies the plug-in ID. This argument is required. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.
`--target=STRING`	Specifies a target file or folder to add to the dataset. This argument is required. You must supply either --target=STRING or --target-file=FILE. Multiple --target=STRING arguments are allowed and must be separated by a space. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.
`--target-file=FILE`	Specifies the full path of a text FILE that contains target entries. Each include entry must conform to allowable --target=STRING syntax and be on a single line. You must supply either --target=STRING or --target-file=FILE.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset add-target** command.

23009   Invalid plugin specified on the CLI.

22220   Dataset modified.

22288   Dataset does not exist.

23022   Error parsing input file.

23023   Input file could not be found.

dataset copy

The **mccli dataset copy** command copies a dataset, creating a new dataset.

Synopsis

```
mccli dataset copy [GLOBAL-OPTIONS] [--domain=STRING(/)] --name=STRING
    [--new-domain=STRING(/)] --new-name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset copy** command.

Table 48 Command options for **mccli dataset copy**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--name=STRING	Specifies the dataset to copy. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show,** discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
--new-domain=STRING(/)	Specifies the Avamar server domain where the new dataset should be created.
--new-name=STRING	Specifies a name for the new dataset. This argument is required. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --new-domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset copy** command.

22219   Dataset created.

22288   Dataset does not exist.

dataset delete

The **mccli dataset delete** command permanently deletes a dataset from the Avamar server. You cannot delete a dataset if it is currently assigned to a client or group.

Synopsis

```
mccli dataset delete [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset delete** command.

Table 49 Command options for **mccli dataset delete**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--name=STRING	Specifies the dataset to delete. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show**, discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset delete** command.

22221   Dataset deleted.

22288   Dataset does not exist.

dataset delete-exclude

The **mccli dataset delete-exclude** command permanently deletes exclusion entries from a dataset.

Synopsis

```
mccli dataset delete-exclude [GLOBAL-OPTIONS] {--all |
   --exclude=STRING [--exclude=STRING] ... | --exclude-file=FILE}
   [--domain=STRING(/)] --name=STRING --plugin=INTEGER
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset delete-exclude** command.

Table 50 Command options for **mccli dataset delete-exclude**

Option	Description
--all	Deletes all existing exclusion entries from the dataset. You must supply either --all, --exclude=STRING, or --exclude-file=FILE.
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--exclude=STRING	Specifies the exclusion entry to delete from the dataset. You must supply either --all, --exclude=STRING, or --exclude-file=FILE. Multiple --exclude=STRING arguments are allowed and must be separated by a space. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.
--exclude-file=FILE	Specifies the full path of a text FILE that contains exclusion entries. Each exclusion entry must conform to allowable --exclude=STRING syntax and be on a single line. You must supply either --all, --exclude=STRING, or --exclude-file=FILE.
--name=STRING	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show**, discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
--plugin=INTEGER	Specifies the plug-in ID. This argument is required. Use **mccli plugin show**, discussed on page 152, to return a list of valid plug-in IDs.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset delete-exclude** command.

22220	Dataset modified.
22288	Dataset does not exist.
23009	Invalid plugin specified on the CLI.
23022	Error parsing input file.
23023	Input file could not be found.

dataset delete-include

The **mccli dataset delete-include** command permanently deletes inclusion entries from a dataset.

Synopsis

```
mccli dataset delete-include [GLOBAL-OPTIONS] {--all |
   {--include=STRING [--inlude=STRING] ... | --include-file=FILE}
   [--domain=STRING(/)] --name=STRING --plugin=INTEGER
   [DISPLAY-OPTIONS]
```

Global options

provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset delete-include** command.

Table 51 Command options for **mccli dataset delete-include** (page 1 of 2)

Option	Description
--all	Deletes all inclusion entries from the dataset. You must supply either --all, --include=STRING, or --include-file=FILE.
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--include=STRING	Specifies the inclusion entry to delete from the dataset. You must supply either --all, --include=STRING, or --include-file=FILE. Multiple --include=STRING arguments are allowed and must be separated by a space. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.

Table 51 Command options for **mccli dataset delete-include** (page 2 of 2)

Option	Description
--include-file=FILE	Specifies the full path of a text FILE that contains inclusion entries. Each include entry must conform to allowable --include=STRING syntax and be on a single line. You must supply either --all, --include=STRING, or --include-file=FILE.
--name=STRING	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show,** discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the **--domain** argument is ignored.
--plugin=INTEGER	Specifies the plug-in ID. This argument is required. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset delete-include** command.

22220   Dataset modified.

22288   Dataset does not exist.

23009   Invalid plugin specified on the CLI.

23022   Error parsing input file.

23023   Input file could not be found.

dataset delete-option

The **mccli dataset delete-option** command permanently deletes a plug-in command (option) from a dataset.

Synopsis

```
mccli dataset delete-option [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING {--option=STRING [--option=STRING] ...}
   --plugin=INTEGER [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset delete-option** command.

Table 52 Command options for **mccli dataset delete-option**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--name=STRING	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show**, discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
--option=STRING	Specifies the plug-in command to delete from the dataset. This argument is required. Multiple --option=STRING arguments are allowed and must be separated by a space.
--plugin=INTEGER	Specifies the plug-in ID. This argument is required. Use **mccli plugin show**, discussed on page 152, to return a list of valid plug-in IDs.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset delete-option** command.

22220   Dataset modified.

22288   Dataset does not exist.

23009   Invalid plugin specified on the CLI.

dataset delete-target

The **mccli dataset delete-target** command permanently deletes a target file or folder from an existing dataset.

Synopsis

```
mccli dataset delete-target [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING --plugin=INTEGER {--target=STRING
   [--target-file=FILE] ...} [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset delete-target** command.

Table 53 Command options for **mccli dataset delete-target**

Option	Description
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
`--name=STRING`	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show,** discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
`--plugin=INTEGER`	Specifies the plug-in ID. This argument is required. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.
`--target=STRING`	Specifies the file or folder to delete from the dataset. This argument is required. You must supply either --target=STRING or --target-file=FILE. Multiple --target=STRING arguments are allowed and must be separated by a space. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.
`--target-file=FILE`	Specifies the full path of a text FILE that contains target entries. Each include entry must conform to allowable --target=STRING syntax and be on a single line. You must supply either --target=STRING or --target-file=FILE.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset delete-target** command.

22220   Dataset modified.

22288   Dataset does not exist.

23009   Invalid plugin specified on the CLI.

23022   Error parsing input file.

23023   Input file could not be found.

dataset replace

The **mccli dataset replace** command replaces all exclusion and inclusion entries for a dataset with the entries that you supply on the command line.

Synopsis

```
mccli dataset replace [GLOBAL-OPTIONS] [--domain=STRING(/)]
   [{--exclude=STRING [--exclude=STRING] ... | --exclude-file=FILE}]
   [{--include=STRING [--inlude=STRING] ... | --include-file=FILE}]
   --name=STRING --plugin=INTEGER [DISPLAY-OPTIONS]
```

Command options

The following command options are available for the **mccli dataset replace** command.

Table 54 Command options for **mccli dataset replace**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--exclude=STRING	Specifies the folders or files to exclude from the dataset. Multiple --exclude=STRING arguments are allowed and must be separated by a space. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.
--exclude-file=FILE	Specifies the full path of a text FILE that contains exclusion entries. Each exclusion entry must conform to allowable --exclude=STRING syntax and be on a single line.
--include=STRING	Specifies the folders or files to add back to the dataset after an exclusion entry has excluded them. You must supply either --include=STRING or --include-file=FILE. Multiple --include=STRING arguments are allowed and must be separated by a space. This argument accepts regular expression (regex) pattern matching operators, also known as wildcards. "Pattern matching" on page 31 provides additional information.
--include-file=FILE	Specifies the full path of a text FILE that contains inclusion entries. Each include entry must conform to allowable --include=STRING syntax and be on a single line. You must supply either --include=STRING or --include-file=FILE.
--name=STRING	Specifies the dataset to modify. STRING must be a valid dataset name. This argument is required. Use **mccli dataset show**, discussed on page 106, to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
--plugin=INTEGER	Specifies the plug-in ID. This argument is required. Use **mccli plugin show**, discussed on page 152, to return a list of valid plug-in IDs.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset replace** command.

22220	Dataset modified.
22288	Dataset does not exist.
23009	Invalid plugin specified on the CLI.
23022	Error parsing input file.
23023	Input file could not be found.

dataset show

The **mccli dataset show** command lists all datasets with summary information, or detailed information for a specific dataset.

Synopsis

```
mccli dataset show [GLOBAL-OPTIONS] [--domain=STRING(/)]
   [--name=STRING] [--recursive=BOOLEAN(false)]
   [--verbose=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dataset show** command.

Table 55 Command options for **mccli dataset show**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --name argument.
--name=STRING	Specifies a dataset name. STRING must be a valid dataset name. If not supplied, all datasets are listed with summary information. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the --domain argument is ignored.
--recursive=BOOLEAN(false)	If true, then the command shows all datasets in the domain and any subdomains specified by the --domain=STRING argument. If false or not supplied, then the command only shows datasets in the domain. Subdomains are not examined.
--verbose=BOOLEAN(false)	If true, then detailed dataset information is returned. If false or not supplied, then only dataset names are returned. If you supply --name, then detailed information for the dataset is returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dataset show** command.

22288   Dataset does not exist.

22542   Domain does not exist.

Examples

The following command returns a simple list of all datasets currently in use:

```
mccli dataset show
0,23000,CLI command completed successfully.
command completed successfully

Dataset            Domain
---------------    ------
Unix Dataset       /
Windows Dataset    /
Default Dataset    /
Base Dataset       /
```

The following command returns summary information for all datasets currently in use:

```
mccli dataset show --verbose=true
0,23000,CLI command completed successfully.
command completed successfully
Dataset          Domain Num Targets Num Includes Num Excludes Num Options
---------------  ------ ----------- ------------ ------------ -----------
Unix Dataset     /      4           0            12           0
Windows Dataset  /      1           0            0            0
Default Dataset  /      6           0            0            0
Base Dataset     /      0           0            0            0
```

The following command returns detailed information for a single dataset:

```
mccli dataset show --name='Unix Dataset'
0,23000,CLI command completed successfully.
command completed successfully
Attribute                        Value
---------------------------      ------------
Name                             Unix Dataset
ID                               UNIX:SNAPID
Domain                           /
ReadOnly                         false
ALLDATA                          true
Num Targets                      4
Linux File System Target         ALL
Linux File System Target ID      1001
Solaris File System Target       ALL
Solaris File System Target ID    2001
HP-UX File System Target         ALL
HP-UX File System Target ID      4001
AIX File System Target           ALL
```

```
AIX File System Target ID            5001
Num Includes                         0
Num Excludes                         12
Linux File System Exclude            /tmp
Linux File System Exclude            /var/tmp
Linux File System Exclude            /usr/tmp
Linux File System Exclude            core
Linux File System Exclude            *cache.dat
Linux File System Exclude            *scan.dat
Solaris File System Exclude          /tmp
Solaris File System Exclude          /var/tmp
Solaris File System Exclude          /usr/tmp
Solaris File System Exclude          core
Solaris File System Exclude          *cache.dat
Solaris File System Exclude          *scan.dat
Num Options                          0
Is Link                              false
```

dd

The **mccli dd** resource is used to configure and monitor Data Domain systems. Available commands are:

- ◆ "dd add" on page 108
- ◆ "dd delete" on page 110
- ◆ "dd edit" on page 111
- ◆ "dd show-prop" on page 113
- ◆ "dd show-util" on page 114

dd add

The **mccli dd add** command adds a new Data Domain system to an Avamar server.

Synopsis

```
mccli dd add [GLOBAL-OPTIONS] [--default-storage=Boolean(false)]
   [--force] --max-streams=Integer --name=String --password=String
   --password-confirm=String [--rw-community=String(public)]
   [--snmp-port=Integer(161)] [--trap-port=Integer(162)]
   --user-name=String [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dd add** command.

Table 56 Command options for **mccli dd add**

Option	Description
`--default-storage=Boolean(false)`	If the Data Domain system is a replication target, whether to use the Data Domain system as the default storage for data that replicates from the source Data Domain system.
`--force`	If another Data Domain system is currently designated as default storage on the Avamar server and you want the Data Domain system to be the new default storage, you must supply --force with --default-storage to override the current default storage setting (that is, force the change to occur).
`--max-streams=Integer`	Specifies the maximum number of Data Domain system streams that Avamar uses for backups, restores, and replication.
`--name=String`	Specifies the fully qualified domain name of the Data Domain system.
`--password-confirm=String`	Confirmation of the password.
`--password=String`	Specifies the password for the Data Domain OST account that Avamar should use to access the Data Domain system for backups, restores, and replication.
`--rw-community=String`	Specifies the SNMP community string for Avamar to have read/write access to the Data Domain system.
`--snmp-port=Integer(161)`	Specifies the data port number on the Data Domain system from which to receive and on which to set SNMP objects.
`--trap-port=Integer(163)`	Specifies the data port number on the Avamar server to which SNMP trap messages are sent.
`--user-name=String`	Specifies the username of the Data Domain OST account that Avamar should use to access the Data Domain system for backups, restores, and replication.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dd add** command.

30934	Added Data Domain system.
30937	Failed to add Data Domain system.
30940	Data Domain system already exists.
30941	Failed to connect to Data Domain system.
30942	Unsupported DDOS version.
30948	The minimum number of allowed streams was exceeded.
30949	The maximum number of allowed streams was exceeded.
30953	Added Data Domain system but unable to retrieve the Data Domain system ssh key file.
30958	The DDBoost user is not set.
30959	The OST is disabled.
30960	DDBoost user is disabled.
30961	Failed to authenticate with ssh key file.
30962	Unable to add ssh public key for user.
30963	User does not exist.
30964	User is disabled.
30965	The user is not an admin user.
31004	Data Domain system added but the hostname may not be valid.

dd delete

The **mccli dd delete** command deletes a Data Domain system from an Avamar server.

Synopsis

```
mccli dd delete [GLOBAL-OPTIONS] --name=String
    [--force=Boolean(false)] [DISPLAY-OPTIONS]
```

Global options

provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dd delete** command.

Table 57 Command options for **mccli dd delete**

Option	Description
--name=String	Specifies the fully qualified domain name of the Data Domain system to delete.
--force=Boolean(false)	Whether to force the deletion of the Data Domain system even if the system cannot be reached, has backups or checkpoints, or is the default replication storage system.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dd delete** command.

30936   Deleted Data Domain system.

30939   Failed to delete Data Domain system.

30941   Failed to connect to Data Domain system.

30943   Data Domain system does not exist.

dd edit

The **mccli dd edit** command enables you to edit the Data Domain system configuration.

Synopsis

```
mccli dd edit [GLOBAL-OPTIONS] [--default-storage=Boolean(false)]
    [--force] [--max-streams=Integer] --name=String [--new-name=String]
    [--password=String] [--password-confirm=String]
    [--rw-community=String(public)] [--snmp-port=Integer(161)]
    [--trap-port=Integer(162)] [--user-name=String] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dd edit** command.

Table 58 Command options for **mccli dd edit** (page 1 of 2)

Option	Description
--default-storage=Boolean(false)	If the Data Domain system is a replication target, whether to use the Data Domain system as the default storage for data that replicates from the source Data Domain system.
--force	If another Data Domain system is currently designated as default replication storage on the Avamar server and you want to make the Data Domain system the new default storage, you must supply --force with --default-storage to override the current default storage setting (that is, force the change to occur).
--max-streams=Integer	Specifies the maximum number of Data Domain system streams that Avamar uses for backups, restores, and replication.
--name=String	Specifies the fully qualified domain name of the Data Domain system. This parameter is read-only.

Table 58 Command options for **mccli dd edit** (page 2 of 2)

Option	Description
`--new-name=String`	If renaming a Data Domain system, specifies the new fully qualified domain name of the Data Domain system.
`--password=String`	Specifies the password for the Data Domain OST account that Avamar should use to access the Data Domain system for backups, restores, and replication.
`--password-confirm=String`	Confirmation of the password.
`--rw-community=String`	Specifies the SNMP community string for Avamar to have read/write access to the Data Domain system.
`--snmp-port=Integer(161)`	Specifies the data port number on the Data Domain system from which to receive and on which to set SNMP objects.
`--trap-port=Integer(163)`	Specifies the data port number on the Avamar server to which SNMP trap messages are sent.
`--user-name=String`	Specifies the username of the Data Domain OST account that Avamar should use to access the Data Domain system for backups, restores, and replication.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dd edit** command.

30935   Updated Data Domain system in persitent store.

30938   Failed to update Data Domain system in persitent store.

30941   Failed to connect to Data Domain system.

30943   Data Domain system does not exist.

30948   The minimum number of allowed streams was exceeded.

30949   The maximum number of allowed streams was exceeded.

30953   Updated Data Domain system in persitent store but unable to retrieve the Data Domain system ssh key file.

30958   The DDBoost user is not set.

30959   The OST is disabled.

30960   DDBoost user is disabled.

30961   Failed to authenticate with ssh key file.

30962   Unable to add ssh public key for user.

30963   User does not exist.

30964   User is disabled.

30965   The user is not an admin user.

31005   Data Domain system added but the hostname may not be valid.

dd show-prop

The **mccli dd show-prop** command shows Data Domain system properties.

Synopsis

mccli dd show-prop [GLOBAL-OPTIONS] [--name=String] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command option is available for the **mccli dd show-prop** command.

Table 59 Command options for **mccli dd show-prop**

Option	Description
--name=String	Specifies the fully qualified Data Domain system name. If not supplied, a summary report of properties for all known Data Domain systems is returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event code is associated with the **mccli dd show-prop** command.

30943   Data Domain system does not exist.

Example

```
mccli dd show-prop datadomain1.example.com
0,23000,CLI command completed successfully.
Attribute                               Value
--------------------------------------  -----------------------
Hostname                                datadomain1.example.com
Total Capacity (post-comp size)         1526.0 GiB
Server Utilization (post-comp use%)     16.0%
Bytes Protected                         3828.0 GiB
File System Available (post-comp avail) 1278 GiB
File System Used (post-comp used)       247 GiB
User Name                               avamar
Default Replication Storage System      No
Maximum Streams                         50
DDOS Version                            4.9.1.0-203537
Serial Number                           9F40339887
Model Number                            DD880
Monitoring Status                       OK
```

dd show-util

The **mccli dd show-util** command shows Data Domain system utilization statistics.

Synopsis

mccli dd show-util [GLOBAL-OPTIONS] [--name=String] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli dd show-util** command.

Table 60 Command options for **mccli dd show-util**

Option	Description
--name=String	Specifies the fully qualified Data Domain system name.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli dd show-util** command.

30943   Data Domain system does not exist.

domain

The **mccli domain** resource is used to create and manage domains and subdomains on the Avamar server. Available commands are:

domain add

The **mccli domain add** command creates a new domain or subdomain on the Avamar server.

Synopsis

```
mccli domain add [GLOBAL-OPTIONS] [--contact=STRING]
   [--domain=STRING(/)] [--email=STRING] [--location=STRING]
   --name=STRING [--phone=STRING] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli domain add** command.

Table 61 Command options for **mccli domain add**

Option	Description
--contact=STRING	Specifies contact information for the new domain.
--domain=STRING(/)	Specifies the parent domain for the new domain.
--email=STRING	Specifies the email address for the primary contact (responsible party).
--location=STRING	Specifies location information for the new domain.
--name=STRING	Specifies the name of the new domain. STRING must be a valid domain name. This argument is required.
--phone=STRING	Specifies telephone information for the primary contact (responsible party).

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli domain add** command.

22526   Failed folder add.

22527   Domain added.

22540   Invalid name character.

22541   Domain already exists.

22558   A domain or client with this name already exists.

domain delete

The **mccli domain delete** command permanently deletes a domain from the Avamar server. Deleting a domain deletes the domain and all subdomains, clients, and backups stored in that domain.

Synopsis

```
mccli domain delete [GLOBAL-OPTIONS] [--domain=STRING] --name=STRING
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli domain delete** command.

Table 62 Command options for **mccli domain delete**

Option	Description
--domain=STRING	Specifies the parent domain. Default parent domain is root (/).
--name=STRING	Specifies the domain to delete. STRING must be a valid domain name. This argument is required. Use **mccli domain show**, discussed on page 118, to return a list of valid domain names. If you supply a fully qualified domain name (for example, /domain/subdomain), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli domain delete** command.

22507   Failed domain deletion.

22513   Domain removed.

22542   Domain does not exist.

22543   Domain has groups.

22544   Domain has datasets

22545   Domain has event profiles.

22627   Domain has schedules.

22628   Domain has retentions.

domain edit

The **mccli domain edit** command edits the properties for a domain.

Synopsis

```
mccli domain edit [GLOBAL-OPTIONS] [--contact=STRING]
   [--domain=STRING(/)] [--email=STRING] [--location=STRING]
   --name=STRING [--phone=STRING] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli domain edit** command.

Table 63 Command options for **mccli domain edit**

Option	Description
`--contact=STRING`	Specifies contact information for the domain.
`--domain=STRING(/)`	Specifies the parent domain.
`--email=STRING`	Specifies the email address for the primary contact (responsible party).
`--location=STRING`	Specifies location information for the domain.
`--name=STRING`	Specifies the domain to edit. STRING must be a valid domain name. This argument is required. Use **mccli domain show,** discussed on page 118, to return a list of valid domain names.
`--phone=STRING`	Specifies telephone information for the primary contact (responsible party).

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli domain edit** command.

22519   Domain modified.

22542   Domain does not exist.

domain show

The **mccli domain show** command displays properties for a domain.

Synopsis

```
mccli domain show [GLOBAL-OPTIONS] [--domain=STRING(/)]
   [--name=STRING] [--recursive=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli domain show** command.

Table 64 Command options for **mccli domain show**

Option	Description
--domain=STRING(/)	Specifies the parent domain under which domains and subdomains are listed.
--name=STRING	Specifies the name of the domain for which to show information. STRING must be a valid domain name.
--recursive=BOOLEAN(false)	If true, then the command examines all domains in the domain and any subdomains. If false or not supplied, then the command only examines the domain. Subdomains are not examined.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event code is associated with the **mccli domain show** command.

22542   Domain does not exist.

dump

The **mccli dump** resource is used to dump various DPNProxyService caches for troubleshooting purposes. Available commands are:

dump clientcache

The **dump clientcache** command is used to dump the DPNProxyService client cache for troubleshooting purposes.

> **NOTICE**

The **dump clientcache** command is strictly reserved for EMC internal use only.

Synopsis

```
mccli dump clientcache [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

dump domaincache

The **dump clientcache** command is used to dump the DPNProxyService domain cache for troubleshooting purposes.

> **NOTICE**

The **dump domaincache** command is strictly reserved for EMC internal use only.

Synopsis

```
mccli dump domaincache [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

dump jobcache

The **dump jobcache** command is used to dump the DPNProxyService client job cache for troubleshooting purposes.

> **NOTICE**

The **dump jobcache** command is strictly reserved for EMC internal use only.

Synopsis

```
mccli dump jobcache [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

event

The **mccli event** resource is used to access and manage event codes on the Avamar server. Available commands are:

- "event ack" on page 120
- "event get-info" on page 123
- "event show" on page 124

event ack

The **mccli event ack** command acknowledges events.

Synopsis

```
mccli event ack [GLOBAL-OPTIONS] [--after=STRING]
   [--all=BOOLEAN(false)] [--before=STRING] [--category=STRING]
   [--domain=STRING] [--exclude=STRING] [--id=INTEGER]
   [--include=STRING] [--range=STRING] [--severity=STRING]
   [--source=STRING] [--type=STRING] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli event ack** command.

Table 65 Command options for **mccli event ack** (page 1 of 2)

Option	Description
`--after=STRING`	Acknowledges events after a date. STRING must be in the format of: `YYYY-MM-DD HH:mm:ss` where YYYY is the four-digit year, MM is the two-digit calendar month, DD is the two-digit day of the month, and HH:mm:ss is a timestamp.
`--all=BOOLEAN(false)`	Acknowledges all events.
`--before=STRING`	Acknowledges events before a date. STRING must be in the format of: `YYYY-MM-DD HH:mm:ss` where YYYY is the four-digit year, MM is the two-digit calendar month, DD is the two-digit day of the month, and HH:mm:ss is a timestamp.
`--category=STRING`	Specifies the category of the events to acknowledge. STRING must be one of the following: • APPLICATION • SECURITY • SYSTEM • USER
`--domain=STRING`	Specifies the domain of the events to acknowledge.
`--exclude=STRING`	Comma-separated list of event codes to exclude from acknowledgement. The comma-separated list cannot contain spaces. The --include and --exclude options are mutually exclusive.
`--id=INTEGER`	Unique identifier of the event to acknowledge.
`--include=STRING`	Comma-separated list of event codes to include for acknowledgement. The comma-separated list cannot contain spaces. The --include and --exclude options are mutually exclusive.
`--range=STRING`	Specifies a range of event IDs for the events to acknowledge. STRING must be in the format of: `ID-1:ID-2` where ID-1 and ID-2 are the lower and upper limits of the range, respectively.

Table 65 Command options for **mccli event ack** (page 2 of 2)

Option	Description
`--severity=STRING`	Specifies the severity of the events to acknowledge. STRING must be one of the following: • NODE • NODE_FATAL • OK • PROCESS • PROCESS_FATAL • SYSTEM_FATAL • USER • USER_FATAL
`--source=STRING`	Specifies a source storage location for the events to acknowledge, either: • avamar • dd If you specify avamar, then the command acknowledges events for the Avamar server. If you specify dd, then the command acknowledges events for all configured Data Domain systems.
`--type=STRING`	Specifies the type of event to acknowledge. STRING must be one of the following: • AUDIT • DEBUG • ERROR • INFORMATION • INTERNAL • WARNING

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event code is associated with the **mccli event ack** command.

23022   Event id does not exist.

Notes

Consider the following notes for the **mccli event ack** command:

◆ This command requires either the --all option or some other filtering criteria specified using --exclude and --include options.

◆ Output is a count of the events acknowledged.

◆ The --include and --exclude options are mutually exclusive.

Examples

The following command acknowledges an event with a specific event code (ID 148710):

```
mccli event ack --id=148710
0,23000,CLI command completed successfully.
Attribute       Value
-----------     -----
events-acked    1
```

event get-info

The **mccli event get-info** command returns detailed information for a specific event code.

Synopsis

```
mccli event get-info [GLOBAL-OPTIONS] --code=INTEGER
   [--description=BOOLEAN(false)] [--remedy=BOOLEAN(false)]
   [--summary=BOOLEAN(true)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli event get-info** command.

Table 66 Command options for **mccli event get-info**

Option	Description
--code=INTEGER	Specifies the event code number. This argument is required.
--description=BOOLEAN(false)	If true, then other properties are also returned, including the remedy and summary properties. If false or not supplied, then only the code is returned.
--remedy=BOOLEAN(false)	If true, then event remedy information is returned. If false or not supplied, then the event remedy is not returned.
--summary=BOOLEAN(true)	If true or not supplied, then event summary information is returned. If false, then the event summary is not returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

event show

The **mccli event show** command returns event occurrence details.

Synopsis

```
mccli event show [GLOBAL-OPTIONS] [--after=STRING] [--before=STRING]
    [--category=STRING] [--domain=STRING] [--exclude=STRING]
    [--id=INTEGER] [--include=STRING] [--severity=STRING]
    [--source=STRING] [--type=STRING] [--unack=BOOLEAN(false)]
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli event show** command.

Table 67 Command options for **mccli event show** (page 1 of 2)

Option	Description
--after=STRING	Returns events after a date. STRING must be in the format of: YYYY-MM-DD HH:mm:ss where YYYY is the four-digit year, MM is the two-digit calendar month, DD is the two-digit day of the month, and HH:mm:ss is a timestamp.
--before=STRING	Returns events before a date. STRING must be in the format of: YYYY-MM-DD HH:mm:ss where YYYY is the four-digit year, MM is the two-digit calendar month, DD is the two-digit day of the month, and HH:mm:ss is a timestamp.
--category=STRING	Specifies the category of the event for which to show information. STRING must be one of the following: • APPLICATION • SECURITY • SYSTEM • USER
--domain=STRING	Specifies the domain of the event for which to show information.
--exclude=STRING	Comma-separated list of event codes to exclude from the events that are shown. The comma-separated list cannot contain spaces. The --include and --exclude options are mutually exclusive.
--id=INTEGER	Unique identifier of the event for which to show information.
--include=STRING	Comma-separated list of event codes to include in the events that are shown. The comma-separated list cannot contain spaces. The --include and --exclude options are mutually exclusive.

Table 67 Command options for **mccli event show** (page 2 of 2)

Option	Description
`--severity=STRING`	Specifies the severity of the event for which to show information. STRING must be one of the following: • NODE • NODE_FATAL • OK • PROCESS • PROCESS_FATAL • SYSTEM_FATAL • USER • USER_FATAL
`--source=STRING`	Specifies a source storage location of the event for which to show information, either: • avamar • dd If you specify avamar, then the command shows events for the Avamar server. If you specify dd, then the command shows events for all configured Data Domain systems.
`--type=STRING`	Specifies the event types for which to show information. STRING must be one of the following: • AUDIT • DEBUG • ERROR • INFORMATION • INTERNAL • WARNING You can supply more than one --type option on a single command line.
`--unack=BOOLEAN(false)`	Shows only information for unacknowledged events.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli event show** command.

23022   Event id does not exist.

Notes

Consider the following notes for the **mccli event show** command:

◆   If multiple events are displayed, sorting is by date in descending order.

◆   The --include and --exclude options are mutually exclusive.

Examples

The following command returns output for multiple event occurrences:

```
mccli event show
0,23000,CLI command completed successfully.
command completed successfully

ID     Date                   Type         Code Category Severity
------ ---------------------- ------------ ---- -------- --------
149897 2011-04-19 14:19:45 PDT INFORMATION 578  SYSTEM   OK


Domain Summary
------ -------------------------------------------
/      hfscheckresults::merge, in checkpoint cp.20110419140741, this
           ((start 1176995040, nodestart 1176995040, nodefinish 0,
           valid false, totalerrors 0)) tried to merge with ((start 0,
           nodestart 0, nodefinish 0, valid false, totalerrors 0)) but
           the uniquestarttime fields differ
```

NOTICE

The Domain and Summary columns appear in line with the ID, Date, Type, Code, Category, and Severity columns in the command output, but are separated for space considerations in this guide.

The following command returns detailed information a specific event occurrence (ID):

```
mccli event show --id=149897
0,23000,CLI command completed successfully.
command completed successfully

Attribute   Value
----------  --------------------------------------------------------
ID          149897
Date        2011-04-19 14:19:45 PDT
Type        INFORMATION
Code        578
Category    SYSTEM
Severity    OK
Domain      /
Summary     hfscheckresults::merge, in checkpoint cp.20110419140741,
                this ((start 1176995040, nodestart 1176995040,
                nodefinish 0, valid false, totalerrors 0)) tried to
                merge with ((start 0, nodestart 0, nodefinish 0, valid
                false, totalerrors 0)) but the uniquestarttime fields
                differ
SW Source   DPN:Unknown
For Whom    N/A
HW Source   node-10-0-54-249.example.com
Description N/A
```

```
Remedy      N/A
Notes       N/A
Data        <data><entry key="code" type="text" value="0578"
              version=""/><entry key="type" type="text" value="INFO"
              version=""/><entry key="time" type="text"
              value="21:19:45.93935" version=""/><entry
              key="message" type="text"
              value="hfscheckresults::merge, in checkpoint
              cp.20110419140741, this ((start 1176995040, nodestart
              1176995040, nodefinish 0, valid false, totalerrors 0))
              tried to merge with ((start 0, nodestart 0, nodefinish
              0, valid false, totalerrors 0)) but the
              uniquestarttime fields differ" version=""/><entry
              key="date" type="text" value="2011/04/19"
              version=""/><entry key="thread" type="text"
              value="srvm-14070642#srv:265" version=""/></data>
```

The following command shows when the maximum number of events displayed is
reached:

mccli event show
```
INFO: The maximum number of events displayed limit reached. The maximum
  is 5000.
0,23000,CLI command completed successfully.
 ID     Date                    Type        Code Category Severity
------ ----------------------- ----------- ---- -------- --------
149997 2011-04-19 14:44:46 PDT INFORMATION 578  SYSTEM   OK


Domain Summary
------ --------------------------------------------
/      hfscheckresults::merge, in checkpoint cp.20110419140741, this
         ((start 1176995040, nodestart 1176995040, nodefinish 0,
         valid false, totalerrors 0)) tried to merge with ((start 0,
         nodestart 0, nodefinish 0, valid false, totalerrors 0)) but
         the uniquestarttime fields differ
```

NOTICE

The Domain and Summary columns appear in line with the ID, Date, Type, Code, Category,
and Severity columns in the command output, but are separated for space considerations
in this guide.

group

The **mccli group** resource is used to create and manage groups and group policy on the Avamar server. Available commands are:

Enumerating group resources

Several **mccli group** commands require precise case-sensitive names (STRING values) for the following client resources:

- Datasets
- Schedules
- Retention policies
- Plug-ins

The preferred way to view a list of valid datasets, schedules, retention policies, and plug-ins is to use the following **mccli** commands, respectively:

- **dataset show,** discussed on page 106
- **schedule show,** discussed on page 167
- **retention show,** discussed on page 158
- **plugin show,** discussed on page 152

You can also view a list of valid datasets, retention policies and plug-ins using the Avamar Administrator **Tools › Manage All Datasets, Manage All Retention Policies,** and **Manage Agents & Plug-ins** features, respectively.

group add

The **mccli group add** command creates a group.

Synopsis

```
mccli group add [GLOBAL-OPTIONS] --dataset=STRING(Default Dataset)
   [--dataset-domain=STRING(/)] [--domain=STRING(/)]
   --enabled=BOOLEAN(false) [--encryption=STRING] --name=STRING
   --retention=STRING(Default Retention)[--retention-domain=STRING(/)]
   --schedule=STRING(Default Schedule) [--schedule-domain=STRING(/)]
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group add** command.

Table 68 Command options for **mccli group add** (page 1 of 2)

Option	Description
`--dataset-domain=STRING(/)`	Specifies the Avamar server domain that contains the dataset specified by the --dataset argument.
`--dataset=STRING(Default Dataset)`	Specifies the dataset to assign to the group. This argument is required.
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the group specified by the --name argument.
`--enabled=BOOLEAN(false)`	If true, then the group is eligible to immediately participate in scheduled backup activities. This argument is required.
`--encryption=STRING`	Specifies the encryption method that the client should use when performing on-demand backups and restores, or when the group encryption method is overridden. STRING must be one of the following: • High — Strongest available encryption setting for that specific client platform. • Medium — Medium strength encryption. • None — No encryption. **Note:** The exact encryption technology and bit strength used for any given client/server connection depends on a number of factors, including the client platform and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.

Table 68 Command options for **mccli group add** (page 2 of 2)

Option	Description
`--name=STRING`	Specifies a name for the new group. This argument is required. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.
`--retention=STRING(Default Retention)`	Specifies the retention policy to assign to the group. This argument is required.
`--retention-domain=STRING(/)`	Specifies the Avamar server domain that contains the retention policy specified by the --retention argument.
`--schedule=STRING(Default Schedule)`	Specifies the schedule to assign to the group. This argument is required.
`--schedule-domain=STRING(/)`	Specifies the Avamar server domain that contains the schedule specified by the --schedule argument.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group add** command.

22207 New group created.

22233 Group already exists.

22235 Group add failed.

23003 Invalid dataset name specified on the CLI.

23004 Invalid schedule name specified on the CLI.

23005 Invalid retention policy name specified on the CLI.

23012 Invalid encryption method specified on the CLI.

31002 Group add or update failed due to a policy hierarchical management violation.

group add-client

The **mccli group add-client** command adds a client to a group.

Synopsis

```
mccli group add-client [GLOBAL-OPTIONS] [--client-domain=STRING(/)]
   --client-name=STRING [--domain=STRING(/)] --name=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group add-client** command.

Table 69 Command options for **mccli group add-client**

Option	Description
`--client-domain=STRING(/)`	Specifies the domain of the client to add to the group.
`--client-name=STRING`	Specifies the client to add to the group. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /MyDomain/MyClient), then the --client-domain argument is ignored.
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the group specified by the --name argument.
`--name=STRING`	Specifies the group to which to add the client. STRING must be a valid group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group add-client** command.

22234	Group does not exist.
22236	Client does not exist.
22242	Client is already a member of this group.
22243	Client group membership successfully updated.
22269	Unable to add a client to a group.
22358	Unable to add a client to group due to incompatibility.

group add-proxy

The **mccli group add-proxy** command adds the specified Avamar image proxy client to the specified vCenter group.

Synopsis

```
mccli group add-proxy GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [--proxy-domain=STRING(/)] --proxy-name=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group add-proxy** command.

Table 70 Command options for **mccli group add-proxy**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the vCenter group specified by the --name argument.
--name=STRING	Specifies the group to which to add the client. STRING must be a valid vCenter group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.
--proxy-domain=STRING(/)	Specifies the Avamar server domain that contains the Avamar image proxy client specified by the --proxy-name argument.
--proxy-name=STRING	Specifies which Avamar image proxy client to add. STRING must be a valid Avamar image proxy client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --proxy-domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group add-proxy** command.

22236   Client does not exist.

22234   Group does not exist.

24001   Failed to update proxy client mappings of a group.

24002   Proxy client mappings of a group successfully updated.

Examples

The following example adds proxy client backupproxy225 to the Test1 group:

```
mccli group add-proxy --domain=/10.31.183.228 --name=Test1
   --proxy-domain=/clients --proxy-name=backupproxy225
0,24002,Proxy client mappings of a group successfully updated.
Attribute    Value
---------    --------------------
group        /10.31.183.228/Test1
```

group backup

The **mccli group backup** command initiates an on-demand group backup.

Synopsis

```
mccli group backup [GLOBAL-OPTIONS] [--domain=STRING(/)] --name=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group backup** command.

Table 71 Command options for **mccli group backup**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the group specified by the --name argument.
--name=STRING	Specifies the group to back up. STRING must be a valid group name. This argument is required. Use **mccli group show**, discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group backup** command.

22226	Group disabled.
22227	Group does not contain any clients.
22228	A client was not backed up because it is disabled, retired or one or more of its plug-ins has backups disabled.
22234	Group does not exist.
22301	Scheduled group backups initiated for all clients.
22311	Scheduled group backups failed to start.

group copy

The **mccli group copy** command copies an existing group, creating a new group. The new group inherits all the properties and settings of the original group except for client members. No clients are assigned to the new group.

> **NOTICE**
>
> You must copy groups within the same domain. You cannot copy a group to another domain under any circumstances.

Synopsis

```
mccli group copy [GLOBAL-OPTIONS] [--domain=STRING(/)] --name=STRING
   --new-name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group copy** command.

Table 72 Command options for **mccli group copy**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the group specified by the --name argument.
--name=STRING	Specifies the group to copy. STRING must be a valid group name. This argument is required. Use **mccli group show**, discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.
--new-name=STRING	Specifies the name for the new group. This argument is required.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group copy** command.

22207	New group created.
22233	Group already exists.
22234	Group does not exist.
22235	Group add failed.
22540	Invalid name character.

group delete

The **mccli group delete** command permanently deletes a group from the Avamar server.

A client must always be a member of at least one group. Therefore, if the group that you are deleting contains any clients that are not also members of at least one other group, then you must move those clients to other groups using the **mccli group move-client** command, discussed on , before you delete the group.

Synopsis

```
mccli group delete [GLOBAL-OPTIONS] [--domain=STRING(/)] --name=STRING
    [DISPLAY-OPTIONS]
```

Global options

provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group delete** command.

Table 73 Command options for **mccli group delete**

Option	Description
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the group specified by the --name argument.
`--name=STRING`	Specifies the group to delete. STRING must be a valid group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group delete** command.

22209	Group successfully deleted.
22234	Group does not exist.
22247	Cannot delete the group because the client members are not present in other groups.

group edit

The **mccli group edit** command edits properties for a group.

There are no default settings for the **mccli group edit** command. If you enter the command but do not supply options and values on the command line, then no changes are made to the group.

Synopsis

```
mccli group edit [GLOBAL-OPTIONS] --dataset=STRING
   [--dataset-domain=STRING(/)] [--domain=STRING] --enabled=BOOLEAN
   [--encryption=STRING] --name=STRING --retention=STRING
   [--retention-domain=STRING(/)] --schedule=STRING
   [--schedule-domain=STRING(/)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group edit** command.

Table 74 Command options for **mccli group edit** (page 1 of 2)

Option	Description
--dataset=STRING	Specifies the dataset assigned to the group. This argument is required.
--dataset-domain=STRING(/)	Specifies the Avamar server domain that contains the dataset specified by the --dataset argument.
--domain=STRING	Specifies the Avamar server domain that contains the group specified by the --name argument.
--enabled=BOOLEAN	If true, then the group is eligible to immediately participate in scheduled backup activities. This argument is required.

Table 74 Command options for **mccli group edit** (page 2 of 2)

Option	Description
`--encryption=STRING`	Specifies the encryption method that the client should use when performing on-demand backups and restores, or when the group encryption method is overridden. STRING must be one of the following: • High — Strongest available encryption setting for that specific client platform. • Medium — Medium strength encryption. • None — No encryption. **Note:** The exact encryption technology and bit strength used for any given client/server connection depends on a number of factors, including the client platform and Avamar server version. The *EMC Avamar Product Security Guide* provides additional information.
`--name=STRING`	Specifies the group to edit. STRING must be a valid group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.
`--retention=STRING`	Specifies the retention policy for the group. This argument is required.
`--retention-domain=STRING(/)`	Specifies the Avamar server domain that contains the retention policy specified by the --retention argument.
`--schedule=STRING`	Specifies the schedule for the group. This argument is required.
`--schedule-domain=STRING(/)`	Specifies the Avamar server domain that contains the schedule specified by the --schedule argument.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group edit** command.

22208	Group successfully modified.
22234	Group does not exist.
23003	Invalid dataset name specified on the CLI.
23004	Invalid schedule name specified on the CLI.
23005	Invalid retention policy name specified on the CLI.
23012	Invalid encryption method specified on the CLI.
31002	Group add or update failed due to a policy hierarchical management violation.

group export

The **mccli group export** command exports a group, group settings (that is, name, domain, encryption method, and so forth), and assigned policy objects (dataset, schedule, and retention policy) to an XML file. However, groups exported in this way do not contain a list of client members.

Synopsis

```
mccli group export [GLOBAL-OPTIONS] [--domain=STRING(/)] --file=STRING
    --name=STRING [--force=Boolean(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group export** command.

Table 75 Command options for **mccli group export**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the group specified by the --name argument.
--file=STRING	Specifies the name of the export XML file to create. This option is required.
--force=Boolean(false)	If true, then the option forces the overwrite of an existing XML file.
--name=STRING	Specifies the group to export. This option is required. Use **mccli group show**, discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/In-group), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group export** command.

22234   Group does not exist.

group move-client

The **mccli group move-client** command moves a client from one group to another.

Synopsis

```
mccli group move-client [GLOBAL-OPTIONS] [--client-domain=STRING(/)]
   --client-name=STRING [--domain=STRING(/)] --name=STRING
   [--old-group-domain=STRING(/)] [--old-group-name=STRING(Default
   Group)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group move-client** command.

Table 76 Command options for **mccli group move-client**

Option	Description
--client-domain=STRING(/)	Specifies the domain to which to move the client.
--client-name=STRING	Specifies the client to move. This argument is required.
--domain=STRING(/)	Specifies the Avamar server domain that contains the group specified by the --name argument.
--name=STRING	Specifies the group to which to move the client. STRING must be a valid group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.
--old-group-domain=STRING(/)	Specifies the domain from which to move the client.
--old-group-name=STRING(Default Group)	Specifies the group from which to move the client.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group move-client** command.

22234	Group does not exist.
22236	Client does not exist.
22241	Client is not a member of old group.
22242	Client is already a member of new group.
22243	Client group membership successfully updated.

group remove-client

The **mccli group remove-client** command removes a client from a group.

A client must always be a member of at least one group. Therefore, if the client belongs only to the group from which you want to remove it, then you must first add the client to another group using the **mccli group add-client** command, discussed on page 130.

Synopsis

```
mccli group remove-client [GLOBAL-OPTIONS] [--client-domain=STRING(/)]
   --client-name=STRING [--domain=STRING(/)] --name=STRING
   [DISPLAY-OPTIONS]
```

Global options

provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group remove-client** command.

Table 77 Command options for **mccli group remove-client**

Option	Description
`--client-domain=STRING(/)`	Specifies the domain from which to remove the client.
`--client-name=STRING`	Specifies the client to remove. This argument is required.
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the group specified by the --name argument.
`--name=STRING`	Specifies the group from which to remove the client. STRING must be a valid group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group remove-client** command.

22234	Group does not exist.
22236	Client does not exist.
22241	Client is not a member of this group.
22243	Client group membership successfully updated.
22246	Client is not a member of any other group.
22270	Unable to remove a client from a group.

group remove-proxy

The **mccli group remove-proxy** command removes the specified Avamar image proxy client from the specified vCenter group.

Synopsis

```
mccli group remove-proxy GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [--proxy-domain=STRING(/)] --proxy-name=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group remove-proxy** command.

Table 78 Command options for **mccli group remove-proxy**

Option	Description
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the vCenter group specified by the --name argument.
`--name=STRING`	Specifies the group from which to remove the client. STRING must be a valid vCenter group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.
`--proxy-domain=STRING(/)`	Specifies the Avamar server domain that contains the Avamar image proxy client specified by the --proxy-name argument.
`--proxy-name=STRING`	Specifies which Avamar image proxy client to remove. STRING must be a valid Avamar image proxy client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --proxy-domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group remove-proxy** command.

22236	Client does not exist.
22234	Group does not exist.
24001	Failed to update proxy client mappings of a group.
24004	Proxy client mappings of a group successfully updated.

Examples

The following example removes proxy client backupproxy225 from theTest1 group:

```
mccli group remove-proxy --domain=/10.31.183.228 --name=Test1
   --proxy-domain=/clients --proxy-name=backupproxy225
0,24002,Proxy client mappings of a group successfully updated.
Attribute    Value
---------    --------------------
group        /10.31.183.228/Test1
```

group show

The **mccli group show** command returns information for all groups.

Synopsis

```
mccli group show [GLOBAL-OPTIONS] [--clients=BOOLEAN(false)]
    [--domain=STRING(/)] [--name=STRING] [--recursive=BOOLEAN(false)]
    [--verbose=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group show** command.

Table 79 Command options for **mccli group show**

Option	Description
--clients=BOOLEAN(false)	If true, then all clients that belong to the group are listed.
--domain=STRING(/)	Specifies the Avamar server domain that contains the group specified by the --name argument.
--name=STRING	Specifies the name of the group for which to show information. If not supplied, all groups in the domain are listed. STRING must be a valid group name. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.
--recursive=BOOLEAN(false)	If true, then the command shows all groups in the domain and any subdomains specified by the --domain=STRING argument. If false or not supplied, then the command only shows groups in the domain. Subdomains are not displayed.
--verbose=BOOLEAN(false)	If true, then detailed information is returned. If false or not supplied, then only the group names are returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group show** command.

22542   Domain does not exist.

Examples

The following command returns a simple list of all groups in use:

```
mccli group show
0,23000,CLI command completed successfully.
command completed successfully
 Group           Domain
 -------------   ------
 Default Group   /
```

The following command returns detailed information for all groups currently in use:

```
mccli group show --verbose=true
0,23000,CLI command completed successfully.
command completed successfully
 Group   Domain Enabled Dataset  Schedule Retention Policy  Encryption Method
 ------- ------ ------- -------- -------- ----------------  -----------------
 GroupA  /      false   Data2    Sched2   Default Retention High
```

The following command returns detailed information for a single group:

```
mccli group show --name='Default Group'
0,23000,CLI command completed successfully.
command completed successfully
 Attribute         Value
 ----------------- -----------------
 Name              Default Group
 Domain            /
 Enabled           false
 Dataset           Default Dataset
 Schedule          Default Schedule
 Retention Policy  Default Retention
 Encryption Method High
```

group show-client-members

The **mccli group show-client-members** command returns a list of all clients that belong to a group.

Synopsis

```
mccli group show-client-members [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group show-client-members** command.

Table 80 Command options for **mccli group show-client-members**

Option	Description
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the group specified by the --name argument.
`--name=STRING`	Specifies the group from which to show client members. STRING must be a valid group name. This argument is required.
	Use **mccli group show,** discussed on page 143, to return a list of valid group names.
	If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event code is associated with the **mccli group show-client-members** command.

22234   Group does not exist.

Examples

The following command returns a simple list of all clients in the Default Group:

```
mccli group show-client-members --name='Default Group'
0,23000,CLI command completed successfully.
command completed successfully
Group           Client
-------------   -----------------
Default Group   /clients/MyClient
```

group snapup

The **mccli group snapup** command initiates an on-demand group backup.

> **NOTICE**
>
> This command has been deprecated in favor of **mccli group backup,** which is discussed on page 133. Whenever possible, use **mccli group backup** instead of **mccli group snapup.**

Synopsis

```
mccli group snapup [GLOBAL-OPTIONS] [--domain=STRING(/)] --name=STRING
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli group snapup** command.

Table 81 Command options for **mccli group snapup**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the group specified by the --name argument.
--name=STRING	Specifies the group to back up. STRING must be a valid group name. This argument is required. Use **mccli group show,** discussed on page 143, to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli group snapup** command.

22226	Group disabled.
22227	Group does not contain any clients.
22228	A client was not backed up because it is disabled, retired or one or more of its plug-ins has backups disabled.
22234	Group does not exist.
22301	Scheduled group backups initiated for all clients.
22311	Scheduled group backups failed to start.

help

The **mccli help** resource includes a single command, **mccli help,** which returns online help for a resource and then exits.

help

The **mccli help** command returns online help for a resource and then exits.

Synopsis

```
mccli help {activity | backup | checkpoint | client | dataset | event |
    group | mcs | retention | schedule | snapup | version}
```

mcs

The **mccli mcs** resource is used to view status and control various adminstrator server functions. Available commands are:

- ◆ "mcs import" on page 147
- ◆ "mcs list" on page 148
- ◆ "mcs reboot-proxy" on page 149
- ◆ "mcs resume-scheduler" on page 150
- ◆ "mcs scheduler-status" on page 150
- ◆ "mcs stop" on page 151
- ◆ "mcs suspend-scheduler" on page 151
- ◆ "mcs waitforflushcomplete" on page 152

mcs import

The **mccli mcs import** command accepts an XML file created by the **group export** command, which is discussed on page 138, and imports the group and its policy objects (that is, dataset, schedule, and retention policy) into the target Avamar server. The group name remains the same, but you can specify a new target domain.

Synopsis

```
mccli mcs import [GLOBAL-OPTIONS] --file=STRING
    [--force=Boolean(false)] --target-domain=STRING [--prefix=STRING]
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli mcs import** command.

Table 82 Command options for **mccli mcs import**

Option	Description
`--file=STRING`	Specifies the name of the import XML file. This option is required.
`--force=Boolean(false)`	XML files created by the export command are digitally signed. If an XML file is edited or manually created, then digital signature verification fails during import. If true, then the XML import is forced without validation of the digital signature. If false, then an error is returned, and the file is not imported.
`--prefix=STRING`	Specifies a prefix string that is affixed to the beginning of the imported object names. This is done to avoid conflicts and distinguish imported objects from existing ones that might otherwise have similar names.
`--target-domain=STRING`	Specifies the domain on the target server for the group and its policy objects (that is, dataset, schedule, and retention policy). Original domain hierarchy for the imported objects is not preserved. In other words, all imported objects reside at the top-most level of the domain. The target domain must already exist on the target server. If not, then the command returns an error. To avoid conflicts, if a policy object with the same name already exists in the target domain, then _1, _2, _3 and so forth is appended to the name of the newly imported policy object.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event code is associated with the **mccli mcs import** command.

22540   Invalid name character.

mcs list

The **mcs list** command accepts an XML file created by the **group export** command, which is discussed on page 138, and lists all groups and group policy objects (that is, dataset, schedule, and retention policy) in that file.

Synopsis

mccli mcs list [GLOBAL-OPTIONS] --file=STRING [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command option is available for the **mccli mcs list** command.

Table 83 Command options for **mccli mcs list**

Option	Description
--file=STRING	Specifies the name of the import XML file. This option is required.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

mcs reboot-proxy

The **mccli mcs reboot-proxy** command reboots VMware proxy clients.

Synopsis

```
mcs reboot-proxy [GLOBAL-OPTIONS] [--all=Boolean(false) |
   --name=STRING] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli mcs reboot-proxy** command.

Table 84 Command options for **mccli mcs reboot-proxy**

Option	Description
--all=Boolean(false)	If supplied, reboot all known Avamar proxy clients.
--name=STRING	Specifies the proxies to reboot. STRING must be a fully qualified Avamar proxy client name. You can supply multiple --name options on the same command line.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

mcs resume-scheduler

The **mccli mcs resume-scheduler** command turns on the MCS scheduler so that scheduled operations occur at the scheduled times.

Synopsis

```
mccli mcs resume-scheduler [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli mcs resume-scheduler** command.

22308   Scheduler successfully resumed.

22310   Change of the scheduler status to suspended or resume failed.

22631   Server has reached the capacity health check limit

mcs scheduler-status

The **mccli mcs scheduler-status** command returns the status of the MCS scheduler.

Synopsis

```
mccli mcs scheduler-status [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Example

```
mccli mcs scheduler-status
0,23000,CLI command completed successfully.
command completed successfully
Enabled
-------
true
```

mcs stop

The **mccli mcs stop** command shuts down the MCS.

Synopsis

mccli mcs stop [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Example

```
mccli mcs stop
0,23000,CLI command completed successfully.
command completed successfully
```

mcs suspend-scheduler

The **mccli mcs suspend-scheduler** command disables the MCS scheduler. When this occurs, no scheduled operations are performed until the scheduler is re-enabled using the **mccli mcs resume-scheduler** command, which is discussed on page 150.

Synopsis

mccli mcs suspend-scheduler [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli mcs resume-scheduler** command.

22309   Scheduler successfully suspended.

22310   Change of the scheduler status to suspended or resume failed.

mcs waitforflushcomplete

The **mccli mcs waitforflushcomplete** command causes the MCS to wait for any in-progress flush to complete.

Synopsis

`mccli mcs waitforflushcomplete` [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

plugin

The **mccli plugin** resource is used to manage client plug-ins. Available commands are:

- ◆ "plugin show" on page 152
- ◆ "plugin update" on page 152

plugin show

The **plugin show** command shows summary properties for all client plug-ins.

Synopsis

`mccli plugin show` [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

plugin update

> **NOTICE**
>
> The **plugin update** command is strictly reserved for EMC internal use only.

Event codes

The following event codes may be returned by the **mccli plugin show** command.

21010   Plugin catalog dynamically updated.

retention

The **mccli retention** resource is used to create and manage retention policies on the Avamar server. Available commands are:

Advanced retention calendar units

All advanced retention options, which begin with --adv-, require a string value in the following format:

Table 85 Advanced retention option string values

Format	Description	Examples
+nnnD	Specifies the number (nnn) of calendar days (D).	• Specify +5D to retain five daily backups. • Specify +13D to retain 13 daily backups.
+nnW	Specifies the number (nn) of calendar weeks (W).	• Specify +1W to retain one week of daily or weekly backups. • Specify +13W to retain 13 weeks of daily or weekly backups.
+nnM	Specifies the number (nn) of calendar months (M).	• Specify +1M to retain one month of daily, weekly, or monthly backups. • Specify +9M to retain nine months of daily, weekly, or monthly backups.
+nnY	Specifies the number (nn) of calendar years (Y).	• Specify +1Y to retain one year of daily, weekly, monthly, or yearly backups. • Specify +5Y to retain five years of daily, weekly, monthly, or yearly backups.

retention add

The **mccli retention add** command creates a retention policy.

If you do not specify options, then the new retention policy has a basic expiration setting of NO_EXPIRATION and the advanced options are set as follows:

◆ Retain one week of daily backups.
◆ Retain one month of weekly backups.
◆ Retain one year of monthly backups.
◆ Retain one year of yearly backups.

However, unless the --override option is set true, then these settings are not enabled. Basic retention settings are used instead of advanced retention settings.

Synopsis

mccli retention add [GLOBAL-OPTIONS] [--adv-daily=STRING(+1D)]
   [--adv-monthly=STRING(+1M)] [--adv-weekly=STRING(+1W)]
   [--adv-yearly=STRING(+1Y)] [--basic=STRING(NO_EXPIRATION)]
   [--domain=STRING(/)] --name=STRING [--override=BOOLEAN(false)]
   [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli retention add** command.

Table 86 Command options for **mccli retention add**

Option	Description
--adv-daily=STRING(+1W)	Specifies the daily advanced retention setting. STRING must be +nnnD, +nnW, +nnM, or +nnY. "Advanced retention calendar units" on page 153 provides additional information.
--adv-monthly=STRING(+1Y)	Specifies the monthly advanced retention setting. STRING must be +nnnD, +nnW, +nnM, or +nnY. "Advanced retention calendar units" on page 153 provides additional information.
--adv-weekly=STRING(+1M)	Specifies the weekly advanced retention setting. STRING must be +nnnD, +nnW, +nnM, or +nnY. "Advanced retention calendar units" on page 153 provides additional information.
--adv-yearly=STRING(+1Y)	Specifies the yearly advanced retention setting. STRING must be +nnnD, +nnW, +nnM, or +nnY. "Advanced retention calendar units" on page 153 provides additional information.
--basic=STRING(NO_EXPIRATION)	Specifies basic retention, which assigns a fixed expiration date. STRING must be one of the following: • YYYY-MM-DD to specify an explicit date • +nn{D|W|M|Y} to specify a duration from today (for example, +4W specifies four weeks from today) • NO_EXPIRATION
--domain=STRING(/)	Specifies the Avamar server domain that contains the retention policy specified by the --name argument.
--name=STRING	Specifies a name for the new retention policy. This argument is required. If you supply a fully qualified retention policy name (for example, /MyDomain/MyRetentionPolicy), then the --domain argument is ignored.
--override=BOOLEAN(false)	If true, then basic retention settings are overridden in favor of advanced retention settings. This option only applies to scheduled backups.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli retention add** command.

22216   Retention Policy created.

23005   Invalid Retention Policy specified on the CLI.

23011   Retention Policy already exists.

retention copy

The **mccli retention copy** command copies an existing retention policy, creating a new retention policy.

Synopsis

```
mccli retention copy [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [--new-domain=STRING] --new-name=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli retention copy** command.

Table 87 Command options for **mccli retention copy**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the retention policy specified by the --name argument.
--name=STRING	Specifies the retention policy to copy. STRING must be a valid retention policy name. This argument is required. Use **mccli retention show**, discussed on page 158, to return a list of valid retention policy names. If you supply a fully qualified retention policy name (for example, /MyDomain/MyRetentionPolicy), then the --domain argument is ignored.
--new-domain=STRING	Specifies an optional new Avamar server destination domain for the copied retention policy.
--new-name=STRING	Specifies a name for the new retention policy. This argument is required.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli retention copy** command.

22216   Retention Policy created.

22289   Retention policy does not exist.

retention delete

The **mccli retention delete** command permanently deletes a retention policy from the Avamar server.

You cannot delete a retention policy if it is currently assigned to a client or group.

Synopsis

```
mccli retention delete [GLOBAL-OPTIONS] [--domain=STRING(/)]
    --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli retention delete** command.

Table 88 Command options for **mccli retention delete**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the retention policy specified by the --name argument.
--name=STRING	Specifies the retention policy to delete. STRING must be a valid retention policy name. This argument is required. Use **mccli retention show,** discussed on page 158, to return a list of valid retention policy names. If you supply a fully qualified retention policy name (for example, /MyDomain/MyRetentionPolicy), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli retention delete** command.

22218   Retention Policy deleted.

22289   Retention policy does not exist.

retention edit

The **mccli retention edit** command edits the properties for a retention policy.

There are no default settings for the **mccli retention edit** command. If you enter the command but do not explicitly supply options and values on the command line, then no changes are applied to the retention policy.

Synopsis

```
mccli retention edit [GLOBAL-OPTIONS] [--adv-daily=STRING]
    [--adv-monthly=STRING] [--adv-weekly=STRING] [--adv-yearly=STRING]
    [--basic=STRING] [--domain=STRING(/)] --name=STRING
    [--override=BOOLEAN] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli retention edit** command.

Table 89 Command options for **mccli retention edit** (page 1 of 2)

Option	Description
`--adv-daily=STRING(+1W)`	Specifies the daily advanced retention setting. STRING must be +nnnD, +nnW, +nnM, or +nnY. "Advanced retention calendar units" on page 153 provides additional information.
`--adv-monthly=STRING(+1Y)`	Specifies the monthly advanced retention setting. STRING must be +nnnD, +nnW, +nnM, or +nnY. "Advanced retention calendar units" on page 153 provides additional information.
`--adv-weekly=STRING(+1M)`	Specifies the weekly advanced retention setting. STRING must be +nnnD, +nnW, +nnM, or +nnY. "Advanced retention calendar units" on page 153 provides additional information.
`--adv-yearly=STRING(+1Y)`	Specifies the yearly advanced retention setting. STRING must be +nnnD, +nnW, +nnM, or +nnY. "Advanced retention calendar units" on page 153 provides additional information.
`--basic=STRING`	Specifies basic retention, which aassigns a fixed expiration date. STRING must be one of the following: • YYYY-MM-DD to specify an explicit date • +nn{D|W|M|Y} to specify a duration from today (for example, +4W specifies four weeks from today) • NO_EXPIRATION

Table 89 Command options for **mccli retention edit** (page 2 of 2)

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the retention policy specified by the --name argument.
--name=STRING	Specifies the retention policy to edit. STRING must be a valid retention policy name. This argument is required. Use **mccli retention show**, discussed on page 158, to return a list of valid retention policy names. If you supply a fully qualified retention policy name (for example, /MyDomain/MyRetentionPolicy), then the --domain argument is ignored.
--override=BOOLEAN	If true, then basic retention settings are overridden in favor of advanced retention settings. This option only applies to scheduled backups.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli retention edit** command.

22217   Retention Policy modified.

22289   Retention policy does not exist.

23005   Invalid Retention Policy specified on the CLI.

retention show

The **mccli retention show** command returns information for all retention policies in an Avamar domain.

Synopsis

```
mccli retention show [GLOBAL-OPTIONS] [--domain=STRING(/)]
   [--name=STRING] [--recursive=BOOLEAN(false)]
   [--verbose=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli retention show** command.

Table 90 Command options for **mccli retention show**

Option	Description
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the retention policy specified by the --name argument.
`--name=STRING`	Specifies the retention policy for which to show information. If not supplied, information for all retention policies is returned. If you supply a fully qualified retention policy name (for example, /MyDomain/MyRetentionPolicy), then the --domain argument is ignored.
`--recursive=BOOLEAN(false)`	If true, then the command shows all retention policies in the domain and any subdomains specified by the --domain=STRING argument. If false or not supplied, then the command only shows retention policies in the domain. Subdomains are not examined.
`--verbose=BOOLEAN(false)`	If true, then detailed information is returned.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli retention show** command.

22289   Retention policy does not exist.

22542   Domain does not exist.

Examples

The following command returns verbose information for all retention policies:

```
mccli retention show --verbose
0,23000,CLI command completed successfully.
Name                          Expiration   Override
----------------------------  ----------   --------
Default Retention             90 Days      Yes
End User On Demand Retention  90 Days      No
Minimal Retention             90 Days      No
Monthly Retention             1 Months     No
Weekly Retention              1 Weeks      No
```

The following command lists information for the Default Retention policy:

```
mccli retention show --name='Default Retention'
0,23000,CLI command completed successfully.
Attribute               Value
----------------------  -----------------
Name                    Default Retention
Basic Expiration Date   90 Days
Override                Yes
Keep days of daily      1
Keep weeks of weekly    1
Keep months of monthly  1
Keep years of yearly    7
```

The following command lists verbose information for the Minimal Retention policy:

```
mccli retention show --name='Minimal Retention'
0,23000,CLI command completed successfully.
Attribute               Value
----------------------  -----------------
Name                    Minimal Retention
Basic Expiration Date   90 Days
Disabled                No
```

NOTICE

It is normal for the Minimal Retention policy to return fewer properties than other retention policies.

The following command lists verbose information for the End User On-Demand Retention policy:

```
mccli retention show --name='End User On Demand Retention'
0,23000,CLI command completed successfully.
Attribute               Value
----------------------  ---------------------------
Name                    End User On Demand Retention
Basic Expiration Date   90 Days
```

NOTICE

It is normal for the End User On Demand Retention policy to return fewer properties than other retention policies.

schedule

The **mccli schedule** resource is used to create and manage schedules on the Avamar server. Available commands are:

schedule add

The **mccli schedule add** command creates a schedule.

Synopsis

```
mccli schedule add {--days=STRING | --nth-day=STRING |
  --on-demand=BOOLEAN(false) | --time=STRING | --week=STRING}
  [--desc=STRING] [--domain=STRING(/)] [--duration=STRING]
  --name=STRING [--start=STRING] [--tz=STRING]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli schedule add** command.

Table 91 Command options for **mccli schedule add** (page 1 of 2)

Option	Description
`--days=STRING`	Creates a schedule that runs weekly on the specified days of the week or monthly on the first such specified day of the month. STRING must be any of the following: SU, M, TU, W, TH, F, or SA. Separate multiple values with commas (for example, SU,TU,F).
`--on-demand=BOOLEAN(false)`	If true, the new schedule is designated as an on-demand schedule. If set true, this option overrides other schedule type designations.
`--nth-day=STRING`	Creates a schedule that runs on the specified calendar day of each month. STRING must be a valid numerical calendar date between 1 and 28, or last.
`--week=STRING`	Creates a schedule that runs on the specified week each month. STRING must be first, second, third, fourth, or last.
`--desc=STRING`	Specifies a short text description of the schedule.
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the schedule specified by the --name argument.

Table 91 Command options for **mccli schedule add** (page 2 of 2)

Option	Description
`--duration=STRING`	Specifies a backup window duration. STRING must be in the format of HH:MM, where HH is the number of hours and MM is the number of minutes. If you do not supply --duration, then the default duration for a daily schedule is one hour and the default duration for a weekly schedule is to eight hours.
`--name=STRING`	Specifies a name for the new schedule. This argument is required. If you supply a fully qualified schedule name (for example, /MyDomain/MySchedule), then the --domain argument is ignored.
`--start=STRING`	Specifies the earliest start time for the schedule. STRING must be in the format of HH:MM, where HH is hours and MM is minutes. The --start option does not apply for daily schedules. If you do not specify --start for a weekly schedule, then the default start time is 10 p.m. in the timezone in which the schedule was created.
`--time=STRING`	Creates a schedule that runs daily at the specified time of the day. STRING must be in the format of HH:MM, where HH is the hour of the day (0-23) and MM is the minute in the hour. Separate multiple values with commas (for example, 4:00,10:07,18:58). Multiple --time options can be specified on the same command line.
`--tz=STRING`	Specifies the timezone for the schedule. If not supplied, then the local timezone is used. Use **schedule show-timezones**, discussed on page 169, to return a list of valid retention timezone names.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli schedule add** command.

22213 Schedule created.

22248 Schedule already exists.

22540 Invalid name character.

22542 Domain does not exist.

Notes

◆ You must specify either --days, --nth-day, --time, or --week to indicate the schedule recurrence. Specify --days to create a weekly schedule, --time to create a daily schedule, and --nth-day or --week to create a monthly schedule.

◆ If you specify --duration, then the default duration for a daily schedule is one hour and the default duration for a weekly schedule is eight hours.

◆ The --start option does not apply for daily schedules.

◆ If you do not specify the --start option for a weekly schedule, then the default start time is 10 p.m. in the timezone in which the schedule was created.

◆ If you do not specify the --tz option, then the default timezone is the local timezone.

Examples

The following example creates a monthly schedule, monthly_2nd_Friday, that runs on the second Friday of each month:

```
mccli schedule add --week=second --days=Friday
--name=monthly_2nd_Friday
```

The following example creates a monthly schedule, my_monthly_sch, that runs on the 28th calendar day of each month:

```
mccli schedule add --name=my_monthly_sch --nth-day=28
```

schedule copy

The **mccli schedule copy** command copies an existing schedule, creating a new schedule.

Synopsis

```
mccli schedule copy [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [--new-domain=STRING] --new-name=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli schedule copy** command.

Table 92 Command options for **mccli schedule copy**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the schedule specified by the --name argument.
--name=STRING	Specifies the schedule to copy. STRING must be a valid schedule name. This argument is required. Use **mccli schedule show**, discussed on page 167, to return a list of valid schedule names. If you supply a fully qualified schedule name (for example, /MyDomain/MySchedule), then the --domain argument is ignored.
--new-domain=STRING	Specifies an optional new Avamar server destination domain for the copied schedule.
--new-name=STRING	Specifies a name for the new schedule. This argument is required.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli schedule copy** command.

22213   Schedule created.

22249   Schedule does not exist.

schedule delete

The **mccli schedule delete** command permanently deletes a schedule from the Avamar server. You cannot delete a schedule if it is assigned to a group or event profile.

Synopsis

```
mccli schedule delete [GLOBAL-OPTIONS] [--domain=STRING(/)]
   --name=STRING [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli schedule delete** command.

Table 93 Command options for **mccli schedule delete**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the schedule specified by the --name argument.
--name=STRING	Specifies the schedule to delete. STRING must be a valid schedule name. This argument is required. Use **mccli schedule show**, discussed on page 167, to return a list of valid schedule names. If you supply a fully qualified schedule name (for example, /MyDomain/MySchedule), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli schedule delete** command.

22215   Schedule deleted.

22249   Schedule does not exist.

22277   Failed to delete a schedule because it is being used by a group or an event profile.

22530   Schedule is read-only.

22531   Unexpected exception occurred.

schedule edit

The **mccli schedule edit** command edits the properties for a schedule.

There are no default settings for the **mccli schedule edit** command. If you enter the command but do not explicitly supply options and values on the command line, then there are no changes to the schedule.

Synopsis

```
mccli schedule edit [{--days=STRING | --nth-day=STRING |
    --on-demand=BOOLEAN(false) | --time=STRING | --week=STRING}]
    [--desc=STRING] [--domain=STRING(/)] [--duration=STRING]
    --name=STRING [--start=STRING] [--tz=STRING]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli schedule edit** command.

Table 94 Command options for **mccli schedule edit** (page 1 of 2)

Option	Description
--days=STRING	Edits the schedule to run weekly on the specified days of the week or monthly on the first such specified day of the month. STRING must be any of the following: SU, M, TU, W, TH, F, or SA. Separate multiple values with commas (for example, SU,TU, F).
--on-demand=BOOLEAN(false)	If true, the new schedule is designated as an on-demand schedule. If set true, this option overrides other schedule type designations.
--nth-day=STRING	Edits the schedule to run on the specified calendar day of each month. STRING must be a valid numerical calendar date between 1 and 28, or last.
--week=STRING	Edits the schedule to run on the specified week each month. STRING must be first, second, third, fourth, or last.
--desc=STRING	Specifies a short text description of the schedule.
--domain=STRING(/)	Specifies the Avamar server domain that contains the schedule specified by the --name argument.
--duration=STRING	Specifies a backup window duration for the schedule. STRING must be in the format of HH:MM, where HH is the number of hours and MM is the number of minutes.
--name=STRING	Specifies the schedule to edit. STRING must be a valid schedule name. This argument is required. Use **mccli schedule show**, discussed on page 167, to return a list of valid schedule names. If you supply a fully qualified schedule name (for example, /MyDomain/MySchedule), then the --domain argument is ignored.

Table 94 Command options for **mccli schedule edit** (page 2 of 2)

Option	Description
`--start=STRING`	Specifies the earliest start time for the schedule. STRING must be in the format of HH:MM, where HH is hours and MM is minutes.
`--time=STRING`	Edits the schedule to run daily at the specified time of the day. STRING must be in the format of HH:MM, where HH is the hour of the day (0-23) and MM is the minute in the hour. Separate multiple values with commas (for example, 4:00,10:07,18:58). Multiple --time options can be specified on the same command line.
`--tz=STRING`	Specifies the timezone for the schedule. If not supplied, then the local timezone is used. Use **schedule show-timezones**, discussed on page 169, to return a list of valid timezone names.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli schedule edit** command.

22214   Schedule modified.

22249   Schedule does not exist.

Notes

◆ You must specify either --days, --nth-day, --time, or --week to indicate the schedule recurrence. Specify --days to create a weekly schedule, --time to create a daily schedule, and --nth-day or --week to create a monthly schedule.

◆ If you specify --duration, then the default duration for a daily schedule is one hour and the default duration for a weekly schedule is eight hours.

◆ The --start option does not apply for daily schedules.

◆ If you do not specify the --start option for a weekly schedule, then the default start time is 10 p.m. in the timezone in which the schedule was created.

◆ If you do not specify the --tz option, then the default timezone is the local timezone.

Example

The following example edits a monthly schedule called my_monthly_sch to run on the last calendar day of each month.

```
mccli schedule edit --name=my_monthly_sch --nth-day=last
```

schedule show

The **mccli schedule show** command lists all schedules and detailed schedule information.

Synopsis

```
mccli schedule show [GLOBAL-OPTIONS] [--domain=STRING(/)]
   [--name=STRING] [--recursive=BOOLEAN(false)]
   [--verbose=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli schedule show** command.

Table 95 Command options for **mccli schedule show**

Option	Description
--domain=STRING(/)	Specifies the Avamar server domain that contains the schedule specified by the --name argument.
--name=STRING	Specifies the name of the schedule for which to show information. If not supplied, all schedules are listed with summary information. If you supply a fully qualified schedule name (for example, /MyDomain/MySchedule), then the --domain argument is ignored.
--recursive=BOOLEAN(false)	If true, then the command shows all schedules in the domain and any subdomains specified by the --domain=STRING argument. If false or not supplied, then the command only shows schedules in the domain. Subdomains are not examined.
--verbose=BOOLEAN(false)	If true, then detailed schedule information is returned. If false or not supplied, then only schedule names are returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli schedule show** command.

22249   Schedule does not exist.

22542   Domain does not exist.

Examples

The following command returns a list of properties for the "Default Schedule:"

```
mccli schedule show --name=Default Schedule
0,23000,CLI command completed successfully.
Attribute                          Value
-----------------------------      -------------------------------
Name                               Default Schedule
Domain                             /
ReadOnly                           true
Native Timezone                    America/Los_Angeles
Daylight Savings Adjustment (m)    60
Next Run Time                      2011-09-20 10:00 PM
Start Time                         10:00 PM Pacific Daylight Time
Backup Window Duration (hours)     8.0
Repeat                             Weekly
Days of Week                       Sun, Mon, Tue, Wed, Thu, Fri, Sat
Delay Start Until                  N/A
End Policy                         No End Date
Description                        N/A
```

The following table describes **mccli schedule show** output attributes and possible values.

Table 96 Output attributes and possible values for **mccli schedule show**

Attribute	Values	Normalized values	Description of normalized values
Next Run Time	2011-09-13 09:00 AM	1189699200000	UNIX precision time format, represented as UTC milliseconds from the epoch.
Start Time	9:00 AM America/Los_Angeles	61200000	Number of milliseconds into a day starting at midnight in local timezone.
Delay Start Until	2011-09-12 12:19 PM PDT	1189624773776	UNIX precision time format, represented as UTC milliseconds from the epoch.
End Policy	2011-12-10 11:59 PM PST	1228982399999	UNIX precision time format, represented as UTC milliseconds from the epoch.

schedule show-timezones

The **mccli schedule show-timezones** command lists all valid timezones, which you can supply with other **mccli schedule** commands using the **--tz** option.

Synopsis

```
mccli schedule show-timezones [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

server

The **mccli server** resource is used to monitor various Avamar server functions. Available commands are:

server show-prop

The **mccli server show-prop** command returns detailed properties for the Avamar server, similar to what is shown on the Server Management tab in the Server window in Avamar Administrator. The *EMC Avamar Administration Guide* provides a thorough discussion of server properties.

If you run **mccli server show-prop** without command options, then a summary of server properties is shown.

Synopsis

```
mccli server show-prop [GLOBAL-OPTIONS] {--module=STRING |
   --node=STRING | --node=STRING --partition=STRING} --verbose
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli server show-prop** command.

Table 97 Command options for **mccli server show-prop**

Option	Description
`--module=STRING`	Specifies the Avamar server module from which to show properties. STRING must be a valid Avamar server module designator. Module designators are typically single-digit integers beginning with zero. For example, 0, 1, 2, and so forth. A value of ALL returns status for all modules.
`--node=STRING`	Specifies the Avamar server node from which to show properties. STRING must be a valid Avamar server numeric node designator in the format of MODULE.NODE. For example, 0.0 is typically the first storage node in a multi-node server. A value of ALL returns status for all nodes.
`--partition=STRING`	Specifies the Avamar server data partition from which to show properties. STRING must be a valid Avamar server partition designator. Partition designators are typically single-digit integers beginning with zero. For example, 0, 1, 2, and so forth. You must also supply the --node=STRING option. A value of ALL returns status for all partitions.
`--verbose`	If supplied, maximum information is returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Notes

Consider the following notes for the **mccli server show-prop** command:

◆ The --module=STRING and --node=STRING options are mutually exclusive.

◆ The --partition=STRING option requires --node=STRING.

Examples

The following command returns a summary of server properties:

```
mccli server show-prop
0,23000,CLI command completed successfully.
Attribute                            Value
-------------------------------      -------------------------
State                                Node Offline
Active sessions                      0
Total capacity                       1.3 TB
Capacity used                        91 GB
Server utilization                   0.7%
Bytes protected                      0.0 bytes
Bytes protected quota                Not configured
License expiration                   Never
Time since Server initialization     17 days 21h:13m
Last checkpoint                      2011-10-01 04:00:28 PDT
Last validated checkpoint            2011-09-30 22:00:28 PDT
System Name                          avamar-1
System ID                            1216944475@00:19:B9:BA:14:EA
HFSAddr                              avamar-1.example.com
HFSPort                              27000
IP address                           10.6.248.196
Number of nodes                      6
Nodes Online                         5
Nodes Offline                        1
Nodes Read-only                      0
Nodes Timed-out                      0
```

The following command returns information for server node 0.0 partition zero (0):

```
mccli server show-prop --node=0.0 --partition=0
0,23000,CLI command completed successfully.
Attribute                            Value
-------------------------------      -------------------------
Node ID                              0.0
State                                ONLINE
Runlevel                             fullaccess
Accessmode                           mhpu+0hpu+0hpu
Port                                 26000
Dispatcher                           27000
Server uptime                        0 days 18h:08m
Server utilization                   0.1%
Number of stripes                    22
Server version                       4.1.0-113
Version                              v2.6.9-67.ELsmp
Node uptime                          6 days 00h:07m
Total capacity                       86.0 TB
Capacity used                        6.4 TB
Load average                         0.22
CPU %                                1.9%
Disk reads                           0.11/sec
Disk writes                          2.21/sec
Network reads                        2K/sec
Network writes                       2K/sec
IP address                           10.6.249.38
Mac address                          00:13:72:59:8F:AE
Number of partitions 4
```

The following table describes **mccli server show-prop** output attributes and possible values.

Table 98 Output attributes and possible values for **mccli server show-prop**

Attribute	Values	Normalized values	Description of normalized values
Bytes protected quota	None	0	
	Not Configured	-1	
License expiration	Never	0	
	Not Set	-1	
Time since Server initialization	0 days 20h:06m	72389000	Number of milliseconds ago.
Last checkpoint	2011-09-12 11:00:17 PDT	1189620017000	Unix precision time format, represented as UTC milliseconds from the epoch.
	No checkpoints performed	-1	
Last validated checkpoint	2011-09-11 23:00:16 PDT	1189576816000	Unix precision time format, represented as UTC milliseconds from the epoch.
	No validated checkpoints.	-1	
Server uptime	0 days 20h:05m	72331000	Number of milliseconds ago.
Node uptime	12 days 17h:30m	1099814000	Number of milliseconds ago.

server show-services

The **mccli server show-services** command returns detailed properties for Avamar server services, similar to what is shown in the Services Administration feature in Avamar Administrator. The *EMC Avamar Administration Guide* provides a thorough discussion of server properties.

Synopsis

```
mccli server show-services [GLOBAL-OPTIONS] [--service=STRING]
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command option is available for the **mccli server show-services** command.

Table 99 Command options for **mccli server show-services**

Option	Description
--service=STRING	Specifies the service for which to show information. STRING must be a valid Avamar service name (for example, replicate). If not supplied, then all services are shown.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

server show-util

The **mccli server show-util** command returns detailed utilization information from the Avamar server, similar to what is shown in the Server Monitor in Avamar Administrator. The *EMC Avamar Administration Guide* provides a thorough discussion of server properties.

If you run **mccli server show-util** without command options, then a summary of server utilization is shown.

Synopsis

```
mccli server show-util [GLOBAL-OPTIONS] [--module=STRING |
    --node=STRING] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli server show-util** command.

Table 100 Command options for **mccli server show-util**

Option	Description
--module=STRING	Specifies the Avamar server module from which to show utilization information. STRING must be a valid Avamar server module designator. Module designators are typically single-digit integers beginning with zero. For example, 0, 1, 2, and so forth.
--node=STRING	Specifies the Avamar server node from which to show utilization information. STRING must be a valid Avamar server numeric node designator in the format of MODULE.NODE. For example, 0.1 is typically the first storage node in a multi-node server. A value of ALL returns status for all nodes.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Notes

The --module=STRING and --node=STRING options are mutually exclusive.

Examples

The **mccli server show-util** command returns a summary of server utilization, as shown in the following example output. The sample output appears in a single table on-screen but is split into two tables here to fit on the page.

Node	State	CPU-Load	CPU-User	CPU-Sys	Network-Ping
0.0	ONLINE	0.00	0.0%	0.1%	0.00
0.1	ONLINE	0.00	0.1%	0.1%	0.00
0.2	ONLINE	0.00	0.1%	0.0%	0.00
0.3	OFFLINE	Unknown	Unknown	Unknown	
0.4	ONLINE	0.03	0.0%	0.0%	0.00
0.5	ONLINE	0.00	0.1%	0.0%	0.00
0.6	ONLINE	0.00	0.1%	0.1%	0.00
0.7	ONLINE	0.00	0.0%	0.1%	0.04

Network-In	Network-Out	Disk-Reads	Disk-Writes	Disk-Utilization
1K/sec	1K/sec	0/sec	0.61/sec	0.9%
1K/sec	2K/sec	0/sec	0.49/sec	0.8%
2K/sec	3K/sec	0/sec	0.77/sec	0.7%
NaN	NaN	0/sec	0/sec	0.0%
2K/sec	3K/sec	0/sec	0.59/sec	0.9%
1K/sec	1K/sec	0/sec	0.56/sec	0.6%
1K/sec	1K/sec	0/sec	0.62/sec	0.4%
1K/sec	1K/sec	0/sec	0.64/sec	0.7%

snapup

The **mccli snapup** resource is used to restore folders and files to clients, as well as to manage backups on the Avamar server. Available commands are:

- "snapup delete" on page 175
- "snapup edit" on page 176
- "snapup restore" on page 177
- "snapup show" on page 179
- "snapup validate" on page 181

> *NOTICE*

The snapup management resource has been deprecated in favor of the backup management resource, which is discussed starting on page 39.

snapup delete

The **mccli snapup delete** command permanently deletes a backup from the server.

Synopsis

```
mccli snapup delete [--created=STRING] [--domain=STRING(/)]
   [--labelNum=INTEGER] [--name=STRING]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli snapup-delete** command.

Table 101 Command options for **mccli snapup-delete**

Option	Description
--created=STRING	Specifies the date that the backup was created. STRING must be in the format of YYYY-MM-DD, where YYYY, MM, and DD are a calendar year, month, and date, respectively.
[--domain=STRING(/)]	Specifies the Avamar server domain that contains the client specified by the --name argument.
--labelNum=INTEGER	Specifies the label number of the backup to delete.
--name=STRING	Specifies the client to which the backup belongs. This argument is required. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli snapup-delete** command.

22236   Client does not exist.

22552   Backup does not exist.

snapup edit

The **mccli snapup edit** command enables you to edit the backup expiration date by one of the following methods:

◆ Directly specifying a new expiration date.

◆ Adding days (d), weeks (w), months (m), or years to the existing backup expiration date.

◆ Specifying that the backup should never expire.

Synopsis

```
mccli snapup edit [--created=STRING] [--domain=STRING(/)]
   [--expiration=STRING] [--labelNum=INTEGER] [--name=STRING]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli snapup edit** command.

Table 102 Command options for **mccli snapup edit** (page 1 of 2)

Option	Description
--created=STRING	Specifies the date that the backup was created. STRING must be in the format of YYYY-MM-DD, where YYYY, MM, and DD are a calendar year, month, and date, respectively.
[--domain=STRING(/)]	Specifies the Avamar server domain that contains the client specified by the --name argument.

Table 102 Command options for **mccli snapup edit** (page 2 of 2)

Option	Description
--expiration=STRING	Specifies a new expiration date. STRING must be in one of the following formats: YYYY-MM-DD | +nn{D|W|M|Y} | NO_EXPIRATION where: • YYYY-MM-DD is a specific year (YYYY), month (MM), and day (DD) • +nn{D|W|M|Y is the number (nn) of additional days (d), weeks (w), months (m), or years to add to the existing backup expiration date. • NO_EXPIRATION specifies that the backup should never expire.
--labelNum=INTEGER	Specifies the label number of the backup to edit.
--name=STRING	Specifies the client to which the backup belongs. This argument is required. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli snapup edit** command.

22236   Client does not exist.

22552   Backup does not exist.

snapup restore

The **mccli snapup restore** command initiates a restore of folders and files to a client.

This command returns an activity ID, which can be passed to **mccli activity show**, discussed on page 37, to get status for the restore activity.

Synopsis

```
mccli snapup restore [GLOBAL-OPTIONS] [--cmd=STRING [--cmd=STRING]
   ...] [--data=STRING] [--dest-dir=STRING] [--domain=STRING(/)]
   --labelNum=INTEGER --name=STRING --plugin=INTEGER [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli snapup restore** command.

Table 103 Command options for **mccli snapup restore**

Option	Description
`--cmd=STRING`	Specifies optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command. For example, this is valid --cmd syntax: `--cmd="verbose=5" --cmd="throttle=5"` However, this is not valid --cmd syntax: `--cmd="verbose=5 throttle=5"`
`--data=STRING`	Specifies a single optional target file or folder to restore. Each --data option can only specify one target file or folder to restore. Use multiple --data options to restore multiple targets.
`--dest-dir=STRING`	Specifies an optional destination folder for the restored folders or files.
`--domain=STRING(/)`	Specifies the Avamar server domain that contains the client specified by the --name argument.
`--labelNum=INTEGER`	Specifies the backup to use for the restore. This argument is required. Use **mccli snapup show,** discussed on page 179, to return a list of valid backup label numbers.
`--name=STRING`	Specifies the target client for the restore. STRING must be a valid client name. This argument is required. Use **mccli client show,** discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain option is ignored.
`--plugin=INTEGER`	Specifies the plug-in ID for the plug-in to use for the restore. This argument is required. Use **mccli plugin show,** discussed on page 152, to return a list of valid plug-in IDs.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli snapup restore** command.

22236	Client does not exist.
22312	Client restore scheduled.
23009	Invalid plugin specified on the CLI.

snapup show

The **mccli snapup show** command returns all backups currently stored on the Avamar server for a client.

Synopsis

```
mccli snapup show [GLOBAL-OPTIONS] [--after=STRING --before=STRING]
   [--dir=STRING(/)] [--domain=STRING(/)] [--labelNum=INTEGER]
   --name=STRING [--recursive=BOOLEAN(false)]
   [--verbose=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli snapup show** command.

Table 104 Command options for **mccli snapup show**

Option	Description
--after=STRING	Used with --before to specify a range of backup dates to return. STRING must be in the format of YYYY-MM-DD, where YYYY is the four-digit year, MM is the two-digit calendar month, and DD is the two-digit day of the month.
--before=STRING	Used with --after to specify a range of backups dates to return. STRING must be in the format of YYYY-MM-DD, where YYYY is the four-digit year, MM is the two-digit calendar month, and DD is the two-digit day of the month.
--dir=STRING(/)	Specifies the initial folder of the backup from which to begin listing folders and files.
--domain=STRING(/)	Specifies the Avamar server domain that contains the client specified by the --name argument.
--labelNum=INTEGER	Specifies a backup for which to show information.
--name=STRING	Specifies the client from which to show backups. This argument is required. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
--recursive=BOOLEAN(false)	If true, then backup folders and files are recursively shown.
--verbose=BOOLEAN(false)	If true, then detailed information is returned.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli snapup show** command.

22236   Client does not exist.

22504   Failed to retrieve the backups for a client.

Examples

The following command returns a list of backups stored in the system for MyClient.example.com:

```
mccli snapup show --name=clients/MyClient.example.com
0,23000,CLI command completed successfully.
 Created                 LabelNum   Size
 ----------------------  --------   ----------
 2010-01-31 15:51:30 PST  2         4767841280
 2010-01-31 15:00:23 PST  1         4750878720
```

The following command returns verbose file and folder information for a parent folder (C:/) in a client backup (the second backup taken from MyClient.example.com):

```
mccli snapup show --name=clients/MyClient.example.com --verbose --labelNum=2 --dir=C:/
0,23000,CLI command completed successfully.
Type Date               Size          Owner          Group    Permissions Name
---- ------------------ ------------- -------------- -------- ----------- ------------------------
Dir  2010-12-12 08:59:27 205,526,604  Administrators unknown  drwxrwxr-x  C:/Documents and Settings/
Dir  2010-01-17 11:33:54 1,601,595,257 Administrators unknown dr-xr-x---  C:/Program Files/
Dir  2010-12-12 08:59:55 1,841        Administrator  None     drwx------  C:/RECYCLER/
Dir  2010-01-31 15:45:30 20,480       Administrators unknown  d--x------  C:/System Volume Info/
Dir  2010-01-17 15:18:22 211          Administrators unknown  drwxrwx---  C:/tmp/
Dir  2010-01-18 09:14:25 1,600,258,500 Administrators unknown drwxrwx---  C:/WINDOWS/
File 2010-12-09 15:21:15 211          Administrators unknown  -r-xr-x---  C:/boot.ini
File 2010-12-12 08:56:49 4,128        Administrators unknown  -rwxrwx---  C:/INFCACHE.1
File 2010-08-11 15:15:00 0            Administrators unknown  -rwxrwx---  C:/IO.SYS
File 2010-08-11 15:15:00 0            Administrators unknown  -rwxrwx---  C:/MSDOS.SYS
File 2010-08-04 03:00:00 47,564       Administrators unknown  -r-xr-x---  C:/NTDETECT.COM
File 2010-08-04 03:00:00 250,032      Administrators unknown  -r-xr-x---  C:/ntldr
```

snapup validate

The **mccli snapup validate** command initiates the validation of a backup.

This command returns an activity ID, which can be passed to **mccli activity show**, discussed on page 37, to get status for this validate activity.

Synopsis

```
mccli snapup validate [GLOBAL-OPTIONS] [--cmd=STRING [--cmd=STRING]
   ...] [--domain=STRING(/)] --labelNum=INTEGER --name=STRING
   --plugin=INTEGER [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli snapup validate** command.

Table 105 Command options for **mccli snapup validate**

Option	Description

--cmd=STRING	Specifies optional plug-in commands. You can supply multiple --cmd arguments, but each argument can only specify one plug-in command. For example, this is valid --cmd syntax:
	`--cmd="verbose=5" --cmd="throttle=5"`
	
	However, this is not valid --cmd syntax:
	`--cmd="verbose=5 throttle=5"`

--domain=STRING(/)	Specifies the Avamar server domain that contains the client specified by the --name argument.
--labelNum=INTEGER	Specifies the backup to validate. This argument is required.
--name=STRING	Specifies the client associated with the backup to validate. STRING must be a valid client name. This argument is required. Use **mccli client show**, discussed on page 80, to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the --domain argument is ignored.
--plugin=INTEGER	Specifies the plug-in ID for the plug-in used for the backup to validate. This argument is required. Use **mccli plugin show**, discussed on page 152, to return a list of valid plug-in IDs.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli snapup validate** command.

22236	Client does not exist.
22312	Client restore scheduled.
23009	Invalid plugin specified on the CLI.

user

The **mccli user** resource is used to create and manage backup user accounts on the Avamar server. Available commands are:

- "user add" on page 183
- "user authenticate" on page 185
- "user delete" on page 187
- "user edit" on page 188
- "user show" on page 190
- "user show-auth" on page 191

Enumerating user resources

Several **mccli user** commands require precise case-sensitive names (STRING values) for the following client resources:

- Authentication system
- Client name
- Domain name

You can view a list of valid authentication systems, client names and domain names using the following commands, respectively:

- **mccli user show-auth,** discussed on page 191
- **client show,** discussed on page 80
- **domain show,** discussed on page 118

user add

The **mccli user add** command creates a backup user account for a client or domain.

Synopsis

```
mccli user add [GLOBAL-OPTIONS] [--authenticator=STRING(Axion)]
   --client-domain=STRING [--client-name=STRING] --name=STRING
   [--password=STRING] [--password-confirm=STRING] --role=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli user add** command.

Table 106 Command options for **mccli user add** (page 1 of 2)

Option	Description
`--authenticator=String(Axion)`	Specifies the authentication system to use to grant the new user access to the Avamar server. STRING must be either Axion or the name of another valid authentication system that has been configured for use with the Avamar server; STRING is case-insensitive. Use **mccli user show-auth**, discussed on page 191, to return a list of valid authentication systems.
`--client-domain=STRING`	Specifies the domain for the new user. STRING must be a valid domain name. This argument is required. If you supply both --client-domain and --client-name, then the user is added to the client. If you supply only --client-domain, then the user is added to the domain. Use **mccli domain show**, discussed on page 118, to return a list of valid domain names.
`--client-name=STRING`	Specifies the client to which to addthe new user account. STRING must be a valid client name. Use **mccli client show**, discussed on page 80, to return a list of valid client names.
`--name=STRING`	Specifies the user ID or name for the new user account. This argument is required.
`--password=STRING`	Specifies the password for the new user. If you specify an external authentication system using the --authenticator argument, then do not supply the --password and --password-confirm arguments. This is because the external authentication system manages the password for the user.

Table 106 Command options for **mccli user add** (page 2 of 2)

Option	Description
`--password-confirm=STRING`	Specifies the password confirmation string. If you supply --password=STRING, then STRING must be the same value as --password=STRING. This is intended to prevent a single incorrect entry of the password. If you specify an external authentication system using the --authenticator argument, then do not supply the --password and --password-confirm arguments. This is because the external authentication system manages the password for the user.
`--privilege=STRING`	Specifies the privilege level for the user. Beginning with version 4.0, use of this option is deprecated in favor of --role.
`--role=STRING`	Specifies the role for the user. STRING must be one of the following: • ActivityOperator • Administrator • Auditor • Backup • BackupOperator • BackupRestore • BackupRestoreOperator • Restore • RestoreIgnoreFilePermissions • RestoreOperator You can only use the RestoreIgnoreFilePermissions role if you use an external authentication system. The *EMC Avamar Administration Guide* provides additional information about roles.

Display options

provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli user add** command.

22528   Failed to add user.

22529   User added.

22540   Invalid name character.

22542   Domain does not exist.

22546   Password mismatch.

22548   User already exists.

22549   Invalid privilege.

22550   Invalid authenticator.

Notes

Consider the following notes for the **mccli user add** command:

◆ Beginning with version 4.0, use of the --privilege option is deprecated in favor of --role.

◆ Beginning with version 4.0, use of the Snapup and SnapupRestore roles are deprecated in favor of Backup and BackupRestore, respectively.

◆ You cannot add new user accounts to the MC_RETIRED domain or to any clients in that domain.

Examples

The following command adds a new user account, jsmith, with Administrator privileges to /clients/MyDomain:

```
mccli user add --name=jsmith --client-domain=/clients/MyDomain
--role=Administrator --password=XXXXXX --password-confirm=XXXXXX
0,22529,User added.
 Attribute Value
 --------- -----------------------------------------------------------------
 user      <appuser authentication="Avamar Authentication System"
           folder="/clients/MyDomain" name="jsmith" privilege="Administrator"/>
```

The following command adds a new user account, jsmith, with backup privileges to /clients/MyClient:

```
mccli user add --name=jsmith --client-domain=/clients
--client-name=MyClient --role=Administrator --password=XXXXXX
--password-confirm=XXXXXX
0,22529,User added.
 Attribute Value
 --------- -----------------------------------------------------------------
 user      <appuser authentication="Avamar Authentication System"
           folder="/clients/MyClient" name="jsmith" privilege="Back up Only"/>
```

user authenticate

The **mccli user authenticate** command verifies user authentication settings. This is useful for testing usernames, passwords, and authentication system settings before creating a new user account.

Synopsis

```
mccli user authenticate [GLOBAL-OPTIONS]
   [--authenticator=STRING(Axion)] --client-domain=STRING
   [--client-name=STRING] --name=STRING --password=STRING
   [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli user authenticate** command.

Table 107 Command options for **mccli user authenticate**

Option	Description
`--authenticator=String(Axion)`	Specifies the authentication system used to grant user access to the Avamar server. STRING must be either Axion or the name of another valid authentication system that has been configured for use with the Avamar server. STRING is case-insensitive. Use **mccli user show-auth,** discussed on page 191, to return a list of valid authentication systems.
`--client-domain=STRING`	Specifies the domain associated with the client or user account. STRING must be a valid domain name. This argument is required. Use **mccli domain show,** discussed on page 118, to return a list of valid domain names.
`--client-name=STRING`	Specifies the name of the client associated with the user account. STRING must be a valid client name. Use **mccli client show,** discussed on page 80, to return a list of valid client names.
`--name=STRING`	Specifies the user ID or name to authenticate.
`--password=STRING`	Specifies the password to authenticate the user. This argument is required.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli user authenticate** command.

22348   Successful authenticate.

22349   Failed authenticate.

22542   Domain does not exist.

22550   Invalid authenticator

22801   Login failure.

user delete

The **mccli user delete** command permanently deletes a user from the Avamar server.

Synopsis

```
mccli user delete [GLOBAL-OPTIONS] --authenticator=STRING
    --client-domain=STRING [--client-name=STRING] --name=STRING
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli user delete** command.

Table 108 Command options for **mccli user delete**

Option	Description
--authenticator=String	Specifies the authentication system associated with the user account. STRING must be either Axion or the name of another valid authentication system that has been configured for use with the Avamar server. STRING is case-insensitive. Use **mccli user show-auth,** discussed on page 191, to return a list of valid authentication systems.
--client-domain=STRING	Specifies the domain associated with the client or user account. STRING must be a valid domain name. This argument is required. If you supply both --client-domain and --client-name, then the user is deleted from the client. If you supply only --client-domain, then the user is deleted from the domain. Use **mccli domain show,** discussed on page 118, to return a list of valid domain names.
--client-name=STRING	Specifies the client name associated with the user account. STRING must be a valid client name. Use **mccli client show,** discussed on page 80, to return a list of valid client names.
--name=STRING	Specifies the user ID or name of the user to delete.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli user delete** command.

22522   Failed user delete.

22523   User deleted.

22540   Invalid name character.

22542   Domain does not exist.

22547   User does not exist.

22550   Invalid authenticator.

22600   Server inactive.

user edit

The **mccli user edit** command edits the properties for a user.

Synopsis

```
mccli user edit [GLOBAL-OPTIONS] --authenticator=STRING
    --client-domain=STRING [--client-name=STRING] --name=STRING
    [--password=STRING] [--password-confirm=STRING] [--role=STRING]
    [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli user edit** command.

Table 109 Command options for **mccli user edit** (page 1 of 2)

Option	Description
`--authenticator=String`	Specifies the authentication system associated with the user account. STRING must be either Axion or the name of another valid authentication system that has been configured for use with the Avamar server. STRING is case-insensitive. This argument is required. Use **mccli user show-auth,** discussed on page 191, to return a list of valid authentication systems.
`--client-domain=STRING(/)`	Specifies the domain associated with the client or user account. STRING must be a valid domain name. This argument is required. If you supply both --client-domain and --client-name, then the user resides on the client. If you supply only --client-domain, then the user resides in the domain. Use **mccli domain show,** discussed on page 118, to return a list of valid domain names.
`--client-name=STRING`	Specifies the name of the client associated with the user account. STRING must be a valid client name. Use **mccli client show,** discussed on page 80, to return a list of valid client names.

Table 109 Command options for **mccli user edit** (page 2 of 2)

Option	Description
`--name=STRING`	Specifies the user ID or name for the user account. STRING must be a valid user name. This argument is required. Use **mccli user show,** discussed on page 190, to return a list of valid domain names.
`--password=STRING`	Specifies the password for the user.
`--password-confirm=STRING`	Specifies the password confirmation string. If you supply --password=STRING, then STRING must be the same value as --password=STRING. This is intended to prevent a single incorrect entry of the password.
`--privilege=STRING`	Specifies the privilege level for the user. Beginning with version 4.0, use of this option is deprecated in favor of --role.
`--role=STRING`	Specifies the role for the user. STRING must be one of the following: • ActivityOperator • Administrator • Auditor • Backup • BackupOperator • BackupRestore • BackupRestoreOperator • Restore • RestoreIgnoreFilePermissions • RestoreOperator You can use the **RestoreIgnoreFilePermissions** role only if you use an external authentication system. The *EMC Avamar Administration Guide* provides additional information about roles.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli user edit** command.

22520	User updated.
22521	Failed user update.
22542	Domain does not exist.
22546	Password mismatch.
22547	User does not exist.
22549	Invalid privilege.
22550	Invalid authenticator.
22551	Can not edit reserved user.
23001	Missing required arguments.

Notes

Consider the following notes for the **mccli user edit** command:

◆ Beginning with version 4.0, use of the --privilege option is deprecated in favor of --role.

◆ Beginning with version 4.0, use of the Snapup and SnapupRestore roles are deprecated in favor of Backup and BackupRestore, respectively.

◆ You cannot edit user accounts in the MC_RETIRED domain.

user show

The **mccli user show** command displays properties for users on an access list for a client or domain.

Synopsis

```
mccli user show [GLOBAL-OPTIONS] [--authenticator=String(Axion)]
   [--client-domain=STRING(/)] [--client-name=STRING] [--name=STRING]
   [--recursive=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli user show** command.

Table 110 Command options for **mccli user show**

Option	Description
--authenticator=String(Axion)	Specifies the authentication system used to grant user access to the Avamar server. STRING must be either Axion or the name of another valid authentication system that has been configured for use with the Avamar server. STRING is case-insensitive. Use **mccli user show-auth**, discussed on page 191, to return a list of valid authentication systems.
--client-domain=STRING(/)	Specifies the domain associated with the client.
--client-name=STRING	Specifies the name of the client to examine for user access. STRING must be a valid client name. Use **mccli client show**, discussed on page 80, to return a list of valid client names.
--name=STRING	Specifies the user ID or name.
--recursive=BOOLEAN(false)	If true, then the command examines all domains in the domain and any subdomains. If false or not supplied, then the command only examines the domain. Subdomains are not examined.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Event codes

The following event codes may be returned by the **mccli user show** command.

22542   Domain does not exist.

22547   User does not exist.

user show-auth

The **mccli user show-auth** command displays all authentication systems configured for use with the Avamar server.

Synopsis

mccli user show-auth [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

vcenter

The **mccli vcenter** resource comprises a single command, **vcenter browse**.

vcenter browse

The **mccli vcenter browse** command is used to browse a vCenter for virtual machines, either by specifying a folder path, or ESX server or datacenter names. Both Host and Clusters, and VMs and Templates views are supported.

Synopsis

```
mccli vcenter browse [GLOBAL-OPTIONS] [--datacenter=STRING]
    [--esx-host=STRING] [--folder=STRING] --name=STRING
    [--recursive=BOOLEAN(false)] [--type=STRING(VM)]
    [--vsphere-hosts-clusters-view=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

Global options

"Global options" on page 34 provides a list of GLOBAL-OPTIONS common to all **mccli** commands.

Command options

The following command options are available for the **mccli vcenter browse** command.

Table 111 Command options for **mccli vcenter browse** (page 1 of 2)

Option	Description
--datacenter=STRING	Specifes a fully qualified datacenter name in vCenter in the format of PATH/NAME, where PATH is the full folder location and NAME is the datacenter name. This option is only valid when browsing for virtual machines.
--esx-host=STRING	Specifies a fully qualified ESX server hostname in a datacenter, in the format of PATH/NAME, where PATH is the full folder location and NAME is the hostname. This option is only valid when browsing for virtual machines in the vSphere Hosts and Clusters view (that is, --vsphere-hosts-clusters-view=true).
--folder=STRING	Specifies a folder path in a datacenter that will be browsed. This option is only valid when browsing for virtual machines in the vSphere VMs and Templates view (that is, --vsphere-hosts-clusters-view=false).
--name=STRING	Specifies the vCenter name. This argument is required.
--recursive=BOOLEAN(false)	Recursively browses all levels of the specified folder path in a datacenter.

Table 111 Command options for **mccli vcenter browse** (page 2 of 2)

Option	Description
`--type=STRING(VM)`	Specifies the type of entity that should be searched for by this command. String must be one of the following: • VM — Browse virtual machines. Returns detailed virtual machine information such as Name, Guest OS, Server, Location, Template, Powered On/Off, Changed Block Tracking, and Is protected. This is the default entity type. • DATASTORE — Browse datastores. Returns detailed datastore information such as Name, Type, Accessible/Not, Hosts.
`--vsphere-hosts-clusters-view=BOOLEAN(false)`	If true, the vSphere Hosts and Clusters view is processed. If false or not supplied, the vSphere VMs and Templates view is processed. This option is only valid when browsing for virtual machines.

Display options

"Display options" on page 34 provides a list of DISPLAY-OPTIONS common to all **mccli** commands.

Examples

The following example recursively browses vCenter 10.31.183.228 for virtual machines, and formats the output as XML:

```
mccli vcenter browse --name=10.31.183.228 --recursive=true --xml
<CLIOutput>
   <Results>
      <ReturnCode>1</ReturnCode>
      <EventCode>23999</EventCode>
      <EventSummary />
   </Results>
   <Data>
      <Row>
         <Name>ACMCommunity</Name>
         <GuestOS>debian5Guest</GuestOS>
         <Server>10.31.183.7</Server>

   <Location>/VAAYU-DEV-WIN/vm/Test1/Test1/ACMCommunity</Location>
         <Template>No</Template>
         <PoweredOn>Yes</PoweredOn>
         <ChangedBlock>No</ChangedBlock>
         <Protected>Yes</Protected>
      </Row>
      <Row>
         <Name>RHEL564BUILDDOWNLOAD224</Name>
         <GuestOS>rhel5_64Guest</GuestOS>
         <Server>10.31.183.7</Server>
         <Location>/VAAYU-DEV-WIN/vm/Discovered virtual
   machine/RHEL564BUILDDOWNLOAD224</Location>
         <Template>No</Template>
         <PoweredOn>Yes</PoweredOn>
         <ChangedBlock>No</ChangedBlock>
         <Protected>No</Protected>
      </Row>
   </Data>
</CLIOutput>
```

The following example browses vCenter 10.31.183.228 for datastores:

```
mccli vcenter browse --name=10.31.183.228 --type=DATASTORE --xml
Name      Type  Accessible  Hosts
--------  ----  ----------  -----------
Storage1  VMFS  Yes         10.31.183.7
Storage2  VMFS  Yes         10.31.183.7
```

The following example browses vCenter 10.31.183.228 for virtual machines in datacenter VAAYU-DEV-WIN Hosts and Clusters view, and formats the output as XML:

```
mccli vcenter browse --name=10.31.183.228 --datacenter=VAAYU-DEV-WIN
    --esx-host=10.31.183.7 --vsphere-hosts-clusters-view=true --xml
```

```
<CLIOutput>
    <Results>
        <ReturnCode>1</ReturnCode>
        <EventCode>23999</EventCode>
        <EventSummary />
    </Results>
    <Data>
        <Row>
            <Name>ACMCommunity</Name>
            <GuestOS>debian5Guest</GuestOS>
            <Server>10.31.183.7</Server>

    <Location>/VAAYU-DEV-WIN/host/10.31.183.7/10.31.183.7/ACMCommunity<
    /Location>
            <Template>No</Template>
            <PoweredOn>Yes</PoweredOn>
            <ChangedBlock>No</ChangedBlock>
            <Protected>Yes</Protected>
        </Row>
        <Row>
            <Name>RHEL564BUILDDOWNLOAD224</Name>
            <GuestOS>rhel5_64Guest</GuestOS>
            <Server>10.31.183.7</Server>

    <Location>/VAAYU-DEV-WIN/host/10.31.183.7/10.31.183.7/RHEL564BUILDD
    OWNLOAD224</Location>
            <Template>No</Template>
            <PoweredOn>Yes</PoweredOn>
            <ChangedBlock>No</ChangedBlock>
            <Protected>No</Protected>
        </Row>
    </Data>
```

version

The **mccli version** resource comprises a single command, **mccli version show,** which returns the version of **mccli** currently installed and then exits.

version show

The **mccli version show** command returns the version of **mccli** currently installed and then exits.

Synopsis

```
mccli version show
```

INDEX

U

V