

EMC[®] Avamar[®] 7.1
Management Console
Command Line Interface (MCCLI)
Programmer Guide

P/N 302-000-833
REV 01

Copyright © 2001 - 2014 EMC Corporation. All rights reserved. Published in the USA.

Published June, 2014

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to the technical documentation and advisories section on the EMC online support website.

CONTENTS

Preface	
Chapter 1	Introduction
	System requirements 16
	Capabilities and limitations 16
Chapter 2	Installation and Configuration
	Install the Java Runtime Environment 20
	Install the MCCLI RPM package 21
	Run avsetup_mccli 22
	Additional configuration 23
	mccli.xml preferences file 23
	mcclimcs.xml default options file 24
	Remote clients must run mccli as root 26
	Installing both Avamar Administrator and MCCLI on the same Linux client... 26
Chapter 3	Command Reference
	General programming notes 28
	Command line elements 28
	Data types 28
	Default values 28
	Optional and required arguments 29
	Pattern matching 29
	Output description 30
	Global command line syntax and behavior 31
	Deprecated resources and commands 34
	activity 35
	activity cancel 35
	activity get-log 36
	activity show 37
	agent 39
	agent show 39
	backup 39
	backup delete 39
	backup edit 41
	backup restore 43
	backup show 48
	backup validate 51
	checkpoint 53
	checkpoint cancel-validate 53
	checkpoint create 53
	checkpoint delete 54
	checkpoint show 54
	checkpoint validate 55
	client 56
	Enumerating client resources 56
	client add 57

client add-datastore	64
client backup-dataset	65
client backup-group-dataset	67
client backup-target	69
client delete	71
client edit.....	72
client import-clients-from-file	77
client invite	78
client move	79
client remove-datastore	80
client retire.....	81
client show	82
client show-datastore.....	85
client show-plugins	86
client validate-clients-from-file	87
dataset	88
dataset add.....	88
dataset add-exclude	89
dataset add-include	90
dataset add-option	91
dataset add-target	92
dataset copy	93
dataset delete.....	94
dataset delete-exclude.....	95
dataset delete-include	96
dataset delete-option.....	97
dataset delete-target.....	98
dataset edit-option	99
dataset replace	100
dataset show	102
dd.....	104
dd add	104
dd delete	106
dd edit	107
dd show-prop.....	109
dd show-util.....	110
domain	111
domain add	111
domain delete.....	112
domain edit	113
domain show	114
dump.....	115
dump clientcache	115
dump domaincache	115
dump jobcache	115
event	116
event ack	116
event clear-data-integrity-alerts.....	119
event get-info.....	119
event show	120
group.....	123
Enumerating group resources.....	123
group add	124
group add-client.....	126
group add-proxy.....	127

- group backup..... 128
- group copy..... 129
- group delete..... 130
- group edit 131
- group export 133
- group move-client 134
- group remove-client 135
- group remove-proxy 136
- group replicate..... 137
- group show 138
- group show-members 140
- help..... 142
 - help 142
- mcs..... 143
 - mcs import..... 143
 - mcs list 144
 - mcs reboot-proxy 144
 - mcs resume-scheduler 145
 - mcs scheduler-status 145
 - mcs stop 145
 - mcs suspend-scheduler 145
 - mcs waitforflushcomplete 146
- plugin 147
 - plugin show 147
 - plugin update 147
- retention..... 148
 - Advanced retention calendar units 148
 - retention add 149
 - retention copy..... 151
 - retention delete 152
 - retention edit 153
 - retention show 154
- schedule..... 156
 - schedule add 156
 - schedule copy..... 158
 - schedule delete 159
 - schedule edit 160
 - schedule show 162
 - schedule show-timezones..... 163
- server 164
 - server show-prop 164
 - server show-services 168
 - server show-util 168
 - server start-service 169
 - server stop-service 169
- user 170
 - Enumerating user resources 170
 - user add 171
 - user authenticate 174
 - user delete..... 175
 - user edit 176
 - user show 178
 - user show-auth 178
- vcenter..... 179
 - vcenter browse..... 179

Contents

version.....	183
version show.....	183
vmcache	184
vmcache show	184
vmcache sync	185

TABLES

	Title	Page
1	Revision history	11
2	Minimum requirements for MCCLI software	16
3	MCCLI essential files	22
4	MCCLI default paths	22
5	mccli.xml file parameters	24
6	Data types for mccli commands.....	28
7	Pattern matching operator capabilities and limitations.....	29
8	Resources for the mccli command	31
9	Global options for mccli	32
10	Display options for mccli	32
11	Returned values in default and normalized formats.....	33
12	Deprecated resources	34
13	Deprecated commands	34
14	Command options for mccli activity cancel	35
15	Command options for mccli activity get-log	36
16	Command options for mccli activity show	37
17	Command options for mccli backup delete	39
18	Command options for mccli backup edit	41
19	Command options for mccli backup restore	43
20	Command options for mccli backup show	48
21	Command options for mccli backup validate	51
22	Command options for mccli checkpoint create	53
23	Command options for mccli checkpoint delete	54
24	Command options for mccli checkpoint show	54
25	Command options for mccli checkpoint validate	55
26	Command options for mccli client add	57
27	Command options for mccli client add-datastore	64
28	Command options for mccli client backup-dataset	65
29	Command options for mccli client backup-group-dataset	67
30	Command options for mccli backup-target	69
31	Command options for mccli client delete	71
32	Command options for mccli client edit	72
33	Command options for mccli client import-clients-from-file	77
34	Command options for mccli client invite	78
35	Command options for mccli client move	79
36	Command options for mccli client remove-datastore	80
37	Command options for mccli client retire	81
38	Command options for mccli client show	82
39	Command options for mccli client show-datastore	85
40	Command options for mccli client show-plugins	86
41	Command options for mccli client validate-clients-from-file	87
42	Command options for mccli dataset add	88
43	Command options for mccli dataset add-exclude	89
44	Command options for mccli dataset add-include	90
45	Command options for mccli dataset add-option	91
46	Command options for mccli dataset add-target	92
47	Command options for mccli dataset copy	93
48	Command options for mccli dataset delete	94
49	Command options for mccli dataset delete-exclude	95
50	Command options for mccli dataset delete-include	96

51	Command options for mccli dataset delete-option	97
52	Command options for mccli dataset delete-target	98
53	Command options for mccli dataset edit-option	99
54	Command options for mccli dataset replace	100
55	Command options for mccli dataset show	102
56	Command options for mccli dd add	104
57	Command options for mccli dd delete	106
58	Command options for mccli dd edit	107
59	Command options for mccli dd show-prop	109
60	Command options for mccli dd show-util	110
61	Command options for mccli domain add	111
62	Command options for mccli domain delete	112
63	Command options for mccli domain edit	113
64	Command options for mccli domain show	114
65	Command options for mccli event ack	116
66	Command options for mccli event clear-data-integrity-alerts	119
67	Command options for mccli event get-info	119
68	Command options for mccli event show	120
69	Command options for mccli group add	124
70	Command options for mccli group add-client	126
71	Command options for mccli group add-proxy	127
72	Command options for mccli group backup	128
73	Command options for mccli group copy	129
74	Command options for mccli group delete	130
75	Command options for mccli group edit	131
76	Command options for mccli group export	133
77	Command options for mccli group move-client	134
78	Command options for mccli group remove-client	135
79	Command options for mccli group remove-proxy	136
80	Command options for mccli group replicate	137
81	Command options for mccli group show	138
82	Command options for mccli group show-members	140
83	Command options for mccli mcs import	143
84	Command options for mccli mcs list	144
85	Command options for mccli mcs reboot-proxy	144
86	Advanced retention option string values	148
87	Command options for mccli retention add	149
88	Command options for mccli retention copy	151
89	Command options for mccli retention delete	152
90	Command options for mccli retention edit	153
91	Command options for mccli retention show	154
92	Command options for mccli schedule add	156
93	Command options for mccli schedule copy	158
94	Command options for mccli schedule delete	159
95	Command options for mccli schedule edit	160
96	Command options for mccli schedule show	162
97	Output attributes and possible values for mccli schedule show	163
98	Command options for mccli server show-prop	164
99	Output attributes and possible values for mccli server show-prop	166
100	Command options for mccli server show-services	168
101	Command options for mccli server show-util	168
102	Command options for mccli server start-service	169
103	Command options for mccli server stop-service	169
104	Command options for mccli user add	171
105	Command options for mccli user authenticate	174

106	Command options for mccli user delete	175
107	Command options for mccli user edit	176
108	Command options for mccli user show	178
109	Command options for mccli vcenter browse	179
110	Command options for mccli vmcache show	184
111	Command options for mccli vmcache sync	185

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 technical support professional if a product does not function properly or does not function as described in this document.

Note: This document was accurate at publication time. Go to EMC Online Support (<https://support.emc.com>) 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

Revision	Date	Description
01	June 11, 2014	Initial release of Avamar 7.1.

Related documentation

The following EMC publications provide additional information:

- ◆ *EMC Avamar Administration Guide*
- ◆ *EMC Avamar Operational Best Practices*
- ◆ *EMC Avamar and EMC Data Domain System Integration Guide*
- ◆ *EMC Avamar Release Notes*

Conventions used in this document

EMC uses the following conventions for special notices:

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:

Bold	Use for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks)
<i>Italic</i>	Use for full titles of publications referenced in text
Monospace	Use for: <ul style="list-style-type: none"> • System output, such as an error message or script • System code • Pathnames, filenames, prompts, and syntax • Commands and options
<i>Monospace italic</i>	Use for variables.
Monospace bold	Use for user input.
[]	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 Support.

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.

Online communities

Visit EMC Community Network at <https://community.EMC.com> for peer contacts, conversations, and content on product support and solutions. Interactively engage online with customers, partners and certified professionals for all EMC products.

Live chat

To engage EMC Customer Support 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 Support, 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 Support.

Your comments

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

DPAD.Doc.Feedback@emc.com

Please include the following information:

- ◆ Product name and version
- ◆ Document name, part number, and revision (for example, 01)
- ◆ 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 provide an introduction to the MCCLI:

- ◆ [System requirements](#) 16
- ◆ [Capabilities and limitations](#) 16

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	<ul style="list-style-type: none"> • Red Hat Enterprise Linux Release 4 (64-bit) • SUSE Linux Enterprise Server 11 (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. 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 import-clients-from-file** command and **client validate-clients-from-file** command 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 approximately 8 seconds to complete. This elapsed time is primarily because the Java Virtual Machine (JVM) must be started to run any MCCLI command, and because decryption framework initialization adds approximately 6 seconds.

If you provide the `--mcspasswd` option directly with each command, you can bypass the decryption framework initialization, which will improve performance.

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. Therefore, when these errors and failure conditions occur, the generic event code for a failed **mccli** command 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](#) 20
- ◆ [Install the MCCLI RPM package](#) 21
- ◆ [Run avsetup_mccli](#) 22
- ◆ [Additional configuration.....](#) 23
- ◆ [Installing both Avamar Administrator and MCCLI on the same Linux client](#) 26

Install the Java Runtime Environment

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

To install the Java Runtime Environment:

1. On the computer where MCCLI will be installed, open a web browser and type the following URL:

```
http://Avamar-server
```

where *Avamar-server* is the Avamar server network hostname or IP address.

The EMC Avamar Web Restore web page appears.

2. Click **Downloads**.
3. Click **+** next to the **Linux for x86 (64 bit)** folder.
4. Click **+** next to the correct operating system version folder.
5. Locate the JRE RPM install package (it is typically the last entry in the folder).
6. If the JRE on the client computer is older than the JRE hosted on the Avamar server, download the **jre-version-platform.rpm** install package to a temporary install folder such as */tmp*.

where *version* is the JRE version, and *platform* is the computing platform.

7. Open a command shell and log in as root.
8. Change directory to the temporary install folder used in step 6. For example:

```
cd /tmp
```

9. Install the JRE by typing:

```
rpm -ivh jre-version-platform.rpm
```

10. 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. On the computer where MCCLI will be installed, open a web browser and type the following URL:

```
http://Avamar-server
```

where *Avamar-server* is the Avamar server network hostname or IP address.

The EMC Avamar Web Restore web page appears.

2. Click **Downloads**.
3. Click **+** next to the **Linux for x86 (64 bit)** folder.
4. Do one of the following:
 - If installing on SUSE Linux Enterprise Server 11, click **+** next to the **SUSE Linux Enterprise Server 11** folder.
 - If installing on Red Hat Enterprise Linux Release 4, click **+** next to the **Red Hat Enterprise Linux 4** folder.
5. Download the **dpmccli-version.platform.x86_64.rpm** install package to a temporary install folder such as */tmp*.

where *version* is the MCCLI software version, and *platform* is either **sles11_64** or **rhel4_64**.

6. Open a command shell and log in as root.
7. Change directory to the temporary install folder. For example:

```
cd /tmp
```

8. Install the MCCLI RPM by typing:

```
rpm -ivh dpmccli-version.platform.x86_64.rpm
```

The following information appears in the command shell:

```
Preparing... ##### [100%]
1:dpmccli      ##### [100%]
Please run /usr/local/avamar/bin/avsetup_mccli to configure MCLI
```

9. 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. When you specify these values during the interactive **avsetup_mccli** session, it automatically updates the mcclimcs.xml default options file.

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

1. Open a command shell and log in as root.
2. Type:

```
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. 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 -->

<MCSCConfig 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 -->
</MCSCConfig>

```

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 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 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 operating 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.

Installing both Avamar Administrator and MCCLI on the same Linux client

If you need to install MCCLI on the same Linux client that already has Avamar Administrator installed, you must include the `--force` option when installing the MCCLI RPM. For example:

```
rpm -ivh --force dpmccli-version.platform.x86_64.rpm
```

where *version* is the MCCLI software version, and *platform* is either `sles11_64` or `rhel4_64`.

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	28
◆ Deprecated resources and commands.....	34
◆ activity.....	35
◆ agent	39
◆ backup	39
◆ checkpoint.....	53
◆ client	56
◆ dataset	88
◆ dd.....	104
◆ domain	111
◆ dump.....	115
◆ event	116
◆ group.....	123
◆ help.....	142
◆ mcs.....	143
◆ plugin	147
◆ retention.....	148
◆ schedule.....	156
◆ server	164
◆ user	170
◆ vcenter.....	179
◆ version.....	183
◆ vmcache	184

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
- ◆ A command
- ◆ Global options
- ◆ Command options
- ◆ Display options

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.

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 [--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, <code>/usr/*</code> matches the contents of <code>/usr</code> but not the contents of <code>/usr/bin</code> or <code>/usr/local</code> .
Double asterisk (**)	Matches zero or more occurrences of any character. This correlates to conventional single asterisk regex behavior. For example, <code>/usr/**</code> matches the entire <code>/usr</code> 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: <ul style="list-style-type: none"> • <code>[0-9]</code> matches any single numeric character • <code>[a-z]</code> 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 [COMMAND-OPTIONS] [GLOBAL-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 35
agent	Client agent management	“agent” on page 39
backup	Backup management	“backup” on page 39
checkpoint	Checkpoint management	“checkpoint” on page 53
client	Client management	“client” on page 56
dataset	Dataset management	“dataset” on page 88
dd	Data Domain system configuration and monitoring	“dd” on page 104
domain	Domain management	“domain” on page 111
dump	Strictly reserved for EMC internal use only	“dump” on page 115
event	Event code information	“event” on page 116
group	Group management	“group” on page 123
help	Help information	“help” on page 142
mcs	Administrator server management	“mcs” on page 143
plugin	Client plug-in management	“plugin” on page 147
retention	Retention policy management	“retention” on page 148
schedule	Schedule management	“schedule” on page 156
server	Server monitoring	“server” on page 164
user	User management	“user” on page 170
vcenter	vCenter management	“vcenter” on page 179
version	Version information	“version” on page 183
vmcache	Strictly reserved for EMC internal use only	“vmcache” on page 184

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.

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.

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]}
```

The following table provides details on each global option.

Table 9 Global options for **mccli**

Global option	Description
<code>--mcsprofile=String</code>	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.
<code>--mcsaddr=String</code>	Specifies the MCS network name or IP address.
<code>--mcspasswd=String</code>	Specifies the <code>--mcsuserid</code> account password.
<code>--mcsport=Integer</code>	Specifies the data port used to contact MCS.
<code>--mcsuserid=String</code>	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. You only supply the global options on an **mccli** command line in cases where the persistent settings must be temporarily overridden.

Display options

Specify the following display options after the command options.

Table 10 Display options for **mccli**

Display option	Description
<code>--normalize</code>	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.
<code>--xml</code>	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.

Deprecated resources and commands

Deprecated resources and commands will continue to work, but should not be used for new integrations. If documentation for these deprecated resources and commands is required, refer to previous versions of this publication. The following tables list deprecated resources and commands:

Table 12 Deprecated resources

Deprecated resource	Use instead
snapup	“backup” on page 39

Table 13 Deprecated commands

Deprecated command	Use instead
client activate	“client invite” on page 78
client load-bulk	“client import-clients-from-file” on page 77
client snapup-dataset	“client backup-dataset” on page 65
client snapup-group-dataset	“client backup-group-dataset” on page 67
client snapup-target	“client backup-target” on page 69
client validate-bulk	“client validate-clients-from-file” on page 87
group show-client-members	“group show-members” on page 140
group snapup	“group backup” on page 128

activity

The **mccli activity** resource is used to cancel or show backup, restore, or validation activities. Commands are:

- ◆ “[activity cancel](#)” on page 35
- ◆ “[activity get-log](#)” on page 36
- ◆ “[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 --id=String [--wait[=MIN]]
```

Command options

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

Option	Description
<code>--id=String</code>	Specifies which activity to cancel. String must be a valid activity ID. This argument is required. Use mccli activity show to return a list of valid activity IDs, or use the activity ID returned by the backup, restore, or validation command.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

- 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 --id=String
```

Command options

Table 15 Command options for **mccli activity get-log**

Option	Description
<code>--id=String</code>	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 or any MCCLI issued backup or restore job.

Event codes

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 [--active=Boolean(false)]
  [--completed=Boolean(false)]
  [--contained-vm-activities=Boolean(false)] [--domain=String]
  [--id=String] --name=String [--queued=Boolean(false)]
  [--source=String] [--verbose=Boolean(false)]
```

Command options

Table 16 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.
--contained-vm-activities=Boolean(false)	If true, shows activities for virtual machine clients within VMware containers or vApps. This option is only valid if the client specified by the --name argument is a VMware container or vApp.
--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: <ul style="list-style-type: none"> • 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.

Examples

This command returns activities for virtual machine clients in the 10.31.183.55/F02 container, with output formatted as XML:

```
mccli activity show --name=/10.31.183.55/F02 --contained-vm-activities=true --verbose=true --xml
```

```
<CLIOutput>
  <Results>
    <ReturnCode>0</ReturnCode>
    <EventCode>23000</EventCode>
    <EventSummary>CLI command completed successfully.</EventSummary>
  </Results>
  <Data>
    <Row>
      <ID>9134224603666509</ID>
      <Status>Completed</Status>
      <ErrorCode>0</ErrorCode>
      <StartTime>2013-07-14 06:07 UTC</StartTime>
      <Elapsed>00h:00m:46s</Elapsed>
      <EndTime>2013-07-14 06:08 UTC</EndTime>
      <Type>On-Demand Backup</Type>
      <ProgressBytes>4,200,757</ProgressBytes>
      <NewBytes>0.1%</NewBytes>
      <Client>TEST1</Client>
      <Domain>/10.31.183.55/HleDynamicClients</Domain>
      <OS>windows7Server64Guest</OS>
      <ClientRelease>7.1.100-333</ClientRelease>
      <Sched.StartTime>2013-07-14 06:07 UTC</Sched.StartTime>
      <Sched.EndTime>2013-07-15 06:07 UTC</Sched.EndTime>
      <ElapsedWait>00h:00m:29s</ElapsedWait>
      <Group>Admin On-Demand Group</Group>
      <Plug-In>Windows VMware Image</Plug-In>
      <RetentionPolicy>Default Retention</RetentionPolicy>
      <Retention>N</Retention>
      <Schedule>Admin On-Demand Schedule</Schedule>
      <Dataset>/Client On-Demand Data</Dataset>
      <WID>MOD-1342246036652_42992503ce8f11c007c26d73711fac08f5d5e331</WID>
      <Server>Avamar</Server>
      <Container>F02</Container>
    </Row>
  </Data>
</CLIOutput>
```

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
```

backup

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

- ◆ [“backup delete” on page 39](#)
- ◆ [“backup edit” on page 41](#)
- ◆ [“backup restore” on page 43](#)
- ◆ [“backup show” on page 48](#)
- ◆ [“backup validate” on page 51](#)

backup delete

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

Synopsis

```
mccli backup delete [--contained-vm-name=String] [--created=String]
  [--domain=String(/)] [--force=Boolean(false)] [--labelNum=Integer]
  --name=String
```

Command options

Table 17 Command options for **mccli backup delete** (page 1 of 2)

Option	Description
<code>--contained-vm-name=String</code>	Specifies a virtual machine client within a VMware container or vApp. This option is only valid if the client specified by the name <code>--name</code> argument is a VMware container or vApp.
<code>--created=String</code>	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.
<code>--force=Boolean(false)</code>	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.
<code>--labelNum=Integer</code>	Specifies the label number of the backup to delete.

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

Option	Description
<code>--name=String</code>	Specifies the client from which the backup was originally taken. This argument is required. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--domain</code> argument is ignored.
<code>[--domain=String(/)]</code>	Specifies the Avamar server domain that contains the client specified by the <code>--name</code> argument.

Event codes

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

Examples

This command deletes the first backup (`--labelNum=1`) created on 2013-07-14 for virtual machine client TEST1 in the 10.31.183.55/FO2 container:

```
mccli backup delete --name=/10.31.183.55/FO2 --contained-vm-name=TEST1  
--created="2013-07-14" --labelNum=1
```

```
0,22553,Backup deleted.
```

```
Attribute      Value
```

```
-----  
labelnum      1  
createtime    2013-07-14 06:08:21 UTC  
path          /10.31.183.55/HleDynamicClients/UCKZ3FGLz70WRRu3nq2ZuA  
retention     N  
plugin        3016
```


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 [--contained-vm-name=String] [--created=String]
  [--domain=String(/)] [--expiration=String |
  --extend-expiration=String | --retention=String]
  [--force=Boolean(false)] [--labelNum=Integer] --name=String
```

Command options

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

Option	Description
<code>--contained-vm-name=String</code>	Specifies a virtual machine client within a VMware container or vApp. This option is only valid if the client specified by the name <code>--name</code> argument is a VMware container or vApp.
<code>--created=String</code>	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.
<code>[--domain=String(/)]</code>	Specifies the Avamar server domain that contains the client specified by the <code>--name</code> argument.
<code>--expiration=String</code>	Specifies a new expiration date. String must be of the following: <ul style="list-style-type: none"> • YYYY-MM-DD is a specific year (YYYY), month (MM), and day (DD). • NO_EXPIRATION specifies that the backup should never expire. The <code>--expiration</code> , <code>--extend-expiration</code> , and <code>--retention</code> options are mutually exclusive.
<code>--extend-expiration=String</code>	Extends the existing expiration date. String must be in the format of <code>+nn{D W M Y}</code> , where: <ul style="list-style-type: none"> • <code>+nnD</code> is the number (nn) of additional days (d) to be added to the existing backup expiration date • <code>+nnW</code> is the number (nn) of additional weeks (w) to be added to the existing backup expiration date • <code>+nnM</code> is the number (nn) of additional months (m) to be added to the existing backup expiration date • <code>+nnY</code> is the number (nn) of additional years to be added to the existing backup expiration date The <code>--expiration</code> , <code>--extend-expiration</code> , and <code>--retention</code> options are mutually exclusive.

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

Option	Description
<code>--force=Boolean(false)</code>	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.
<code>--labelNum=Integer</code>	Specifies the label number of the backup to change.
<code>--name=String</code>	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.
<code>--retention=String</code>	Specifies the retention types to assign to the backup. String must be one or more of the following values: <ul style="list-style-type: none"> • 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: <ul style="list-style-type: none"> • --retention=D,weekly • --retention=none • --retention=Daily,W,monthly The --expiration, --extend-expiration, and --retention options are mutually exclusive.

Event codes

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.

Examples

This command changes backup retention to daily for the second backup (--labelNum=2) created on 2013-07-14 for virtual machine client TEST1 in the 10.31.183.55/FO2 container:

```
mccli backup edit --name=/10.31.183.55/FO2 --contained-vm-name=TEST1
--labelNum=2 --created=2013-07-14 --retention=D

0,22556,Changed backup retention.
Attribute      Value
-----
labelnum      2
createtime    2013-07-14 09:11:41 UTC
path          /10.31.183.55/HleDynamicClients/UCKZ3FGLz70WRRu3nq2ZuA
retention     D
plugin        3016
```

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** to get status for the restore activity.

This command supports any of the following restore scenarios:

- ◆ If restoring from a 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 a 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 [--contained-vm-name=String] [--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] [--esx-host=String]
[--folder=String] --labelNum=Integer --name=String
[--network-old=String --network-new=String] [--plugin=Integer]
[--resource-pool=String] [--restore-vm-to=String]
[--virtual-center-name=String] [--vm-configuration=Boolean(false)]
[--wait[=MIN]]
```

Command options

Table 19 Command options for **mccli backup restore** (page 1 of 4)

Option	Description
<code>--cmd=String</code>	Specifies one or more optional plug-in commands. You can supply multiple <code>--cmd</code> arguments, but each argument can only specify one plug-in command. For example, this is valid <code>--cmd</code> syntax: <code>--cmd="verbose=5" --cmd="throttle=5"</code> However, this is not valid <code>--cmd</code> syntax: <code>--cmd="verbose=5 throttle=5"</code>
<code>--contained-vm-name=String</code>	Specifies a virtual machine client within a VMware container or vApp. This option is only valid if the client specified by the <code>--name</code> argument is a VMware container or vApp.

Table 19 Command options for **mccli backup restore** (page 2 of 4)

Option	Description
<code>--data=String</code>	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: <ul style="list-style-type: none"> All—restores all virtual disks 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.
<code>--datacenter=String</code>	When restoring a virtual machine or vApp, specifies a fully qualified datacenter name in vCenter. 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.
<code>--datastore-name=String</code>	When restoring a virtual machine or vApp to a new virtual machine, specifies the datastore name to be removed from the Avamar proxy. This argument is required.
<code>--dest-client-domain=String</code>	Specifies the Avamar server domain that contains the alternative client specified by the --dest-client-name argument.
<code>--dest-client-name=String</code>	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.
<code>--dest-client-password=String</code>	When restoring folders or files from a VMware image backup, specifies the destination client password.
<code>--dest-client-username=String</code>	When restoring folders or files from a VMware image backup, specifies the destination client username.
<code>--dest-dir=String</code>	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.
<code>--domain=String (/)</code>	Specifies the Avamar server domain that contains the client specified by the --name argument.
<code>--esx-host-name=String</code>	When restoring a VMware image backup to a new virtual machine, specifies a fully qualified ESX server hostname in a datacenter.
<code>--esx-host=String</code>	When restoring a VMware image backup to a new vApp, specifies the path to a host or cluster inside the datacenter.
<code>--folder=String</code>	When restoring a virtual machine or vApp to an existing or new virtual machine, specifies the folder path for the destination virtual machine in the datacenter.

Table 19 Command options for **mccli backup restore** (page 3 of 4)

Option	Description
<code>--labelNum=Integer</code>	Specifies the backup to use for the restore. This argument is required. Use mccli backup show to return a list of valid backup label numbers.
<code>--name=String</code>	Specifies the target client for the restore. String must be a valid client name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the <code>--domain</code> option is ignored.
<code>--network-old=String</code> <code>--network-new=String</code>	When restoring restoring a VMware image backup to a new vApp, these options map old network setting to a new network setting. Both options must be supplied. Multiple <code>--network-old</code> and <code>--network-new</code> pairs are allowed. Multiple comma-separated mapping values are allowed for each option. Example one-to-one mapping: <code>--network-old=vlan1 --network-new=vlan241</code> Example multiple one-to-one mappings: <code>--network-old=vlan1 --network-new=vlan241</code> <code>--network-old=vlan2 --network-new=vlan242</code> <code>--network-old=vlan3 --network-new=vlan243</code> Example many-to-one mapping: <code>--network-old=vlan1, vlan2, vlan3</code> <code>--network-new=vlan241</code> Example combined many-to-one with one-to-one mapping: <code>--network-old=vlan1, vlan2</code> <code>--network-new=vlan241</code> <code>--network-old=vlan3 --network-new=vlan242</code>
<code>--plugin=Integer</code>	Specifies the plug-in ID for the restore. This argument is required for all restores except vApps. Use mccli plugin show 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. When restoring vApps, multiple <code>---</code> plugin and <code>--cmd</code> arguments are allowed. For example: <code>--plugin=3016 --cmd=--test=true</code> <code>--cmd=--test1=false --plugin=1016</code> <code>--cmd=--test3=true</code> In this example, <code>--test</code> and <code>--test1</code> are processed by plug-in 3016; <code>--test3</code> is processed by plug-in 1016.
<code>--resource-pool=String</code>	When restoring restoring a VMware image backup to a new vApp, specifies a resource pool on the ESX host.

Table 19 Command options for **mccli backup restore** (page 4 of 4)

Option	Description
<code>--restore-vm-to=String</code>	When restoring a VMware image backup, specifies the type of restore operation. String must be one of the following: <ul style="list-style-type: none"> • <code>original</code>—restores the entire backup image to the original virtual machine or vApp • <code>existing</code>—restores the entire backup image to a different existing virtual machine • <code>new</code>—restores the entire backup image to a new virtual machine or vApp • <code>flr</code>—restores folders or files
<code>--virtual-center-name=String</code>	When restoring a VMware image backup to an existing or new virtual machine or vApp, specifies the vCenter name.
<code>--vm-configuration=Boolean(false)</code>	When restoring a VMware image backup to an existing virtual machine or vApp, and this argument is true, VMware configuration files are restored.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

22236 Client does not exist.
22297 Request to restore is rejected.
22312 Client restore scheduled.
23009 Invalid plugin specified on the CLI.

Examples

This command 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
```

This command 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
```

This command 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=vcenter-1.example.com
--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
```

This command 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
```

This command restores a vApp to the original vApp:

```
mccli backup restore --name=/10.31.183.55/ABCD1
--contained-vm-name=YYY --labelNum=1 --restore-vm-to=original
--vm-configuration=true
0,22312,client restore scheduled.
Attribute      Value
-----
client         /10.31.183.55/HleDynamicClients/YYY_EDEYNV5CM
activity-id    9134226661172409
```

This command restores a vApp to the original vApp with plug-in command line options:

```
mccli backup restore --name=/10.31.183.55/ABCD1
--contained-vm-name=YYY --labelNum=1 --restore-vm-to=original
--plugin=3016 --cmd=---allnodes=false --cmd=---test=true
--plugin=1016 --cmd=---allnodes=true
0,22312,client restore scheduled.
Attribute      Value
-----
client         /10.31.183.55/HleDynamicClients/YYY_EDEYNV5CM
activity-id    9134226675354509
```

This command restores a vApp to a new vApp:

```
mccli backup restore --name=/10.31.183.55/ABCD1
--contained-vm-name=YYY --labelNum=1 --restore-vm-to=new
--datacenter=DCF1/DCF2 --virtual-center-name=10.31.183.55
--datastore-name=datastore1 --esx-host=HOF1/HOF2/10.31.183.17
--dest-clientname=MyRestorevApp --folder=FO1/FO2
--resource-pool=RP1/RP2 --network-old=VM Network
--network-new=VM Network
0,22312,client restore scheduled.
Attribute      Value
-----
client         /10.31.183.55/HleDynamicClients/YYY_EDEYNV5CM
activity-id    9134226696110509
```

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 [--after=String --before=String]
  [--contained-vm-name=String] [--dir=String(/)] [--domain=String(/)]
  [--labelNum=Integer] --name=String [--recursive=Boolean(false)]
  [--retention=String] [--verbose=Boolean(false)]
```

Command options

Table 20 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.
--contained-vm-name=String	Specifies a virtual machine client within a VMware container or vApp. This option is only valid if the client specified by the --name argument is a VMware container or vApp.
--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.
--recursive=Boolean(false)	If true, backup folders and files are recursively shown.

Table 20 Command options for **mccli backup show** (page 2 of 2)

Option	Description
<code>--retention=String</code>	<p>If supplied, then only backups with the specified retention types are returned. String must be one or more of the following:</p> <ul style="list-style-type: none"> • D or daily • W or weekly • M or monthly • Y or yearly • N or none <p>Both short form and long form retention type values are allowed and can be mixed. For example, all of the following are valid:</p> <pre>--retention=D,weekly --retention=none --retention=Daily,W,monthly</pre>
<code>--verbose=Boolean(false)</code>	If true, then detailed information is returned.

Event codes

22236 Client does not exist.

22504 Failed to retrieve the backups for a client.

Examples

This command returns a list of backups for MyClient.example.com:

```
mccli backup show --name=clients/MyClient.example.com
```

```
0,23000,CLI command completed successfully.
Created                      LabelNum  Size
-----
2013-1-10 15:51:30 PST      2         4767841280
2013-1-10 15:00:23 PST      1         4750878720
```

This command returns backups for virtual machine TEST1, which resides inside VMware container 10.31.183.55/F02:

```
mccli backup show --name=/10.31.183.55/F02 --verbose=true
--contained-vm-name=TEST1 --xml
```

```
<CLIOutput>
  <Results>
    <ReturnCode>0</ReturnCode>
    <EventCode>23000</EventCode>
    <EventSummary>CLI command completed successfully.</EventSummary>
  </Results>
  <Data>
    <Row>
      <Created>2013-07-14 06:08:21 UTC</Created>
      <LabelNum>1</LabelNum>
      <Size>4200757</Size>
      <Retention>N</Retention>
      <Label>MOD-134224603642992503ce8f11c007c26d73711fac</Label>
      <Plugin>Windows VMware Image</Plugin>
      <Expires>2013-09-12 06:14:35 UTC</Expires>
      <Files />
      <Server>Avamar</Server>
    </Row>
  </Data>
</CLIOutput>
```

This command returns backups for vApp 10.31.183.55/APP1:

```
mccli backup show --name=/10.31.183.55/APP1 --verbose=true --xml
```

```
<CLIOutput>
  <Results>
    <ReturnCode>0</ReturnCode>
    <EventCode>23000</EventCode>
    <EventSummary>CLI command completed successfully.</EventSummary>
  </Results>
  <Data>
    <Row>
      <Created>2013-07-14 06:42:46 UTC</Created>
      <LabelNum>1</LabelNum>
      <Size>4204699</Size>
      <Retention>N</Retention>
      <Label>MOD-134224603642992503ce8f11c007c26d73711fac</Label>
      <Plugin />
      <Expires>2013-09-12 06:49:40 UTC</Expires>
      <Files />
      <Server>Avamar</Server>
      <Networks>VM Network,Test2</Networks>
    </Row>
  </Data>
</CLIOutput>
```

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** to get status for this validation activity.

Synopsis

```
mccli backup validate [--cmd=String [--cmd=String] ...]
  [--dest-client-domain=String --dest-client-name=String]
  [--domain=String(/)] --labelNum=Integer --name=String
  --plugin=Integer [--wait[=MIN]]
```

Command options

Table 21 Command options for **mccli backup validate** (page 1 of 2)

Option	Description
<code>--cmd=String</code>	Specifies one or more optional plug-in commands. You can supply multiple <code>--cmd</code> arguments, but each argument can only specify one plug-in command. For example, this is valid <code>--cmd</code> syntax: <code>--cmd="verbose=5" --cmd="throttle=5"</code> However, this is not valid <code>--cmd</code> syntax: <code>--cmd="verbose=5 throttle=5"</code>
<code>--dest-client-domain=String</code>	Used with <code>--dest-client-name</code> to validate a replicated backup (that is, a backup for a client in the <code>/REPLICATE</code> domain) on an alternate client. Specifies the Avamar server domain that contains the client specified by the <code>--dest-client-name</code> argument.
<code>--dest-client-name=String</code>	Used with <code>--dest-client-domain</code> to validate a replicated backup (that is, a backup for a client in the <code>/REPLICATE</code> domain) on an alternate client. Specifies the alternative target client for the validation. String must be a valid client name. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--dest-client-domain</code> argument is ignored. This argument returns an error if you supply anything other than a valid client name from the <code>/REPLICATE</code> domain.
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the client specified by the <code>--name</code> argument.
<code>--labelNum=Integer</code>	Specifies the backup to validate. This argument is required.
<code>--name=String</code>	Specifies the client associated with the backup to validate. String must be a valid client name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--domain</code> argument is ignored.

Table 21 Command options for **mccli backup validate** (page 2 of 2)

Option	Description
<code>--plugin=Integer</code>	Specifies the plug-in ID for the restore. This argument is required. Use mccli plugin show return a list of valid plug-in IDs.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

- 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. Commands are:

- ◆ “[checkpoint cancel-validate](#)” on page 53
- ◆ “[checkpoint create](#)” on page 53
- ◆ “[checkpoint delete](#)” on page 54
- ◆ “[checkpoint show](#)” on page 54
- ◆ “[checkpoint validate](#)” on page 55

checkpoint cancel-validate

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

Synopsis

```
mccli checkpoint cancel-validate
```

Event codes

- 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 [--wait [=MIN]]
```

Command options

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

Option	Description
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

- 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 --cptag=String
```

Command options

Table 23 Command options for **mccli checkpoint delete**

Option	Description
<code>--cptag=String</code>	Specifies the checkpoint to delete. String must be a valid checkpoint ID. This argument is required. Use mccli checkpoint show to return a list of valid checkpoint IDs.

Event codes

- 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 [--verbose=Boolean(false)]
```

Command options

Table 24 Command options for **mccli checkpoint show**

Option	Description
<code>--verbose=Boolean(false)</code>	If true, then detailed checkpoint information is returned. If false or not supplied, then summary information is returned.

Event codes

- 22531 Unexpected exception occurred.

checkpoint validate

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

Synopsis

```
mccli checkpoint validate [--checktype=TYPE] --cptag=String
  [--wait [=MIN]]
```

Command options

Table 25 Command options for **mccli checkpoint validate**

Option	Description
<code>--checktype=TYPE</code>	Constrains checkpoint validation to one or more of the following checkpoint TYPES: <ul style="list-style-type: none"> • full—Validation performed all checks. • rolling—Validation was rolling HFS check.
<code>--cptag=String</code>	Specifies the checkpoint to validate. String must be a valid checkpoint ID. This argument is required. Use mccli checkpoint show to return a list of valid checkpoint IDs.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

- 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. Commands are:

- ◆ “client add” on page 57
- ◆ “client add-datastore” on page 64
- ◆ “client backup-dataset” on page 65
- ◆ “client backup-group-dataset” on page 67
- ◆ “client backup-target” on page 69
- ◆ “client delete” on page 71
- ◆ “client edit” on page 72
- ◆ “client import-clients-from-file” on page 77
- ◆ “client invite” on page 78
- ◆ “client move” on page 79
- ◆ “client remove-datastore” on page 80
- ◆ “client retire” on page 81
- ◆ “client show” on page 82
- ◆ “client show-datastore” on page 85
- ◆ “client show-plugins” on page 86
- ◆ “client validate-clients-from-file” on page 87

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**
- ◆ **mccli retention show**
- ◆ **mccli plugin show**

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 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 [--changed-block-tracking=Boolean(false)]
  [--datacenter=String] [--contact=String] [--container-path=String]
  [--dataset-domain=String(/)] [--dataset=String] --domain=String(/)
  [--email=String] [--encryption=String(None)] [--folder=String]
  [--host=String] [--location=String] --name=String
  [--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)]
  [--vcontainer-inclusion=String] [--view-type=String]
  [--virtual-center-name=String] [--virtual-center-password=String]
  [--virtual-center-port=Integer] [--virtual-center-username=String]
```

Command options

Table 26 Command options for **mccli client add** (page 1 of 4)

Option	Description
<code>--changed-block-tracking=Boolean(false)</code>	If true, when adding a virtual machine, VMware container, or vApp client, changed block tracking is enabled.
<code>--contact=String</code>	Specifies contact information for the new client.
<code>--container-path=String</code>	When adding a VMware container or vApp, specifies the inventory path to that container.
<code>--datacenter=String</code>	When adding a virtual machine client, VMware container or vApp, specifies a fully qualified datacenter name in vCenter. This argument is required.
<code>--dataset-domain=String(/)</code>	Specifies the Avamar server domain that contains the dataset specified by the <code>--dataset</code> argument.
<code>--dataset=String</code>	Specifies an alternative dataset this client will use for on-demand backups, or when the group dataset is overridden.
<code>--domain=String(/)</code>	Specifies the domain for the new client.
<code>--email=String</code>	Specifies the email address for the primary contact (responsible party) for the client.

Table 26 Command options for `mccli client add` (page 2 of 4)

Option	Description
<code>--encryption=String</code>	<p>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:</p> <ul style="list-style-type: none"> • 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. <hr/> <p>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 <i>EMC Avamar Product Security Guide</i> provides additional information.</p> <hr/>
<code>--folder=String</code>	When adding a virtual machine client, specifies the folder path for virtual machines in the datacenter.
<code>--host=String</code>	If adding a VMware container or vApp, and the <code>--view-type=host-cluster</code> , this option specifies the host path inside the datacenter.
<code>--location=String</code>	Specifies location information for the new client.
<code>--name=String</code>	Specifies the new client name. This argument is required. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--domain</code> argument is ignored.
<code>--override-encryption=Boolean</code>	If true, then use the encryption method specified by <code>--encryption</code> instead of the group encryption method for scheduled backups.
<code>--override-retention=Boolean</code>	If true, then use the retention policy specified by <code>--retention</code> instead of the group retention policy for scheduled backups.

Table 26 Command options for `mccli client add` (page 3 of 4)

Option	Description
<code>--overtime-option=String</code>	Specifies scheduled group backup overtime behavior. String must be one of the following: <ul style="list-style-type: none"> • 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.
<code>--overtime=Boolean</code>	If true, client can exceed its backup window during scheduled backups.
<code>--pageable=Boolean</code>	If true, client can be paged for the purpose of initiating activation or picking up new backup or restore work.
<code>--pageaddr=String</code>	Specifies the IP address that the Avamar server can use to contact the client.
<code>--pageport=String</code>	Specifies the data port that the Avamar server can use to contact the client.
<code>--phone=String</code>	Specifies the voice telephone information for the primary contact (responsible party).
<code>--retention-domain=String (/)</code>	Specifies the Avamar server domain that contains the retention policy specified by the <code>--retention</code> argument.
<code>--retention=String</code>	Specifies an alternative retention policy this client will use for on-demand backups, or when the group retention policy is overridden.
<code>--type=String (normal)</code>	Specifies the type of client to add. String must be one of the following: <ul style="list-style-type: none"> • normal—any client that is not one of the other types. This is the default client type. • vcenter—vCenter client. • proxy—Avamar proxy. • vmachine—virtual machine client. • vcontainer—VMware container. • vapp—VMware vApp.

Table 26 Command options for `mccli client add` (page 4 of 4)

Option	Description
<code>--vcontainer-inclusion=String</code>	When adding a VMware container or vApp, specifies the inclusion settings. String must be one of the following: <ul style="list-style-type: none"> • <code>dynamic</code>—includes all contents of the vCenter container, but also continuously monitors the container entity in vCenter, so that if changes occur (for example, virtual machines or folders are added or deleted), those changes will be automatically reflected in Avamar. • <code>static</code>—only includes what is in the vCenter container at the time it is added to Avamar. If subsequent changes occur in vCenter, they will not be reflected in Avamar.
<code>--view-type=String</code>	When adding a VMware container or vApp, specifies which view will be used to locate the container or vApp inside vCenter. String must be one of the following: <ul style="list-style-type: none"> • <code>datastore</code>—Datastores view • <code>host-cluster</code>—Hosts and Clusters view • <code>vm-template</code>—Virtual Machines and Templates view
<code>--virtual-center-name=String</code>	When adding an Avamar proxy, specifies the vCenter name.
<code>--virtual-center-password=String</code>	When adding a vCenter client, specifies the vCenter root user password.
<code>--virtual-center-port=Integer</code>	When adding a vCenter client, specifies the vCenter port address.
<code>--virtual-center-username=String</code>	When adding a vCenter client, specifies the vCenter root username.

Event codes

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

The `--overtime-option` and `--overtime` options interact as follows:

- ◆ 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

This 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
```

This 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>
```

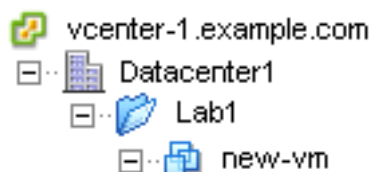
This command adds a new proxy called MyProxy:

```
mccli client add --name=/clients/MyProxy --type=proxy  
--virtual-center-name=vcenter-1.example.com
```

```
0,22210,Client added
```

Attribute	Value
restoreOnly	false
AllowClientOverrideSchedule	false
pagePort	N/A
plugins	<Plugins/>
registeredDate	N/A
nodeName	MyProxy
display-nodeName	MyProxy
DSOverride	false
checkinDate	N/A
windowsHardwareProfile	N/A
clientId	f8888acc3c2fb8db604bd6a1abbacc5bd57b9a9d
encryptionMethod	high
pageAddress	N/A
canPage	true
backedUpDate	N/A
action	add
AllowFileSelectionOnScsBackups	true
agentVersion	unknown
pageAddrLocked	false
windowsID	N/A
enabled	false
isClientOs	false
DATASETID	Default:SNAPID
AllowClientAddToDataset	false
retryCnt	2
overtimeOption	NEXT_SUCCESS
RPOOverride	false
modifiedDate	Thu Aug 25 04:45:21 UTC 2013
AllowScsBackups	true
nodeAddress	N/A
timeOut	10
registered	false
POLICYID	Default:POLICYID
overrideEncryption	false
OverrideStandardScsRetpol	false
fullName	/clients/MyProxy

This 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
-----
restoreOnly                                  false
AllowClientOverrideSchedule                  false
pagePort                                     N/A
plugins                                      <Plugins><Plugin Build="ALL" Description="Linux VMware Image"
Version="7.1.100" /></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
AllowFileSelectionOnScCbBackups              true
agentVersion                                 unknown
pageAddrLocked                              false
windowsID                                    N/A
enabled                                      true
isClientOs                                  false
DATASETID                                    VMWARE:SNAPID
AllowClientAddToDataset                      false
retryCnt                                     2
overtimeOption                              NEXT_SUCCESS
RPOOverride                                 false
modifiedDate                                Thu Aug 25 05:06:38 UTC 2013
AllowScCbBackups                            true
nodeAddress                                  N/A
timeOut                                      10
registered                                   false
POLICYID                                     Default:POLICYID
overrideEncryption                           false
OverrideStandardScCbRetpol                  false
fullName                                     /vcenter-1.example.com/ACMCommunity_UgqlbhZtNPdDKnHEpQLMOQ
```

client add-datastore

The **mccli client add-datastore** command adds one or more new datastores to be protected by the specified Avamar proxy.

Synopsis

```
mccli client add-datastore --datacenter=String --datastore-name=String
  [--domain=String(/)] --name=String
```

Command options

Table 27 Command options for **mccli client add-datastore**

Option	Description
<code>--datacenter=String</code>	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.
<code>--datastore-name=String</code>	Specifies the datastore name to be protected by the Avamar proxy. This argument is required. Multiple <code>--datastore-name</code> arguments can be supplied with a single <code>mccli client add-datastore</code> command.
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the Avamar proxy specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the Avamar proxy to which the datastore will be added. String must be a valid Avamar proxy name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the <code>--domain</code> argument is ignored.

Event codes

22236 Client does not exist.
 24003 Failed to update datastore mappings of a client.
 24004 Datastore mappings of a client successfully updated.

Examples

This command adds datastore Storage1 and assigns proxy 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** 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 [--cmd=String [--cmd=String] ...]
  [--contained-vm-name=String] [--dataset=String]
  [--dataset-domain=String(/)] [--domain=String(/)] --name=String
  [--wait [=MIN]]
```

Command options

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

Option	Description
<code>--cmd=String</code>	Specifies one or more optional plug-in commands. You can supply multiple <code>--cmd</code> arguments, but each argument can only specify one plug-in command. For example, this is valid <code>--cmd</code> syntax: <code>--cmd="verbose=5"</code> <code>--cmd="throttle=5"</code> However, this is not valid <code>--cmd</code> syntax: <code>--cmd="verbose=5 throttle=5"</code>
<code>--contained-vm-name=String</code>	Specifies a virtual machine client within a VMware container or vApp. This option is only valid if the client specified by the <code>--name</code> argument is a VMware container or vApp.
<code>--dataset=String</code>	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 to return a list of valid dataset names.
<code>--dataset-domain=String(/)</code>	Specifies the Avamar server domain that contains the dataset specified by the <code>--dataset</code> argument.
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the client specified by the <code>--name</code> argument.

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

Option	Description
<code>--name=String</code>	Specifies which client is backed up. String must be a valid client name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the <code>--domain</code> argument is ignored.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

- 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.

Examples

This command backs up VMware container TEST1 using the VMware Image Dataset:

```
mccli client backup-dataset --name=/10.31.183.55/FO2  
  --contained-vm-name=TEST1 --dataset=/VMware Image Dataset
```

```
0,22305,client backup scheduled.  
Attribute      Value  
-----  
client         /10.31.183.55/FO2  
dataset        /VMware Image Dataset  
contained-vm   TEST1  
activity-id    9134225754337909
```

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 [--cmd=String [--cmd=String] ...]
  [--domain=String(/)] [--group-domain=String(/)] --group-name=String
  --name=String [--wait [=MIN]]
```

Command options

Table 29 Command options for **mccli client backup-group-dataset**

Option	Description
<code>--cmd=String</code>	Specifies one or more optional plug-in commands. You can supply multiple <code>--cmd</code> arguments, but each argument can only specify one plug-in command. For example, this is valid <code>--cmd</code> syntax: <code>--cmd="verbose=5" --cmd="throttle=5"</code> However, this is not valid <code>--cmd</code> syntax: <code>--cmd="verbose=5 throttle=5"</code>
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the client specified by the <code>--name</code> argument.
<code>--group-domain=String(/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--group-name</code> argument.
<code>--group-name=String</code>	Specifies the group dataset to use for the backup. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, <code>/MyDomain/MyGroup</code>), then the <code>--group-domain</code> argument is ignored.
<code>--name=String</code>	Specifies which client is backed up. String must be a valid client name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--domain</code> argument is ignored.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

- 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** 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 [--cmd=String [--cmd=String] ...]
  [--contained-vm-name=String] [--domain=String(/)] --name=String
  [--plugin=Integer] --target=String [--wait[=MIN]]
```

Command options

Table 30 Command options for **mccli backup-target** (page 1 of 2)

Option	Description
<code>--cmd=String</code>	Specifies one or more optional plug-in commands. You can supply multiple <code>--cmd</code> arguments, but each argument can only specify one plug-in command. For example, this is valid <code>--cmd</code> syntax: <code>--cmd="verbose=5" --cmd="throttle=5"</code> However, this is not valid <code>--cmd</code> syntax: <code>--cmd="verbose=5 throttle=5"</code>
<code>--contained-vm-name=String</code>	Specifies a virtual machine client within a VMware container or vApp. This option is only valid if the client specified by the <code>--name</code> argument is a VMware container or vApp.
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the client specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies which client is backed up. String must be a valid client name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--domain</code> argument is ignored.
<code>--plugin=Integer</code>	Specifies the plug-in ID. Use plugin show to return a list of valid plug-in IDs.

Table 30 Command options for **mccli backup-target** (page 2 of 2)

Option	Description
<code>--target=String</code>	Specifies the folders or files to include in the backup. At least one <code>--target=String</code> argument is required, and you can supply more than one <code>--target=String</code> 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 <code>--target=String</code> arguments. If backing up non-virtual filesystems, this argument accepts regular expression (regex) pattern matching operators, also known as wildcards.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

- 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

This command backs up the Test1Windows222_1.vmdk virtual disk:

```
mccli client backup-target
  --name=/vcenter-1.example.com/Test1Windows222 --plugin=3016
  --target=[Storage2] Test1Windows222/Test1Windows222_1.vmdk
```

```
0,22305,client backup scheduled.
```

```
Attribute      Value
```

```
-----
client        /vcenter-1.example.com/Test1Windows222
target        [Storage2] Test1Windows222/Test1Windows222_1.vmdk
activity-id   9131407958056109
```

This command backs up a the TEST1 VMware container:

```
mccli client backup-target --name=/10.31.183.55/FO2
  --contained-vm-name=TEST1 --plugin=3016 --target=ALL
```

```
0,22305,client backup scheduled.
```

```
Attribute      Value
```

```
-----
client        /10.31.183.55/FO2
target        ALL
contained-vm  TEST1
activity-id   9134225778865209
```

client delete

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

NOTICE

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

Synopsis

```
mccli client delete [--domain=String()] --name=String
```

Command options

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

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

Event codes

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 [--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-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]
  [--vcontainer-inclusion=String] [--virtual-center-name=String]
  [--virtual-center-password=String]
  [--virtual-center-port=Integer(443)]
  [--virtual-center-username=String]
```

Command options

Table 32 Command options for **mccli client edit** (page 1 of 4)

Option	Description
<code>--activated=Boolean</code>	If true, client is assumed to have been previously activated.
<code>--allow-cis-fileselection=Boolean</code>	If true, users can make file selections when they initiate a backup of the client.
<code>--allow-cis=Boolean</code>	If true, users can initiate a backup of the client.
<code>--changed-block-tracking=Boolean(false)</code>	If true, when editing a virtual machine, VMware container, or vApp, changed block tracking is enabled.
<code>--contact=String</code>	Specifies contact information for the client.
<code>--dataset-domain=String(/)</code>	Specifies the Avamar server domain that contains the dataset specified by the <code>--dataset</code> argument.
<code>--dataset=String</code>	Specifies the dataset that the client should use when performing on-demand backups and restores, or when the group dataset is overridden.
<code>--domain=String</code>	Specifies the domain that contains the client specified by the <code>--name</code> argument.

Table 32 Command options for `mccli client edit` (page 2 of 4)

Option	Description
<code>--email=String</code>	Specifies an email address for the primary contact (responsible party).
<code>--enabled=Boolean</code>	If true, client is eligible to immediately participate in on-demand and group backup and restore activities. If false, then the client is disabled.
<code>--encryption=String</code>	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: <ul style="list-style-type: none"> • 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. <p>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 <i>EMC Avamar Product Security Guide</i> provides additional information.</p>
<code>--location=String</code>	Specifies location information for the client.
<code>--name=String</code>	Specifies the client to edit. String must be a valid client name. This argument is required. Use <code>mccli client show</code> to return a list of valid client names. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--domain</code> argument is ignored.
<code>--new-name=String</code>	Specifies a new client name.
<code>--override-cis-retention=Boolean</code>	If true, the retention policy is used for all user-initiated backups of the client.
<code>--override-encryption=Boolean</code>	If true, use the client encryption method instead of the group encryption method for scheduled backups.
<code>--override-retention=Boolean</code>	If true, use the client retention policy instead of the group retention policy for scheduled backups.

Table 32 Command options for `mccli client edit` (page 3 of 4)

Option	Description
<code>--overtime=option=String</code>	Specifies scheduled group backup overtime behavior. String must be one of the following: <ul style="list-style-type: none"> • 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.
<code>--overtime=Boolean</code>	If true, the client can exceed its backup window during backup activities.
<code>--page-detection=String (auto)</code>	Specifies client page detection mode. String must be one of the following: <ul style="list-style-type: none"> • Automatic—use automatic paging detection (that is, ignore <code>--pageaddr</code> and <code>--pageport</code> settings) • Manual—use <code>--pageaddr</code> and <code>--pageport</code> settings
<code>--pageable=Boolean</code>	If true, the client can be paged for the purpose of initiating activation or picking up new backup or restore work.
<code>--pageaddr=String</code>	Specifies client paging IP address.
<code>--pageport=String</code>	Specifies client paging port. The default page port is 28002.
<code>--phone=String</code>	Specifies voice telephone information for the primary contact (responsible party) for the client.
<code>--retention-domain=String (/)</code>	Specifies the Avamar server domain that contains the retention policy specified by the <code>--retention</code> argument.
<code>--retention=String</code>	Specifies the retention policy that the client should use when performing on-demand backups and restores, or when the group retention policy is overridden.
<code>--timeout=String</code>	Specifies the client browse timeout in seconds.

Table 32 Command options for `mccli client edit` (page 4 of 4)

Option	Description
<code>--vcontainer-inclusion=String</code>	When editing a VMware container or vApp, specifies the inclusion settings. String must be one of the following: <ul style="list-style-type: none"> • <code>dynamic</code>—includes all contents of the vCenter container, but also continuously monitors the container entity in vCenter, so that if changes occur (for example, virtual machines or folders are added or deleted), those changes will be automatically reflected in Avamar. • <code>static</code>—only includes what is in the vCenter container at the time it is added to Avamar. If subsequent changes occur in vCenter, they will not be reflected in Avamar.
<code>--virtual-center-name=String</code>	When editing an Avamar proxy, specifies the vCenter name.
<code>--virtual-center-password=String</code>	When editing a vCenter client, specifies the vCenter root user password.
<code>--virtual-center-port=Integer (443)</code>	When editing a vCenter client, specifies the vCenter port address. Default port is 443.
<code>--virtual-center-username=String</code>	When editing a vCenter client, specifies the vCenter root username.

Event codes

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

The `--overtime-option` and `--overtime` options interact as follows:

- ◆ 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

This command 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.

This command 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.

This command 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.

This command sets dynamic inclusion for VMware container F02:

```
mccli client edit --name=/10.31.183.55/F02
--vcontainer-inclusion=dynamic
```

0,22211,Client was updated.

This command turns off changed block tracking for VMware container F02:

```
mccli client edit --name=/10.31.183.55/F02
--changed-block-tracking=false
```

0,22211,Client was updated.

client import-clients-from-file

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

Synopsis

```
mccli client import-clients-from-file --file=String
  [--verbose=Boolean(false)]
```

Command options

Table 33 Command options for **mccli client import-clients-from-file**

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

Event codes

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

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 [--domain=String(/)] --name=String
```

Command options

Table 34 Command options for **mccli client invite**

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

Event codes

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 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 [--domain=String(/)] --name=String
                  --new-domain=String
```

Command options

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

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

Event codes

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 one or more datastores from the specified Avamar proxy.

Synopsis

```
mccli client remove-datastore --datacenter=String
    --datastore-name=String [--domain=String(/)] --name=String
```

Command options

Table 36 Command options for **mccli client remove-datastore**

Option	Description
<code>--datacenter=String</code>	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.
<code>--datastore-name=String</code>	Specifies the datastore name to be removed from the Avamar proxy. This argument is required. Multiple <code>--datastore-name</code> arguments can be supplied with a single mccli client remove-datastore command.
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the Avamar proxy specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the Avamar proxy to which the datastore will be removed. String must be a valid Avamar proxy name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the <code>--domain</code> argument is ignored.

Event codes

- 22236 Client does not exist.
- 24003 Failed to update datastore mappings of a client.
- 24004 Datastore mappings of a client successfully updated.

Examples

This command 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 [--domain=String()] [--expiration=String]
  --name=String
```

Command options

Table 37 Command options for **mccli client retire**

Option	Description
<code>--domain=String()</code>	Specifies the domain that contains the client specified by the <code>--name</code> argument.
<code>--expiration=String</code>	Specifies the expiration of backups for the retired client. String must be one of the following: <ul style="list-style-type: none"> 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
<code>--name=String</code>	Specifies the client to retire. String must be a valid client name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the <code>--domain</code> argument is ignored.

Event codes

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 [--contained-vm-name=String]
  [--contained-vms=Boolean(false)] [--domain=String(/)]
  [--name=String] [--recursive=Boolean] [--replicated=Boolean(false)]
  [--retired=Boolean(false)] [--verbose=Boolean(false)]
```

Command options

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

Option	Description
<code>--contained-vm-name=String</code>	Specifies a virtual machine client within a VMware container or vApp. This option is only valid if the client specified by the <code>--name</code> argument is a VMware container or vApp.
<code>--contained-vms=Boolean(false)</code>	If true, shows details for VMware containers or vApps. This option is only valid if the client specified by the <code>--name</code> argument is a VMware container or vApp.
<code>--domain=String(/)</code>	Specifies the domain that contains the client specified by the <code>--name</code> argument. If you supply <code>--retired</code> , then this argument is ignored.
<code>--name=String</code>	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, <code>/clients/MyClient</code>), then the <code>--domain</code> argument is ignored.
<code>--recursive=Boolean(false)</code>	If true, then the command shows all clients in the domain and any subdomains specified by the <code>--domain=String</code> argument. If false or not supplied, then the command only shows clients in the specified domain. Subdomains are not examined.
<code>--replicated=Boolean(false)</code>	If true, then clients in the REPLICATE domain are shown. If false or not supplied, then clients in the REPLICATE domain are not shown.
<code>--retired=Boolean(false)</code>	If true, then retired clients are listed. If supplied, then the <code>--domain=String</code> argument is ignored.
<code>--verbose=Boolean(false)</code>	If you do not supply <code>--name</code> 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 <code>--name</code> option. When you supply the <code>--name</code> option, the details for the client are always returned.

Event codes

22236 Client does not exist.
22542 Domain does not exist.

Examples

This 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
```

This 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                                 7.1.100-nnn
CID                                            6e6241ae236c2ab58205f413f24d3d6e0
CID Assigned                                  2013-10-13 00:00:00 PDT
Activated                                     Yes
Activated Date                               2013-10-13 12:28:13 PDT
Disabled                                     No
Agent Last Started                           2013-1-10 06:00:00 PST
Last Check-in                                2013-1-10 11:00:59 PST
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                          2013-1-10 12:28:14 PDT
Last Version Registered                       2013-1-10 12:28:14 PDT
Last Successful Backup                        2013-1-10 22:00:41 PST
```

This command shows detailed information for virtual machines in the F02 VMware container:

```
mccli client show --name=/10.31.183.55/F02 --contained-vms=true --xml
```

```
<CLIOutput>
  <Results>
    <ReturnCode>0</ReturnCode>
    <EventCode>23000</EventCode>
    <EventSummary>CLI command completed successfully.</EventSummary>
  </Results>
  <Data>
    <Row>
      <Client>TEST1</Client>
      <Domain>/10.31.183.55/HleDynamicClients</Domain>
      <ClientType>Virtual Machine</ClientType>
    </Row>
    <Row>
      <Client>VA1</Client>
      <Domain>/10.31.183.55/HleDynamicClients</Domain>
      <ClientType>Virtual Container</ClientType>
    </Row>
  </Data>
</CLIOutput>
```

This command shows detailed information for virtual machine TEST1, which resides in VMware container F02:

```
mccli client show --name=/10.31.183.55/F02 --contained-vm-name=TEST1
```

```
0,23000,CLI command completed successfully.
Attribute                               Value
-----
Client Name                             TEST1
Fully Qualified Name                    /10.31.183.55/HleDynamicClients/TEST1_UCKZ3FGLz70WRRu3nq2ZuA
Client Type                             Virtual Machine
Paging                                  No
CID                                      6214c7aac9ce678b0b19e4b05475d8fab58e2718
CID Assigned                            2013-07-14 05:43:38 UTC
Disabled                                No
Agent Last Started                     2013-07-14 05:43:38 UTC
Override Group Retention                No
Retention Policy                       Default Retention
Overtime Option                        Overtime until successful backup
Restore Only                            No
Encryption Method                      High
Override Encryption                    No
Allow CIS                               Yes
Allow CIS file selection                Yes
Override CIS retention                 No
Browse Timeout                         10 seconds
Contact Name                           N/A
Contact Phone                          N/A
Contact Email                          N/A
Contact Location                       N/A
Contact Notes                          N/A
Member of Group                        /10.31.183.55/Default Virtual Machine Group
Plugin                                  Windows VMware Image(3016)
Initial Install Date                   2013-07-14 05:43:38 UTC
Last Version Registered                2013-07-14 05:43:38 UTC
Last Successful Backup                 N/A
```

client show-datastore

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

Synopsis

```
mccli client show-datastore [--domain=String(/)] --name=String
```

Command options

Table 39 Command options for **mccli client show-datastore**

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

Event codes

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

This command shows all datastores protected by client `backupproxy225`:

```
mccli client show-datastore --name=/clients/backupproxy225
```

Name	Type	Accessible	Hosts	Datacenter
-----	----	-----	-----	-----
Storage1	VMFS	Yes	10.31.183.7	VAAJU-DEV-WIN
Storage2	VMFS	Yes	10.31.183.7	VAAJU-DEV-WIN

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 [--domain=String(/)] --name=String
  [--plugin=Integer] [--verbose=Boolean(false)]
```

Command options

Table 40 Command options for **mccli client show-plugins**

Option	Description
<code>--domain=String(/)</code>	Specifies the domain that contains the client specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the client for which plug-ins are shown. String must be a valid client name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--domain</code> argument is ignored.
<code>--plugin=Integer</code>	Specifies the plug-in ID. Use mccli plugin show to return a list of valid plug-in IDs.
<code>--verbose=Boolean(false)</code>	If true, detailed plug-in information is returned.

Event codes

- 22236 Client does not exist.
- 23009 Invalid plugin specified on the CLI.

client validate-clients-from-file

The **mccli client validate-clients-from-file** command validates a clients definition input file, which is used by the **mccli client import-clients-from-file** command. The validation process ensures that the XML data is properly formatted and that the file works with the **mccli client import-clients-from-file** command.

Synopsis

```
mccli client validate-clients-from-file --file=String
  [--verbose=Boolean(false)]
```

Command options

Table 41 Command options for **mccli client validate-clients-from-file**

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

Event codes

The following event code is associated with the **mccli client validate-clients-from-file** 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. Commands are:

- ◆ “dataset add” on page 88
- ◆ “dataset add-exclude” on page 89
- ◆ “dataset add-include” on page 90
- ◆ “dataset add-option” on page 91
- ◆ “dataset add-target” on page 92
- ◆ “dataset copy” on page 93
- ◆ “dataset delete” on page 94
- ◆ “dataset delete-exclude” on page 95
- ◆ “dataset delete-include” on page 96
- ◆ “dataset delete-option” on page 97
- ◆ “dataset delete-target” on page 98
- ◆ “dataset edit-option” on page 99
- ◆ “dataset replace” on page 100
- ◆ “dataset show” on page 102

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. Use other dataset commands to operate on the dataset after it has been created.

Synopsis

```
mccli dataset add [--alldata=Boolean(true)] [--domain=String(/)]
  --name=String
```

Command options

Table 42 Command options for **mccli dataset add**

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

Event codes

- 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 [--domain=String()] [--exclude=String
[--exclude=String] ... | --exclude-file=FILE} --name=String
--plugin=Integer
```

Command options

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

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

Event codes

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 [--domain=String()] [--include=String
[--include=String] ... | --include-file=FILE} --name=String
--plugin=Integer
```

Command options

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

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

Event codes

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 [--domain=String(/)] --name=String
  {--option=String [--option=String] ...} --plugin=Integer
  {--value=String [--value=String] ...}
```

Command options

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

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

Event codes

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 [--domain=String(/)] --name=String
  --plugin=Integer {--target=String [--target-file=FILE] ... |
  --target-file=FILE}
```

Command options

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

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

Event codes

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 [--domain=String()] --name=String
  [--new-domain=String()] --new-name=String
```

Command options

Table 47 Command options for **mccli dataset copy**

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

Event codes

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 [--domain=String(/)] --name=String
```

Command options

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

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

Event codes

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 {--all | --exclude=String
  [--exclude=String] ... | --exclude-file=FILE} [--domain=String()]
  --name=String --plugin=Integer
```

Command options

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

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

Event codes

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 {--all | {--include=String
  [--include=String] ... | --include-file=FILE} [--domain=String(/)]
  --name=String --plugin=Integer
```

Command options

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

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

Event codes

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 [--domain=String(/)] --name=String
  [--option=String [--option=String] ...] --plugin=Integer
```

Command options

Table 51 Command options for **mccli dataset delete-option**

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

Event codes

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 [--domain=String(/)] --name=String
  --plugin=Integer {--target=String [--target-file=FILE] ...}
```

Command options

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

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

Event codes

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 edit-option

The **mccli dataset edit-option** command modifies existing dataset options. Use **mccli dataset add option** to add new dataset options.

Synopsis

```
mccli dataset edit-option --name=String --option=String
    --plugin=Integer --value=String
```

Command options

Table 53 Command options for **mccli dataset edit-option**

Option	Description
<code>--name=String</code>	Specifies which dataset to modify. String must be a valid dataset name. This argument is required. Use mccli dataset show to return a list of valid dataset names. If you supply a fully qualified dataset name (for example, /MyDomain/MyDataset), then the <code>--domain</code> argument is ignored.
<code>--option=String</code>	Specifies which option to modify. This argument is required. Multiple <code>--option=String</code> arguments are allowed and must be separated by a space.
<code>--plugin=Integer</code>	Specifies the plug-in ID. This argument is required. Use mccli plugin show to return a list of valid plug-in IDs.
<code>--value=String</code>	Specifies a new value for <code>--option</code> . This argument is required. Multiple <code>--value=String</code> arguments are allowed and must be separated by a space.

Event codes

22220 Dataset modified.

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 [--domain=String(/)] [--exclude=String
  [--exclude=String] ... | --exclude-file=FILE}] [--include=String
  [--include=String] ... | --include-file=FILE}] --name=String
  --plugin=Integer
```

Command options

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

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

Event codes

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 [--domain=String(/)] [--name=String]
  [--recursive=Boolean(false)] [--verbose=Boolean(false)]
```

Command options

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

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the dataset specified by the <code>--name</code> argument.
<code>--name=String</code>	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, <code>/MyDomain/MyDataset</code>), then the <code>--domain</code> argument is ignored.
<code>--recursive=Boolean(false)</code>	If true, then the command shows all datasets in the domain and any subdomains specified by the <code>--domain=String</code> argument. If false or not supplied, then the command only shows datasets in the domain. Subdomains are not examined.
<code>--verbose=Boolean(false)</code>	If true, then detailed dataset information is returned. If false or not supplied, then only dataset names are returned. If you supply <code>--name</code> , then detailed information for the dataset is returned.

Event codes

22288 Dataset does not exist.

22542 Domain does not exist.

Examples

This 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     /
```

This 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

This 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. Commands are:

- ◆ “dd add” on page 104
- ◆ “dd delete” on page 106
- ◆ “dd edit” on page 107
- ◆ “dd show-prop” on page 109
- ◆ “dd show-util” on page 110

dd add

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

Synopsis

```
mccli dd add [--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(163)]
--user-name=String
```

Command options

Table 56 Command options for **mccli dd add** (page 1 of 2)

Option	Description
<code>--default-storage=Boolean(false)</code>	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.
<code>--force</code>	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 <code>--force</code> with <code>--default-storage</code> to override the current default storage setting (that is, force the change to occur).
<code>--max-streams=Integer</code>	Specifies the maximum number of Data Domain system streams that Avamar uses for backups, restores, and replication.
<code>--name=String</code>	Specifies the fully qualified domain name of the Data Domain system.
<code>--password-confirm=String</code>	Confirmation of the password.
<code>--password=String</code>	Specifies the password for the Data Domain OST account that Avamar should use to access the Data Domain system for backups, restores, and replication.
<code>--rw-community=String</code>	Specifies the SNMP community string for Avamar to have read/write access to the Data Domain system.
<code>--snmp-port=Integer(161)</code>	Specifies the data port number on the Data Domain system from which to receive and on which to set SNMP objects.

Table 56 Command options for `mccli dd add` (page 2 of 2)

Option	Description
<code>--trap-port=Integer (163)</code>	Specifies the data port number on the Avamar server to which SNMP trap messages are sent.
<code>--user-name=String</code>	Specifies the username of the Data Domain OST account that Avamar should use to access the Data Domain system for backups, restores, and replication.

Event codes

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 --name=String [--force=Boolean(false)]
```

Command options

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

Option	Description
<code>--name=String</code>	Specifies the fully qualified domain name of the Data Domain system to delete.
<code>--force=Boolean(false)</code>	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.

Event codes

- 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 [--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(163)] [--user-name=String]
```

Command options

Table 58 Command options for **mccli dd edit**

Option	Description
<code>--default-storage=Boolean(false)</code>	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.
<code>--force</code>	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 <code>--force</code> with <code>--default-storage</code> to override the current default storage setting (that is, force the change to occur).
<code>--max-streams=Integer</code>	Specifies the maximum number of Data Domain system streams that Avamar uses for backups, restores, and replication.
<code>--name=String</code>	Specifies the fully qualified domain name of the Data Domain system. This parameter is read-only.
<code>--new-name=String</code>	If renaming a Data Domain system, specifies the new fully qualified domain name of the Data Domain system.
<code>--password=String</code>	Specifies the password for the Data Domain OST account that Avamar should use to access the Data Domain system for backups, restores, and replication.
<code>--password-confirm=String</code>	Confirmation of the password.
<code>--rw-community=String</code>	Specifies the SNMP community string for Avamar to have read/write access to the Data Domain system.
<code>--snmp-port=Integer(161)</code>	Specifies the data port number on the Data Domain system from which to receive and on which to set SNMP objects.
<code>--trap-port=Integer(163)</code>	Specifies the data port number on the Avamar server to which SNMP trap messages are sent.
<code>--user-name=String</code>	Specifies the username of the Data Domain OST account that Avamar should use to access the Data Domain system for backups, restores, and replication.

Event codes

30935	Updated Data Domain system in persistent store.
30938	Failed to update Data Domain system in persistent 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 persistent 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 [--name=String]
```

Command options

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

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

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 [--name=String]
```

Command options

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

Option	Description
<code>--name=String</code>	Specifies the fully qualified Data Domain system name.

Event codes

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. Commands are:

- ◆ [“domain add” on page 111](#)
- ◆ [“domain delete” on page 112](#)
- ◆ [“domain edit” on page 113](#)
- ◆ [“domain show” on page 114](#)

domain add

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

Synopsis

```
mccli domain add [--contact=String] [--domain=String(/)]
  [--email=String] [--location=String] --name=String [--phone=String]
```

Command options

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

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

Event codes

- 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 [--domain=String] --name=String
```

Command options

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

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

Event codes

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 [--contact=String] [--domain=String(/)]
  [--email=String] [--location=String] --name=String [--phone=String]
```

Command options

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

Option	Description
<code>--contact=String</code>	Specifies contact information for the domain.
<code>--domain=String(/)</code>	Specifies the parent domain.
<code>--email=String</code>	Specifies the email address for the primary contact (responsible party).
<code>--location=String</code>	Specifies location information for the domain.
<code>--name=String</code>	Specifies the domain to edit. String must be a valid domain name. This argument is required. Use mccli domain show to return a list of valid domain names.
<code>--phone=String</code>	Specifies telephone information for the primary contact (responsible party).

Event codes

22519 Domain modified.
22542 Domain does not exist.

domain show

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

Synopsis

```
mccli domain show [--domain=String(/)] [--name=String]
  [--recursive=Boolean(false)]
```

Command options

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

Option	Description
<code>--domain=String(/)</code>	Specifies the parent domain under which domains and subdomains are listed.
<code>--name=String</code>	Specifies the name of the domain for which to show information. String must be a valid domain name.
<code>--recursive=Boolean(false)</code>	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.

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. Commands are:

- ◆ [“dump clientcache” on page 115](#)
- ◆ [“dump domaincache” on page 115](#)
- ◆ [“dump jobcache” on page 115](#)

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
```

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
```

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
```

event

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

- ◆ “event ack” on page 116
- ◆ “event clear-data-integrity-alerts” on page 119
- ◆ “event get-info” on page 119
- ◆ “event show” on page 120

event ack

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

Synopsis

```
mccli event ack [--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]
```

Command options

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

Option	Description
<code>--after=String</code>	Acknowledges events after a date. String must be in the format of: <code>YYYY-MM-DD HH:mm:ss</code> 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.
<code>--all=Boolean(false)</code>	Acknowledges all events.
<code>--before=String</code>	Acknowledges events before a date. String must be in the format of: <code>YYYY-MM-DD HH:mm:ss</code> 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.
<code>--category=String</code>	Specifies the category of the events to acknowledge. String must be one of the following: <ul style="list-style-type: none"> • APPLICATION • SECURITY • SYSTEM • USER
<code>--domain=String</code>	Specifies the domain of the events to acknowledge.
<code>--exclude=String</code>	Comma-separated list of event codes to exclude from acknowledgement. The comma-separated list cannot contain spaces. The <code>--include</code> and <code>--exclude</code> options are mutually exclusive.
<code>--id=Integer</code>	Unique identifier of the event to acknowledge.
<code>--include=String</code>	Comma-separated list of event codes to include for acknowledgement. The comma-separated list cannot contain spaces. The <code>--include</code> and <code>--exclude</code> options are mutually exclusive.

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

Option	Description
<code>--range=String</code>	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.
<code>--severity=String</code>	Specifies the severity of the events to acknowledge. String must be one of the following: <ul style="list-style-type: none"> • NODE • NODE_FATAL • OK • PROCESS • PROCESS_FATAL • SYSTEM_FATAL • USER • USER_FATAL
<code>--source=String</code>	Specifies a source storage location for the events to acknowledge, either: <ul style="list-style-type: none"> • 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.
<code>--type=String</code>	Specifies the type of event to acknowledge. String must be one of the following: <ul style="list-style-type: none"> • AUDIT • DEBUG • ERROR • INFORMATION • INTERNAL • WARNING

Event codes

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

23022 Event id does not exist.

Notes

- ◆ This command requires either the `--all` option or some other filtering criteria specified using `--exclude` and `--include` options.
- ◆ The `--include` and `--exclude` options are mutually exclusive.
- ◆ Output is a count of the events acknowledged.

Examples

This 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 clear-data-integrity-alerts

The **mccli event clear-data-integrity-alerts** command clears all data integrity alerts for this server.

This command requires a reset code, which must be obtained from EMC Technical Support.

Synopsis

```
mccli event clear-data-integrity-alerts --reset-code=String
```

Command options

Table 66 Command options for **mccli event clear-data-integrity-alerts**

Option	Description
<code>--reset-code=String</code>	Reset code provided by EMC Technical Support. This option is required.

Event codes

The following event code is associated with the **mccli event clear-data-integrity-alerts** command.

- 23100 A data integrity alerts reset code was accepted.
- 23101 A data integrity alerts reset code was rejected.

event get-info

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

Synopsis

```
mccli event get-info --code=Integer [--description=Boolean(false)]
  [--remedy=Boolean(false)] [--summary=Boolean(true)]
```

Command options

Table 67 Command options for **mccli event get-info**

Option	Description
<code>--code=Integer</code>	Specifies the event code number. This argument is required.
<code>--description=Boolean(false)</code>	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.
<code>--remedy=Boolean(false)</code>	If true, then event remedy information is returned. If false or not supplied, then the event remedy is not returned.
<code>--summary=Boolean(true)</code>	If true or not supplied, then event summary information is returned. If false, then the event summary is not returned.

event show

The **mccli event show** command returns event occurrence details.

Synopsis

```
mccli event show [--after=String] [--before=String]
  [--category=String] [--domain=String] [--exclude=String]
  [--id=Integer] [--include=String] [--severity=String]
  [--source=String] [--type=String] [--unack=Boolean(false)]
```

Command options

Table 68 Command options for **mccli event show** (page 1 of 2)

Option	Description
<code>--after=String</code>	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.
<code>--before=String</code>	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.
<code>--category=String</code>	Specifies the category of the event for which to show information. String must be one of the following: <ul style="list-style-type: none"> • APPLICATION • SECURITY • SYSTEM • USER
<code>--domain=String</code>	Specifies the domain of the event for which to show information.
<code>--exclude=String</code>	Comma-separated list of event codes to exclude from the events that are shown. The comma-separated list cannot contain spaces. The <code>--include</code> and <code>--exclude</code> options are mutually exclusive.
<code>--id=Integer</code>	Unique identifier of the event for which to show information.
<code>--include=String</code>	Comma-separated list of event codes to include in the events that are shown. The comma-separated list cannot contain spaces. The <code>--include</code> and <code>--exclude</code> options are mutually exclusive.
<code>--severity=String</code>	Specifies the severity of the event for which to show information. String must be one of the following: <ul style="list-style-type: none"> • NODE • NODE_FATAL • OK • PROCESS • PROCESS_FATAL • SYSTEM_FATAL • USER • USER_FATAL

Table 68 Command options for `mccli event show` (page 2 of 2)

Option	Description
<code>--source=String</code>	<p>Specifies a source storage location of the event for which to show information, either:</p> <ul style="list-style-type: none"> • <code>avamar</code> • <code>dd</code> <p>If you specify <code>avamar</code>, then the command shows events for the Avamar server. If you specify <code>dd</code>, then the command shows events for all configured Data Domain systems.</p>
<code>--type=String</code>	<p>Specifies the event types for which to show information. String must be one of the following:</p> <ul style="list-style-type: none"> • <code>AUDIT</code> • <code>DEBUG</code> • <code>ERROR</code> • <code>INFORMATION</code> • <code>INTERNAL</code> • <code>WARNING</code> <p>You can supply more than one <code>--type</code> option on a single command line.</p>
<code>--unack=Boolean(false)</code>	Shows only information for unacknowledged events.

Event codes

23022 Event id does not exist.

Notes

- ◆ If multiple events are displayed, sorting is by date in descending order.
- ◆ The `--include` and `--exclude` options are mutually exclusive.

Examples

This 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           2014-04-19 14:19:45 PDT
Type           INFORMATION
Code           578
Category       SYSTEM
Severity       OK
Domain         /
Summary        hfscheckresults::merge, in checkpoint cp.20130419140741,
                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 unquestarttime 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.20130419140741, 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 unquestarttime
                fields differ" version="" /><entry key="date" type="text"
                value="2014/04/19" version="" /><entry key="thread"
                type="text" value="srvm-14070642#srv:265"
                version="" /></data>
```

group

The **mccli group** resource is used to create and manage groups and group policy on the Avamar server. Commands are:

- ◆ “group add” on page 124
- ◆ “group add-client” on page 126
- ◆ “group add-proxy” on page 127
- ◆ “group backup” on page 128
- ◆ “group copy” on page 129
- ◆ “group delete” on page 130
- ◆ “group edit” on page 131
- ◆ “group export” on page 133
- ◆ “group move-client” on page 134
- ◆ “group remove-client” on page 135
- ◆ “group remove-proxy” on page 136
- ◆ “group replicate” on page 137
- ◆ “group show” on page 138
- ◆ “group show-members” on page 140

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**
- ◆ **schedule show**
- ◆ **retention show**
- ◆ **plugin show**

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 --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(/)]
```

Command options

Table 69 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: <ul style="list-style-type: none"> • High—Strongest available encryption setting for that specific client platform. • Medium—Medium strength encryption. • None—No encryption. <p>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 <i>EMC Avamar Product Security Guide</i> provides additional information.</p>
--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.

Table 69 Command options for `mccli group add` (page 2 of 2)

Option	Description
<code>--schedule=String(Default Schedule)</code>	Specifies the schedule to assign to the group. This argument is required.
<code>--schedule-domain=String(/)</code>	Specifies the Avamar server domain that contains the schedule specified by the <code>--schedule</code> argument.

Event codes

- 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 [--client-domain=String()]
  --client-name=String [--dataset-domain=String()]
  [--dataset=String] [--domain=String()] --name=String
```

Command options

Table 70 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 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.
--dataset-domain=String()	Specifies the Avamar server domain that contains the dataset specified by the --dataset argument.
--dataset=String	Specifies an alternative dataset this client will use when this group dataset is overridden.
--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 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.

Event codes

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 proxy to the specified group.

Synopsis

```
mccli group add-proxy GLOBAL-OPTIONS] [--domain=String()]
  --name=String [--proxy-domain=String()] --proxy-name=String
```

Command options

Table 71 Command options for **mccli group add-proxy**

Option	Description
<code>--domain=String()</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group to which to add the client. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the <code>--domain</code> argument is ignored.
<code>--proxy-domain=String()</code>	Specifies the Avamar server domain that contains the Avamar proxy specified by the <code>--proxy-name</code> argument.
<code>--proxy-name=String</code>	Specifies which Avamar proxy to add. String must be a valid Avamar proxy name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, /clients/MyClient), then the <code>--proxy-domain</code> argument is ignored.

Event codes

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

This command adds proxy backupproxy225 to the Test1 group:

```
mccli group add-proxy --domain=/vcenter-1.example.com --name=Test1
  --proxy-domain=/clients --proxy-name=backupproxy225
```

0,24002,Proxy client mappings of a group successfully updated.

```
Attribute      Value
-----
group          /vcenter-1.example.com/Test1
```

group backup

The **mccli group backup** command initiates an on-demand group backup.

Synopsis

```
mccli group backup [--domain=String(/)] --name=String [--wait[=MIN]]
```

Command options

Table 72 Command options for **mccli group backup**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group to back up. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the <code>--domain</code> argument is ignored.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

- 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.
- 23028 Invalid group type.

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 [--domain=String(/)] --name=String --new-name=String
```

Command options

Table 73 Command options for **mccli group copy**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group to copy. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, <code>/MyDomain/MyGroup</code>), then the <code>--domain</code> argument is ignored.
<code>--new-name=String</code>	Specifies the name for the new group. This argument is required.

Event codes

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 before you delete the group.

Synopsis

```
mccli group delete [--domain=String (/)] --name=String
```

Command options

Table 74 Command options for **mccli group delete**

Option	Description
<code>--domain=String (/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group to delete. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, <code>/MyDomain/MyGroup</code>), then the <code>--domain</code> argument is ignored.

Event codes

- 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 --dataset=String [--dataset-domain=String()]
  [--domain=String] --enabled=Boolean [--encryption=String]
  --name=String --retention=String [--retention-domain=String()]
  --schedule=String [--schedule-domain=String()]
```

Command options

Table 75 Command options for **mccli group edit** (page 1 of 2)

Option	Description
<code>--dataset=String</code>	Specifies the dataset assigned to the group. This argument is required.
<code>--dataset-domain=String()</code>	Specifies the Avamar server domain that contains the dataset specified by the <code>--dataset</code> argument.
<code>--domain=String</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--enabled=Boolean</code>	If true, then the group is eligible to immediately participate in scheduled backup activities. This argument is required.
<code>--encryption=String</code>	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: <ul style="list-style-type: none"> • High—Strongest available encryption setting for that specific client platform. • Medium—Medium strength encryption. • None—No encryption. <p>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 <i>EMC Avamar Product Security Guide</i> provides additional information.</p>
<code>--name=String</code>	Specifies the group to edit. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, <code>/MyDomain/MyGroup</code>), then the <code>--domain</code> argument is ignored.
<code>--retention=String</code>	Specifies the retention policy for the group. This argument is required.
<code>--retention-domain=String()</code>	Specifies the Avamar server domain that contains the retention policy specified by the <code>--retention</code> argument.

Table 75 Command options for **mccli group edit** (page 2 of 2)

Option	Description
<code>--schedule=String</code>	Specifies the schedule for the group. This argument is required.
<code>--schedule-domain=String(/)</code>	Specifies the Avamar server domain that contains the schedule specified by the <code>--schedule</code> argument.

Event codes

- 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 [--domain=String(/)] --file=String --name=String
  [--force=Boolean(false)]
```

Command options

Table 76 Command options for **mccli group export**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--file=String</code>	Specifies the name of the export XML file to create. This option is required.
<code>--force=Boolean(false)</code>	If true, then the option forces the overwrite of an existing XML file.
<code>--name=String</code>	Specifies the group to export. This option is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, <code>/MyDomain/In-group</code>), then the <code>--domain</code> argument is ignored.

Event codes

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 [--client-domain=String()]
  --client-name=String [--domain=String()] --name=String
  [--old-group-domain=String()] [--old-group-name=String(Default
  Group)]
```

Command options

Table 77 Command options for **mccli group move-client**

Option	Description
<code>--client-domain=String()</code>	Specifies the domain to which to move the client.
<code>--client-name=String</code>	Specifies the client to move. This argument is required.
<code>--domain=String()</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group to which to move the client. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, <code>/MyDomain/MyGroup</code>), then the <code>--domain</code> argument is ignored.
<code>--old-group-domain=String()</code>	Specifies the domain from which to move the client.
<code>--old-group-name=String(Default Group)</code>	Specifies the group from which to move the client.

Event codes

- 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.

Synopsis

```
mccli group remove-client [--client-domain=String()]
  --client-name=String [--domain=String()] --name=String
```

Command options

Table 78 Command options for **mccli group remove-client**

Option	Description
<code>--client-domain=String()</code>	Specifies the domain from which to remove the client.
<code>--client-name=String</code>	Specifies the client to remove. This argument is required.
<code>--domain=String()</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group from which to remove the client. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, <code>/MyDomain/MyGroup</code>), then the <code>--domain</code> argument is ignored.

Event codes

- 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 proxy from the specified group.

Synopsis

```
mccli group remove-proxy [--domain=String(/)] --name=String
  [--proxy-domain=String(/)] --proxy-name=String
```

Command options

Table 79 Command options for **mccli group remove-proxy**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group from which to remove the client. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, <code>/MyDomain/MyGroup</code>), then the <code>--domain</code> argument is ignored.
<code>--proxy-domain=String(/)</code>	Specifies the Avamar server domain that contains the Avamar proxy specified by the <code>--proxy-name</code> argument.
<code>--proxy-name=String</code>	Specifies which Avamar proxy to remove. String must be a valid Avamar proxy name. This argument is required. Use mccli client show to return a list of valid client names. If you supply a fully qualified client name (for example, <code>/clients/MyClient</code>), then the <code>--proxy-domain</code> argument is ignored.

Event codes

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

This command removes proxy backupproxy225 from the Test1 group:

```
mccli group remove-proxy --domain=/vcenter-1.example.com --name=Test1
  --proxy-domain=/clients --proxy-name=backupproxy225
```

```
0,24002,Proxy client mappings of a group successfully updated.
```

```
Attribute      Value
-----
group          /vcenter-1.example.com/Test1
```


group replicate

The **mccli group replicate** command replicates the specified group. It is equivalent to the Avamar Administrator **Replicate Now** command.

Synopsis

```
mccli group replicate [--domain=String(/)] --name=String
  [--wait [=MIN]]
```

Command options

Table 80 Command options for **mccli group replicate**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group to replicate. String must be a valid replication group name. This argument is required. Use mccli group show to return a list of valid replication group names.
<code>--wait [=MIN]</code>	Specifies a time period in minutes (MIN) that this command waits for the initiated operation to complete. If the operation completes before then, status is shown sooner. This option also shows more detailed status. If <code>--wait=0</code> or <code>--wait</code> (no value) is supplied, the command waits indefinitely for the initiated operation to complete.

Event codes

22226 Group disabled.
 22801 Login failure.
 30978 No proxy for replication.
 30930 Expired Retention (Expiration).
 22441 Scheduled client replicate (Success).
 22234 Group does not exist.
 23028 Invalid group type.

group show

The **mccli group show** command returns information for all groups.

Synopsis

```
mccli group show [--clients=Boolean(false)] [--domain=String(/)]
  [--name=String] [--proxy-clients=Boolean(false)]
  [--recursive=Boolean(false)] [--verbose=Boolean(false)]
```

Command options

Table 81 Command options for **mccli group show**

Option	Description
<code>--clients=Boolean(false)</code>	If true, then all clients that belong to the group are listed.
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	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, <code>/MyDomain/MyGroup</code>), then the <code>--domain</code> argument is ignored.
<code>--proxy-clients=Boolean(false)</code>	If true, then the command shows all proxies belonging to this group. This option is only valid for vCenter groups.
<code>--recursive=Boolean(false)</code>	If true, then the command shows all groups in the domain and any subdomains specified by the <code>--domain=String</code> argument. If false or not supplied, then the command only shows groups in the domain. Subdomains are not displayed.
<code>--verbose=Boolean(false)</code>	If true, then detailed information is returned. If false or not supplied, then only the group names are returned.

Event codes

22542 Domain does not exist.

Examples

This command returns a simple list of all groups in use:

```
mccli group show
```

```
0,23000,CLI command completed successfully.
command completed successfully
Group          Domain      Type
-----
Default Group /           Normal
```

This 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	/
Type	Normal
Enabled	false
Dataset	Default Dataset
Schedule	Default Schedule
Retention Policy	Default Retention
Encryption Method	High

This command returns all proxies belonging to the Default Virtual Machine Group in vCenter 10.31.183.85:

```
mccli group show --proxy-clients=true
--name=/10.31.183.85/Default Virtual Machine Group
```

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

Group	Proxy
/10.31.183.85/Default Virtual Machine Group	clients/localhost-proxy-1

group show-members

The **mccli group show-members** command returns a list of all members for a group.

Synopsis

```
mccli group show-members [--domain=String(/)] --name=String
```

Command options

Table 82 Command options for **mccli group show-members**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the group specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the group from which to show client members. String must be a valid group name. This argument is required. Use mccli group show to return a list of valid group names. If you supply a fully qualified group name (for example, /MyDomain/MyGroup), then the <code>--domain</code> argument is ignored.

Event codes

The following event code is associated with the **mccli group show-members** command.

22234 Group does not exist.

Examples

This command returns a simple list of all clients in the Default Virtual Machine Group:

```
mccli group show-client-members
--name=/10.31.183.55/Default Virtual Machine Group --xml
```

```
<CLIOutput>
  <Results>
    <ReturnCode>0</ReturnCode>
    <EventCode>23000</EventCode>
    <EventSummary>CLI command completed successfully.</EventSummary>
  </Results>
  <Data>
    <Row>
      <Group>/10.31.183.55/Default Virtual Machine Group</Group>
      <GroupType>Default Virtual Machine Group</GroupType>
      <Client>/10.31.183.55/F02</Client>
      <ClientType>Virtual Container</ClientType>
    </Row>
    <Row>
      <Group>/10.31.183.55/Default Virtual Machine Group</Group>
      <GroupType>Default Virtual Machine Group</GroupType>
      <Client>/10.31.183.55/HleDynamicClients/TEST1</Client>
      <ClientType>Virtual Machine</ClientType>
    </Row>
    <Row>
      <Group>/10.31.183.55/Default Virtual Machine Group</Group>
      <GroupType>Default Virtual Machine Group</GroupType>
      <Client>/10.31.183.55/APP1</Client>
      <ClientType>Virtual Container</ClientType>
    </Row>
    <Row>
      <Group>/10.31.183.55/Default Virtual Machine Group</Group>
      <GroupType>Default Virtual Machine Group</GroupType>
      <Client>/10.31.183.55/HleDynamicClients/ABCD1</Client>
      <ClientType>Virtual Machine</ClientType>
    </Row>
    <Row>
      <Group>/10.31.183.55/Default Virtual Machine Group</Group>
      <GroupType>Default Virtual Machine Group</GroupType>
      <Client>/10.31.183.55/HleDynamicClients/VA1</Client>
      <ClientType>Virtual Container</ClientType>
    </Row>
  </Data>
</CLIOutput>
```

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 | agent | backup | checkpoint | client | dataset |  
dd | domain | dump | event | group | help | mcs | plugin | retention |  
schedule | server | user | vcenter | version | vmcache}
```

mcs

The **mccli mcs** resource is used to view status and control various MCS functions. Commands are:

- ◆ “mcs import” on page 143
- ◆ “mcs list” on page 144
- ◆ “mcs reboot-proxy” on page 144
- ◆ “mcs resume-scheduler” on page 145
- ◆ “mcs scheduler-status” on page 145
- ◆ “mcs stop” on page 145
- ◆ “mcs suspend-scheduler” on page 145
- ◆ “mcs waitforflushcomplete” on page 146

mcs import

The **mccli mcs import** command accepts an XML file created by the **group export** command 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 --file=String [--force=Boolean(false)]
  --target-domain=String [--prefix=String]
```

Command options

Table 83 Command options for **mccli mcs import**

Option	Description
<code>--file=String</code>	Specifies the name of the import XML file. This option is required.
<code>--force=Boolean(false)</code>	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.
<code>--prefix=String</code>	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.
<code>--target-domain=String</code>	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 <code>_1</code> , <code>_2</code> , <code>_3</code> and so forth is appended to the name of the newly imported policy object.

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 and lists all groups and group policy objects (that is, dataset, schedule, and retention policy) in that file.

Synopsis

```
mccli mcs list --file=String
```

Command options

Table 84 Command options for **mccli mcs list**

Option	Description
<code>--file=String</code>	Specifies the name of the import XML file. This option is required.

mcs reboot-proxy

The **mccli mcs reboot-proxy** command reboots VMware proxies.

Synopsis

```
mcs reboot-proxy [--all=Boolean(false) | --name=String]
```

Command options

Table 85 Command options for **mccli mcs reboot-proxy**

Option	Description
<code>--all=Boolean(false)</code>	If supplied, reboot all known Avamar proxies.
<code>--name=String</code>	Specifies the proxies to reboot. String must be a fully qualified Avamar proxy name. You can supply multiple <code>--name</code> options on the same command line.

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
```

Event codes

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
```

mcs stop

The **mccli mcs stop** command shuts down the MCS.

Synopsis

```
mccli mcs stop
```

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.

Synopsis

```
mccli mcs suspend-scheduler
```

Event codes

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
```

plugin

The **mccli plugin** resource is used to manage client plug-ins. Commands are:

- ◆ [“plugin show” on page 147](#)
- ◆ [“plugin update” on page 147](#)

plugin show

The **plugin show** command shows summary properties for all client plug-ins.

Synopsis

```
mccli plugin show
```

plugin update

NOTICE

The **plugin update** command is strictly reserved for EMC internal use only.

Event codes

21010 Plugin catalog dynamically updated.

retention

The **mccli retention** resource is used to create and manage retention policies on the Avamar server. Commands are:

- ◆ “retention add” on page 149
- ◆ “retention copy” on page 151
- ◆ “retention delete” on page 152
- ◆ “retention edit” on page 153
- ◆ “retention show” on page 154

Advanced retention calendar units

All advanced retention options, which begin with --adv-, require a string value in the following format:

Table 86 Advanced retention option string values

Format	Description	Examples
+nnnD	Specifies the number (nnn) of calendar days (D).	<ul style="list-style-type: none"> • Specify +5D to retain five daily backups. • Specify +13D to retain 13 daily backups.
+nnW	Specifies the number (nn) of calendar weeks (W).	<ul style="list-style-type: none"> • 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).	<ul style="list-style-type: none"> • 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).	<ul style="list-style-type: none"> • 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 [--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)]
```

Command options

Table 87 Command options for **mccli retention add**

Option	Description
<code>--adv-daily=String(+1W)</code>	Specifies the daily advanced retention setting. String must be <code>+nnnD</code> , <code>+nnW</code> , <code>+nnM</code> , or <code>+nnY</code> .
<code>--adv-monthly=String(+1Y)</code>	Specifies the monthly advanced retention setting. String must be <code>+nnnD</code> , <code>+nnW</code> , <code>+nnM</code> , or <code>+nnY</code> .
<code>--adv-weekly=String(+1M)</code>	Specifies the weekly advanced retention setting. String must be <code>+nnnD</code> , <code>+nnW</code> , <code>+nnM</code> , or <code>+nnY</code> .
<code>--adv-yearly=String(+1Y)</code>	Specifies the yearly advanced retention setting. String must be <code>+nnnD</code> , <code>+nnW</code> , <code>+nnM</code> , or <code>+nnY</code> .
<code>--basic=String(NO_EXPIRATION)</code>	Specifies basic retention, which assigns a fixed expiration date. String must be one of the following: <ul style="list-style-type: none"> • <code>YYYY-MM-DD</code> to specify an explicit date • <code>+nn{D W M Y}</code> to specify a duration from today (for example, <code>+4W</code> specifies four weeks from today) • <code>NO_EXPIRATION</code>
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the retention policy specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies a name for the new retention policy. This argument is required. If you supply a fully qualified retention policy name (for example, <code>/MyDomain/MyRetentionPolicy</code>), then the <code>--domain</code> argument is ignored.
<code>--override=Boolean(false)</code>	If true, then basic retention settings are overridden in favor of advanced retention settings. This option only applies to scheduled backups.

Event codes

- 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 [--domain=String()] --name=String
  [--new-domain=String] --new-name=String
```

Command options

Table 88 Command options for **mccli retention copy**

Option	Description
<code>--domain=String()</code>	Specifies the Avamar server domain that contains the retention policy specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the retention policy to copy. String must be a valid retention policy name. This argument is required. Use mccli retention show to return a list of valid retention policy names. If you supply a fully qualified retention policy name (for example, <code>/MyDomain/MyRetentionPolicy</code>), then the <code>--domain</code> argument is ignored.
<code>--new-domain=String</code>	Specifies an optional new Avamar server destination domain for the copied retention policy.
<code>--new-name=String</code>	Specifies a name for the new retention policy. This argument is required.

Event codes

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 [--domain=String()] --name=String
```

Command options

Table 89 Command options for **mccli retention delete**

Option	Description
<code>--domain=String()</code>	Specifies the Avamar server domain that contains the retention policy specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the retention policy to delete. String must be a valid retention policy name. This argument is required. Use mccli retention show to return a list of valid retention policy names. If you supply a fully qualified retention policy name (for example, <code>/MyDomain/MyRetentionPolicy</code>), then the <code>--domain</code> argument is ignored.

Event codes

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 [--adv-daily=String] [--adv-monthly=String]
  [--adv-weekly=String] [--adv-yearly=String] [--basic=String]
  [--domain=String(/)] --name=String [--override=Boolean]
```

Command options

Table 90 Command options for **mccli retention edit**

Option	Description
<code>--adv-daily=String(+1W)</code>	Specifies the daily advanced retention setting. String must be +nnnD, +nnW, +nnM, or +nnY.
<code>--adv-monthly=String(+1Y)</code>	Specifies the monthly advanced retention setting. String must be +nnnD, +nnW, +nnM, or +nnY.
<code>--adv-weekly=String(+1M)</code>	Specifies the weekly advanced retention setting. String must be +nnnD, +nnW, +nnM, or +nnY.
<code>--adv-yearly=String(+1Y)</code>	Specifies the yearly advanced retention setting. String must be +nnnD, +nnW, +nnM, or +nnY.
<code>--basic=String</code>	Specifies basic retention, which assigns a fixed expiration date. String must be one of the following: <ul style="list-style-type: none"> 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
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the retention policy specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the retention policy to edit. String must be a valid retention policy name. This argument is required. Use mccli retention show to return a list of valid retention policy names. If you supply a fully qualified retention policy name (for example, /MyDomain/MyRetentionPolicy), then the <code>--domain</code> argument is ignored.
<code>--override=Boolean</code>	If true, then basic retention settings are overridden in favor of advanced retention settings. This option only applies to scheduled backups.

Event codes

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 [--domain=String(/)] [--name=String]
  [--recursive=Boolean(false)] [--verbose=Boolean(false)]
```

Command options

Table 91 Command options for **mccli retention show**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the retention policy specified by the <code>--name</code> argument.
<code>--name=String</code>	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, <code>/MyDomain/MyRetentionPolicy</code>), then the <code>--domain</code> argument is ignored.
<code>--recursive=Boolean(false)</code>	If true, then the command shows all retention policies in the domain and any subdomains specified by the <code>--domain=String</code> argument. If false or not supplied, then the command only shows retention policies in the domain. Subdomains are not examined.
<code>--verbose=Boolean(false)</code>	If true, then detailed information is returned.

Event codes

22289 Retention policy does not exist.

22542 Domain does not exist.

Examples

This 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
```

This 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

schedule

The **mccli schedule** resource is used to create and manage schedules on the Avamar server. Commands are:

- ◆ “schedule add” on page 156
- ◆ “schedule copy” on page 158
- ◆ “schedule delete” on page 159
- ◆ “schedule edit” on page 160
- ◆ “schedule show” on page 162
- ◆ “schedule show-timezones” on page 163

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]
```

Command options

Table 92 Command options for **mccli schedule add** (page 1 of 2)

Option	Description
<code>--days=String</code>	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).
<code>--desc=String</code>	Specifies a short text description of the schedule.
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the schedule specified by the <code>--name</code> argument.
<code>--duration=String</code>	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 <code>--duration</code> , then the default duration for a daily schedule is one hour and the default duration for a weekly schedule is to eight hours.
<code>--name=String</code>	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 <code>--domain</code> argument is ignored.
<code>--nth-day=String</code>	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.
<code>--on-demand=Boolean(false)</code>	If true, the new schedule is designated as an on-demand schedule. If set true, this option overrides other schedule type designations.

Table 92 Command options for **mccli schedule add** (page 2 of 2)

Option	Description
<code>--start=String</code>	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 <code>--start</code> option does not apply for daily schedules. If you do not specify <code>--start</code> for a weekly schedule, then the default start time is 10 p.m. in the timezone in which the schedule was created.
<code>--time=String</code>	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. Multiple <code>--time</code> options can be specified on the same command line.
<code>--tz=String</code>	Specifies the timezone for the schedule. If not supplied, then the local timezone is used. Use schedule show-timezones to return a list of valid retention timezone names.
<code>--week=String</code>	Creates a schedule that runs on the specified week each month. String must be first, second, third, fourth, or last.

Event codes

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

This command 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
```

This command 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 [--domain=String()] --name=String
  [--new-domain=String] --new-name=String
```

Command options

Table 93 Command options for **mccli schedule copy**

Option	Description
<code>--domain=String()</code>	Specifies the Avamar server domain that contains the schedule specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the schedule to copy. String must be a valid schedule name. This argument is required. Use mccli schedule show to return a list of valid schedule names. If you supply a fully qualified schedule name (for example, /MyDomain/MySchedule), then the <code>--domain</code> argument is ignored.
<code>--new-domain=String</code>	Specifies an optional new Avamar server destination domain for the copied schedule.
<code>--new-name=String</code>	Specifies a name for the new schedule. This argument is required.

Event codes

22213 Schedule created.
 22248 Schedule already exists.
 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 [--domain=String(/)] --name=String
```

Command options

Table 94 Command options for **mccli schedule delete**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the schedule specified by the <code>--name</code> argument.
<code>--name=String</code>	Specifies the schedule to delete. String must be a valid schedule name. This argument is required. Use mccli schedule show to return a list of valid schedule names. If you supply a fully qualified schedule name (for example, <code>/MyDomain/MySchedule</code>), then the <code>--domain</code> argument is ignored.

Event codes

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]
```

Command options

Table 95 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).
--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 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.
--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.
--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.
--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. Multiple --time options can be specified on the same command line.

Table 95 Command options for `mccli schedule edit` (page 2 of 2)

Option	Description
<code>--tz=String</code>	Specifies the timezone for the schedule. If not supplied, then the local timezone is used. Use <code>schedule show-timezones</code> to return a list of valid timezone names.
<code>--week=String</code>	Edits the schedule to run on the specified week each month. String must be first, second, third, fourth, or last.

Event codes

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

This command 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 [--domain=String(/)] [--name=String]
  [--recursive=Boolean(false)] [--verbose=Boolean(false)]
```

Command options

Table 96 Command options for **mccli schedule show**

Option	Description
<code>--domain=String(/)</code>	Specifies the Avamar server domain that contains the schedule specified by the <code>--name</code> argument.
<code>--name=String</code>	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, <code>/MyDomain/MySchedule</code>), then the <code>--domain</code> argument is ignored.
<code>--recursive=Boolean(false)</code>	If true, then the command shows all schedules in the domain and any subdomains specified by the <code>--domain=String</code> argument. If false or not supplied, then the command only shows schedules in the domain. Subdomains are not examined.
<code>--verbose=Boolean(false)</code>	If true, then detailed schedule information is returned. If false or not supplied, then only schedule names are returned.

Event codes

22249 Schedule does not exist.

22542 Domain does not exist.

Examples

This 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                                  2013-09-20 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
```

Table 97 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
```

server

The **mccli server** resource is used to monitor various Avamar server functions. Commands are:

- ◆ “server show-prop” on page 164
- ◆ “server show-services” on page 168
- ◆ “server show-util” on page 168
- ◆ “server start-service” on page 169
- ◆ “server stop-service” on page 169

server show-prop

The **mccli server show-prop** command returns detailed properties for the Avamar server, similar to what is shown in the Avamar Administrator Server window Server Management tab. 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 [--gc=Boolean(false)]
  [--maintenance=Boolean(false)] [--module=String] [--node=String]
  [--partition=String] [--verbose]
```

Command options

Table 98 Command options for **mccli server show-prop** (page 1 of 2)

Option	Description
--gc=Boolean(false)	If true, then detailed garbage collection information is shown.
--maintenance=Boolean(false)	If true, then detailed maintenance activity information is shown.
--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.

Table 98 Command options for **mccli server show-prop** (page 2 of 2)

Option	Description
<code>--partition=String</code>	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 <code>--node=String</code> option. A value of ALL returns status for all partitions.
<code>--verbose</code>	If supplied, maximum information is returned.

Notes

- ◆ The `--module=String` and `--node=String` options are mutually exclusive.
- ◆ The `--partition=String` option requires `--node=String`.

Examples

This 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                               2013-10-01 04:00:28 PDT
Last validated checkpoint                    2013-09-30 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
```

This 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                           7.1.0-nnn
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
```

Table 99 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.

This command returns detailed garbage collection information:

mccli server show-prop --gc=true

Listening for transport dt_socket at address: 8003
0,23000,CLI command completed successfully.

Attribute	Value
Status	idle
Result	OK
Start time	2013-1-10 08:15:00 UTC
End time	2013-1-10 08:15:17 UTC
Passes	3
Bytes recovered	81.9 KB
Chunks deleted	16
Index stripes	6
Index stripes processed	6

This command returns detailed maintenance activity information:

mccli server show-prop --maintenance=true

Listening for transport dt_socket at address: 8003
0,23000,CLI command completed successfully.

Attribute	Value
Suspended	No

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 [--service=String]
```

Command options

Table 100 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.

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 [--module=String | --node=String]
```

Command options

Table 101 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.

Notes

The --module=String and --node=String options are mutually exclusive.

server start-service

The **mccli server start-service** command starts a service that is currently stopped.

Synopsis

```
mccli server start-service --service=String
```

Command options

Table 102 Command options for **mccli server start-service**

Option	Description
<code>--service=String</code>	Name of the service to start. String must be a valid service name and it must be stopped. Use mccli server show-services to return a list of valid service names.

server stop-service

The **mccli server stop-service** command stops a service that is currently running.

Synopsis

```
mccli server stop-service --service=String
```

Command options

Table 103 Command options for **mccli server stop-service**

Option	Description
<code>--service=String</code>	Name of the service to stop. String must be a valid service name and it must be running. Use mccli server show-services to return a list of valid service names.

user

The **mccli user** resource is used to create and manage backup user accounts on the Avamar server. Commands are:

- ◆ “user add” on page 171
- ◆ “user authenticate” on page 174
- ◆ “user delete” on page 175
- ◆ “user edit” on page 176
- ◆ “user show” on page 178
- ◆ “user show-auth” on page 178

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:

- ◆ **user show-auth**
- ◆ **client show**
- ◆ **domain show**

user add

The **mccli user add** command creates a backup user account for a client or domain.

Synopsis

```
mccli user add [--authenticator=String(Axion)] --client-domain=String
  [--client-name=String] --name=String [--password=String]
  [--password-confirm=String] --role=String
```

Command options

Table 104 Command options for **mccli user add** (page 1 of 2)

Option	Description
<code>--authenticator=String(Axion)</code>	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 to return a list of valid authentication systems.
<code>--client-domain=String</code>	Specifies the domain for the new user. String must be a valid domain name. This argument is required. If you supply both <code>--client-domain</code> and <code>--client-name</code> , then the user is added to the client. If you supply only <code>--client-domain</code> , then the user is added to the domain. Use mccli domain show to return a list of valid domain names.
<code>--client-name=String</code>	Specifies the client to which to add the new user account. String must be a valid client name. Use mccli client show to return a list of valid client names.
<code>--name=String</code>	Specifies the user ID or name for the new user account. This argument is required.
<code>--password=String</code>	Specifies the password for the new user. If you specify an external authentication system using the <code>--authenticator</code> argument, then do not supply the <code>--password</code> and <code>--password-confirm</code> arguments. This is because the external authentication system manages the password for the user.
<code>--password-confirm=String</code>	Specifies the password confirmation string. If you supply <code>--password=String</code> , then String must be the same value as <code>--password=String</code> . This is intended to prevent a single incorrect entry of the password. If you specify an external authentication system using the <code>--authenticator</code> argument, then do not supply the <code>--password</code> and <code>--password-confirm</code> arguments. This is because the external authentication system manages the password for the user.

Table 104 Command options for `mccli user add` (page 2 of 2)

Option	Description
<code>--privilege=String</code>	Specifies the privilege level for the user. Beginning with version 4.0, use of this option is deprecated in favor of <code>--role</code> .
<code>--role=String</code>	Specifies the role for the user. String must be one of the following: <ul style="list-style-type: none"> • ActivityOperator • Administrator • Auditor • Backup • BackupOperator • BackupRestore • BackupRestoreOperator • Restore • RestoreIgnoreFilePermissions • RestoreOperator You can only use the <code>RestoreIgnoreFilePermissions</code> role if you use an external authentication system. The <i>EMC Avamar Administration Guide</i> provides additional information about roles.

Event codes

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

- ◆ Beginning with version 4.0, the `--privilege` option is deprecated in favor of `--role`.
- ◆ Beginning with version 4.0, 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

This 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"/>

This 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 [--authenticator=String (Axion) ]
  --client-domain=String [--client-name=String] --name=String
  --password=String
```

Command options

Table 105 Command options for **mccli user authenticate**

Option	Description
<code>--authenticator=String (Axion)</code>	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 to return a list of valid authentication systems.
<code>--client-domain=String</code>	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 to return a list of valid domain names.
<code>--client-name=String</code>	Specifies the name of the client associated with the user account. String must be a valid client name. Use mccli client show to return a list of valid client names.
<code>--name=String</code>	Specifies the user ID or name to authenticate.
<code>--password=String</code>	Specifies the password to authenticate the user. This argument is required.

Event codes

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 --authenticator=String --client-domain=String
  [--client-name=String] --name=String
```

Command options

Table 106 Command options for **mccli user delete**

Option	Description
<code>--authenticator=String</code>	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 to return a list of valid authentication systems.
<code>--client-domain=String</code>	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 <code>--client-domain</code> and <code>--client-name</code> , then the user is deleted from the client. If you supply only <code>--client-domain</code> , then the user is deleted from the domain. Use mccli domain show to return a list of valid domain names.
<code>--client-name=String</code>	Specifies the client name associated with the user account. String must be a valid client name. Use mccli client show to return a list of valid client names.
<code>--name=String</code>	Specifies the user ID or name of the user to delete.

Event codes

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 --authenticator=String --client-domain=String
  [--client-name=String] --name=String [--password=String]
  [--password-confirm=String] [--role=String]
```

Command options

Table 107 Command options for **mccli user edit** (page 1 of 2)

Option	Description
<code>--authenticator=String</code>	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 to return a list of valid authentication systems.
<code>--client-domain=String(/)</code>	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 <code>--client-domain</code> and <code>--client-name</code> , then the user resides on the client. If you supply only <code>--client-domain</code> , then the user resides in the domain. Use mccli domain show to return a list of valid domain names.
<code>--client-name=String</code>	Specifies the name of the client associated with the user account. String must be a valid client name. Use mccli client show to return a list of valid client names.
<code>--name=String</code>	Specifies the user ID or name for the user account. String must be a valid user name. This argument is required. Use mccli domain show to return a list of valid domain names.
<code>--password=String</code>	Specifies the password for the user.
<code>--password-confirm=String</code>	Specifies the password confirmation string. If you supply <code>--password=String</code> , then String must be the same value as <code>--password=String</code> . This is intended to prevent a single incorrect entry of the password.

Table 107 Command options for `mccli user edit` (page 2 of 2)

Option	Description
<code>--privilege=String</code>	Specifies the privilege level for the user. Beginning with version 4.0, use of this option is deprecated in favor of <code>--role</code> .
<code>--role=String</code>	Specifies the role for the user. String must be one of the following: <ul style="list-style-type: none"> • 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 <i>EMC Avamar Administration Guide</i> provides additional information about roles.

Event codes

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

- ◆ Beginning with version 4.0, the `--privilege` option is deprecated in favor of `--role`.
- ◆ Beginning with version 4.0, 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 [--authenticator=String(Axion)]
  [--client-domain=String(/)] [--client-name=String] [--name=String]
  [--recursive=Boolean(false)]
```

Command options

Table 108 Command options for **mccli user show**

Option	Description
<code>--authenticator=String(Axion)</code>	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 to return a list of valid authentication systems.
<code>--client-domain=String(/)</code>	Specifies the domain associated with the client.
<code>--client-name=String</code>	Specifies the name of the client to examine for user access. String must be a valid client name. Use mccli client show to return a list of valid client names.
<code>--name=String</code>	Specifies the user ID or name.
<code>--recursive=Boolean(false)</code>	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.

Event codes

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
```

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 [--container-path=String] [--datacenter=String]
  [--esx-host=String] [--folder=String] --name=String
  [--recursive=Boolean(false)] [--type=String(VM)]
  [--vsphere-hosts-clusters-view=Boolean(false)]
```

Command options

Table 109 Command options for **mccli vcenter browse** (page 1 of 2)

Option	Description
<code>--container-path=String</code>	Specifies a container path within the specified datacenter. String can be a vApp container in the VMs and Templates view, or a resource pool in the Hosts and Clusters view.
<code>--datacenter=String</code>	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 option is only valid when browsing for virtual machines or vApps.
<code>--esx-host=String</code>	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 or vApps in the vSphere Hosts and Clusters view (that is, <code>--vsphere-hosts-clusters-view=true</code>).
<code>--folder=String</code>	Specifies a folder path within the specified datacenter. This option is only valid when browsing for virtual machines or vApps in the vSphere VMs and Templates view (that is, <code>--vsphere-hosts-clusters-view=false</code>).
<code>--name=String</code>	Specifies the vCenter name. This argument is required.
<code>--recursive=Boolean(false)</code>	Recursively browses all levels of the specified folder path in a datacenter. This option is only valid when browsing for virtual machines or vApps.

Table 109 Command options for `mccli vcenter browse` (page 2 of 2)

Option	Description
<code>--type=String (VM)</code>	Specifies the type of entity that should be searched for by this command. String must be one of the following: <ul style="list-style-type: none"> • DATACENTER—Browse for datacenters. Returns datacenter names and locations. • DATASTORE—Browse datastores. Returns detailed datastore information such as Datacenters, Name, Type, Accessible/Not, Hosts. • NETWORK—Browse for virtual networks. Returns all available networks and their hosts. • VAPP—Browse for vApps. • 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.
<code>--vsphere-hosts-clusters-view=Boolean(false)</code>	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 or vApps.

Examples

This command recursively browses vCenter `vcenter-1.example.com` for virtual machines:

```
mccli vcenter browse --name=vcenter-1.example.com --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 virtualmachine/RHEL564BUILDDOWNLOAD224
      </Location>
      <Template>No</Template>
      <PoweredOn>Yes</PoweredOn>
      <ChangedBlock>No</ChangedBlock>
      <Protected>No</Protected>
    </Row>
  </Data>
</CLIOutput>
```

This command browses vcenter-1.example.com for datastores:

```
mccli vcenter browse --name=vcenter-1.example.com --type=DATASTORE --xml
```

Name	Type	Accessible	Hosts	Datacenter
Storage1	VMFS	Yes	10.31.183.7	VAAYU-DEV-WIN
Storage2	VMFS	Yes	10.31.183.7	VAAYU-DEV-WIN

This command browses vCenter vcenter-1.example.com for virtual machines in datacenter VAAYU-DEV-WIN Hosts and Clusters view:

```
mccli vcenter browse --name=vcenter-1.example.com --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/RHEL564BUILDDOWNLOAD224
      </Location>
      <Template>No</Template>
      <PoweredOn>Yes</PoweredOn>
      <ChangedBlock>No</ChangedBlock>
      <Protected>No</Protected>
    </Row>
  </Data>
```

This command recursively returns a list of virtual networks and the hosts that use them:

```
mccli vcenter browse --type=NETWORK --name=10.31.183.55
```

Network	Host
VM Network	10.31.183.14
VM Network	10.31.183.17
VM Network	10.31.183.7

This command browses for vApps in datacenter DCF1/DCF2 in folders FO1/FO2:

```
mccli vcenter browse --type=VAPP --name=10.31.183.55  
--datacenter=DCF1/DCF2 --folder=FO1/FO2 --recursive=true --xml
```

Listening for transport dt_socket at address: 8003

```
<CLIOutput>  
  <Results>  
    <ReturnCode>1</ReturnCode>  
    <EventCode>23999</EventCode>  
    <EventSummary />  
  </Results>  
  <Data>  
    <Row>  
      <Name>VA1</Name>  
      <GuestOS>N/A</GuestOS>  
      <Server>10.31.183.17</Server>  
      <Location>/DCF1/DCF2/vm/FO1/FO2/VA1/</Location>  
      <Template>No</Template>  
      <PoweredOn>No</PoweredOn>  
      <ChangedBlock>No</ChangedBlock>  
      <Protected>No</Protected>  
    </Row>  
  </Data>  
</CLIOutput>
```

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
```

vmcache

The **mccli vmcache** resource is used to assist with debugging possible vCenter data cache synchronization issues. Commands are:

- ◆ “[vmcache show](#)” on page 184
- ◆ “[vmcache sync](#)” on page 185

vmcache show

The **mccli vmcache show** command is used to examine cached vCenter data.

NOTICE

The **vmcache show** command is strictly reserved for EMC internal use only.

Synopsis

```
mccli vmcache show --domain=String --name=String
```

Options

Table 110 Command options for **mccli vmcache show**

Option	Description
<code>--domain=String</code>	Specifies the Avamar server domain with the virtual machine or container entity specified by the <code>--name</code> argument. This argument is required.
<code>--name=String</code>	Specifies the virtual machine or container entity for which to show cached vCenter data. This argument is required.

Examples

This command shows cached vCenter data for a single virtual machine:

```
mccli vmcache show --domain=vcenter-1.example.com/ContainerClients
--name=my-vm
```

This command shows cached vCenter data for a container entity:

```
mccli vmcache show --domain=vcenter-1.example.com --name=my-container
```


vmcache sync

The **mccli vmcache sync** command is used to manually synchronize cached vCenter data.

NOTICE

The **vmcache sync** command is strictly reserved for EMC internal use only.

Synopsis

```
mccli vmcache sync --domain=String --name=String
```

Options

Table 111 Command options for **mccli vmcache sync**

Option	Description
--domain=String	Specifies the Avamar server domain with the virtual machine or container entity specified by the --name argument. This argument is required.
--name=String	Specifies the virtual machine or container entity for which to synchronize cached vCenter data. This argument is required.

Examples

This command synchronizes a single virtual machine's cached vCenter data:

```
mccli vmcache sync --name=my-vm
  --domain=vcenter-1.example.com/ContainerClients
```

This command synchronizes a container entity's cached vCenter data:

```
mccli vmcache sync --name=my-container --domain=vcenter-1.example.com
```

