Copyright and Trademark Notices

This document contains information proprietary to EMC. Due to continuing product development, product specifications and capabilities are subject to change without notice. You may not disclose or use any proprietary information or reproduce or transmit any part of this document in any form or by any means, electronic or mechanical, for any purpose, without written permission from EMC.

EMC has made every effort to keep the information in this document current and accurate as of the date of publication or revision. However, EMC does not guarantee or imply that this document is error free or accurate with regard to any particular specification. In no event will EMC be liable for direct, indirect, incidental or consequential damages resulting from any defect in the documentation, even if advised of the possibility of such damages. No EMC agent or employee is authorized to make any modification, extension or addition to the above statements.

EMC may have patents, patent applications, trademarks, copyrights or other intellectual property rights covering subject matter in this document. The furnishing of this document does not provide any license to these patents, trademarks, copyrights or other intellectual property.

The Avamar Agent for Microsoft Windows incorporates Open Transaction Manager (OTM), a product of Columbia Data Products, Inc. (CDP). CDP assumes no liability for any claim that may arise regarding this incorporation. In addition, EMC disclaims all warranties, both express and implied, arising from the use of Open Transaction Manager. Copyright 1999-2002 Columbia Data Products, Inc. Altamonte Springs. All rights reserved.

Avamar, RAIN and AvaSphere are trademarks or registered trademarks of EMC in the US and/or other countries.

All other product names and/or slogans mentioned herein may be trademarks or registered trademarks of their respective companies. All information presented here is subject to change and intended for general information.

Copyright 2002-2009 EMC. All rights reserved.

Protected by US Patents No. 6,704,730, 6,810,398 and patents pending.

Printed in the USA.
# Table of Contents

- **Foreword** ................................................................. 7
  - Scope and Intended Audience ........................................... 7
  - Product Information .................................................... 7
  - Your Comments .......................................................... 8
  - Typeface Conventions ................................................... 8
  - Notes, Tips and Warnings ................................................. 9

- **Introduction** ............................................................. 10
  - System Requirements .................................................. 10
  - Capabilities and Limitations .......................................... 11

- **Installation and Configuration** ..................................... 12
  - Install the Java Runtime Environment (JRE) ......................... 12
  - Install the MCCLI RPM Package ....................................... 13
  - Run the avsetup_mccli Utility ......................................... 14
  - Additional Configuration Notes ...................................... 16
    - mccli.xml Preferences File ........................................ 16
    - mcclimcs.xml Default Options File ............................... 16

- **mccli Command Reference** .......................................... 19
  - General Programming Notes .......................................... 19
    - Command Line Elements .............................................. 19
    - Data Types .......................................................... 20
    - Default Values ..................................................... 20
    - Optional and Required Arguments .................................. 20
    - Pattern Matching Operators ....................................... 21
    - Output Description .................................................. 22
    - Global Command Line Syntax and Behavior ....................... 22
    - Activity Resource Overview ....................................... 27
      - activity cancel .................................................... 27
      - activity show .................................................... 28
    - Agent Resource Overview ........................................... 29
      - agent show .......................................................... 29
    - Backup Resource Overview ......................................... 30
      - backup delete ....................................................... 31
      - backup edit ......................................................... 33
      - backup restore ..................................................... 35
      - backup show .......................................................... 37
backup validate ................................................................. 40
Checkpoint Resource Overview ........................................ 42
checkpoint cancel-validate .............................................. 42
checkpoint create ............................................................ 43
checkpoint delete ............................................................ 44
checkpoint show .............................................................. 45
checkpoint validate ........................................................ 46
Client Resource Overview ................................................. 47
client activate ............................................................... 48
client add ................................................................. 49
client backup-dataset ..................................................... 54
client backup-group-dataset .......................................... 56
client backup-target ..................................................... 58
client delete ............................................................... 60
client edit ................................................................. 61
client invite ............................................................... 66
client load-bulk ........................................................... 67
client move ................................................................. 68
client retire ............................................................... 69
client show ............................................................... 70
client show-plugins ...................................................... 73
client snapup-dataset ................................................... 74
client snapup-group-dataset ........................................ 76
client snapup-target ..................................................... 79
client validate-bulk ...................................................... 81
Dataset Resource Overview ............................................ 82
dataset add ............................................................... 83
dataset add-exclude ...................................................... 84
dataset add-include ...................................................... 86
dataset add-option ....................................................... 88
dataset add-target ........................................................ 90
dataset copy ............................................................... 92
dataset delete ............................................................. 93
dataset delete-exclude .................................................. 94
dataset delete-include .................................................. 96
dataset delete-option ................................................... 98
dataset delete-target ................................................... 99
dataset replace ........................................................... 101
dataset show ............................................................. 103
Domain Resource Overview ............................................ 106
domain add ............................................................... 107
domain delete ............................................................ 108
domain edit ............................................................... 109
domain show ............................................................. 110
Dump Resource Overview ................................................ 111
dump clientcache ......................................................... 111
dump domaincache ...................................................... 112
dump jobcache ........................................................... 112
Event Resource Overview ................................................ 113
event ack ................................................................. 114
event get-info ............................................................ 117
event show ............................................................... 118
Group Resource Overview ............................................... 123
group add ............................................................... 124
group add-client ......................................................... 126
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>group backup</td>
<td>128</td>
</tr>
<tr>
<td>group copy</td>
<td>129</td>
</tr>
<tr>
<td>group delete</td>
<td>131</td>
</tr>
<tr>
<td>group edit</td>
<td>132</td>
</tr>
<tr>
<td>group export</td>
<td>134</td>
</tr>
<tr>
<td>group move-client</td>
<td>135</td>
</tr>
<tr>
<td>group remove-client</td>
<td>137</td>
</tr>
<tr>
<td>group show</td>
<td>139</td>
</tr>
<tr>
<td>group show-client-members</td>
<td>141</td>
</tr>
<tr>
<td>group snapup</td>
<td>142</td>
</tr>
<tr>
<td>mcs import</td>
<td>145</td>
</tr>
<tr>
<td>mcs list</td>
<td>147</td>
</tr>
<tr>
<td>mcs reboot-proxy</td>
<td>148</td>
</tr>
<tr>
<td>mcs resume-scheduler</td>
<td>149</td>
</tr>
<tr>
<td>mcs scheduler-status</td>
<td>150</td>
</tr>
<tr>
<td>mcs stop</td>
<td>151</td>
</tr>
<tr>
<td>mcs suspend-scheduler</td>
<td>151</td>
</tr>
<tr>
<td>mcs waitforflushcomplete</td>
<td>152</td>
</tr>
<tr>
<td>plugin show</td>
<td>153</td>
</tr>
<tr>
<td>plugin update</td>
<td>153</td>
</tr>
<tr>
<td>retention add</td>
<td>154</td>
</tr>
<tr>
<td>retention copy</td>
<td>155</td>
</tr>
<tr>
<td>retention delete</td>
<td>157</td>
</tr>
<tr>
<td>retention edit</td>
<td>158</td>
</tr>
<tr>
<td>retention show</td>
<td>159</td>
</tr>
<tr>
<td>schedule add</td>
<td>163</td>
</tr>
<tr>
<td>schedule copy</td>
<td>164</td>
</tr>
<tr>
<td>schedule delete</td>
<td>167</td>
</tr>
<tr>
<td>schedule edit</td>
<td>168</td>
</tr>
<tr>
<td>schedule show</td>
<td>169</td>
</tr>
<tr>
<td>schedule show-timezones</td>
<td>171</td>
</tr>
<tr>
<td>server show-prop</td>
<td>175</td>
</tr>
<tr>
<td>server show-services</td>
<td>178</td>
</tr>
<tr>
<td>server show-util</td>
<td>179</td>
</tr>
<tr>
<td>snapup delete</td>
<td>182</td>
</tr>
<tr>
<td>snapup edit</td>
<td>183</td>
</tr>
<tr>
<td>snapup restore</td>
<td>185</td>
</tr>
<tr>
<td>snapup show</td>
<td>187</td>
</tr>
<tr>
<td>snapup validate</td>
<td>189</td>
</tr>
<tr>
<td>user add</td>
<td>191</td>
</tr>
<tr>
<td>user authenticate</td>
<td>192</td>
</tr>
<tr>
<td>user delete</td>
<td>195</td>
</tr>
<tr>
<td>user edit</td>
<td>197</td>
</tr>
<tr>
<td>user show</td>
<td>199</td>
</tr>
<tr>
<td>user show-auth</td>
<td>202</td>
</tr>
</tbody>
</table>

**Help Resource Overview** ......................................................... 143

**MCS Resource Overview** ....................................................... 144

**Retention Resource Overview** ............................................... 154

**Schedule Resource Overview** ............................................... 163

**Server Resource Overview** .................................................. 174

**Snapup Resource Overview** ................................................... 181

**User Resource Overview** ..................................................... 191
<table>
<thead>
<tr>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version Resource Overview</td>
<td>204</td>
</tr>
<tr>
<td>version show</td>
<td>204</td>
</tr>
</tbody>
</table>
FOREWORD

Scope and Intended Audience

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

**Intended Audience.** The information in this publication is primarily intended for system administrators who are responsible for installing software and maintaining servers and clients on a network.

**Prerequisites.** This publication assumes that the reader is familiar with the Avamar Administrator graphical management console as documented in the *Avamar System Administration Guide* and does not attempt to repeat information found in that publication.

Product Information

For current documentation, release notes, software updates, as well as information about EMC products, licensing and service, go to the EMC Powerlink web site at http://Powerlink.EMC.com.
Your Comments

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

SSGDocumentation@emc.com

Please include the following information:

- Product name and version
- Document name, part number and revision (for example, A01)
- Page numbers
- Other details that will help us address the documentation issue

Typeface Conventions

The following table provides examples of standard typeface styles used in this publication to convey various kinds of information.

<table>
<thead>
<tr>
<th>EXAMPLE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Click <strong>OK</strong>. - or - Select <strong>File &gt; Close</strong>.</td>
<td>Bold text denotes actual Graphical User Interface (GUI) buttons, commands, menus and options (any GUI element that initiates action). Also note in the second example that sequential commands are separated by a greater-than (&gt;) character. In this example, you are being instructed to select the <strong>Close</strong> command from the <strong>File</strong> menu.</td>
</tr>
<tr>
<td><strong>Type:</strong> cd /tmp</td>
<td>Bold fixed-width text denotes shell commands that must be entered exactly as they appear in this publication.</td>
</tr>
<tr>
<td><strong>--logfile=FILE</strong></td>
<td>All caps text often denotes a placeholder (token) for an actual value that must be supplied by the user. In this example, FILE would be an actual filename.</td>
</tr>
<tr>
<td><strong>Installation Complete.</strong></td>
<td>Regular (not bold) fixed-width text denotes command shell messages. It is also used to list code and file contents.</td>
</tr>
</tbody>
</table>
Notes, Tips and Warnings

The following kinds of notes, tips and warnings appear in this publication:

**IMPORTANT:** This is a warning. Warnings always contain information that if not heeded could result in unpredictable system behavior or loss of data.

**TIP:** This is a tip. Tips present optional information intended to improve your productivity or otherwise enhance your experience with our product. Tips never contain information that will cause a failure if ignored.

**NOTE:** This is a general note. Notes contain ancillary information intended to clarify a topic or procedure. Notes never contain information that will cause a failure if ignored.
INTRODUCTION

The Avamar Management Console Command Line Interface (MCCLI) is a Java software application that provides command line access to Avamar Administrator features and functions.

System Requirements

In order 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 following minimum requirements:

<table>
<thead>
<tr>
<th>REQUIREMENT</th>
<th>MINIMUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
<td>Red Hat Enterprise Linux Release 4 (64-bit)</td>
</tr>
<tr>
<td>RAM</td>
<td>256 MB.</td>
</tr>
<tr>
<td>Hard Drive Space</td>
<td>60 MB permanent hard drive space.</td>
</tr>
<tr>
<td>Network Interface</td>
<td>10baseT or higher, configured with latest drivers for your platform.</td>
</tr>
<tr>
<td>Java Runtime Environment</td>
<td>JRE 1.6</td>
</tr>
</tbody>
</table>
Capabilities and Limitations

This topic discusses the current capabilities and limitations of the MCCLI.

**Version Compatibility.** There is no version/compatibility checking/enforcement between the MCCLI and the Management Console Server (MCS). This allows a current version MCCLI client to run against a any supported Avamar server.

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

**Relative Path Filenames Not Supported.** The use of filenames containing relative paths is not currently supported. Filenames are specified as part of the client bulk validate and load commands.

**Client Bulk Validate/Load Files Must Be in XML Format.** The client load-bulk (page 67) and client validate-bulk (page 81) commands currently only support clients definition input files in XML format. You cannot use a CSV file with these commands.

**Delayed Updates in Graphical Management Console for MCCLI Scheduler Changes.** Suspending or resuming the scheduler using the MCCLI will take effect immediately but Avamar Administrator might not detect the change for approximately 15 seconds.

**Deleting Clients With Backups Pending.** As with the Avamar Administrator, a client cannot be deleted until it is idle without any backups in progress or it is in the wait queue.

**Overall Performance.** MCCLI commands typically require one to two seconds to complete. This elapsed time is primarily due to the fact that the Java Virtual Machine (JVM) must be started in order to run any MCCLI command.

**Performance is Suitable for 5000 or Fewer Clients.** As with the graphical Avamar Administrator application, the MCCLI is suitable for systems comprising a maximum of 5000 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.** Not all errors and failure conditions have specific event codes yet and as a result, the generic event code for a `mccli` command failed will be displayed. For example, missing arguments or failed connection from the MCCLI to the MCS will generate the same 23998 event (`mccli command failed`).
**INSTALLATION AND CONFIGURATION**

This chapter describes how to configure the Avamar Management Console Command Line Interface (MCCLI).

In order to properly install and configure the MCCLI, you must perform the following tasks in the following order:

- **Install the Java Runtime Environment (JRE)** (page 12)
- **Install the MCCLI RPM Package** (page 13)
- **Run the avsetup_mccli Utility** (page 14)

---

**Install the Java Runtime Environment (JRE)**

The MCCLI requires the JRE 1.5.0_12 or higher be installed on the platform running the MCCLI. JRE 1.5.0_12 can be downloaded from the Sun Microsystems web site or from EMC.

1. Obtain either jre-1_5_0_12-linux-i586-rpm.bin or jre-1_5_0_12-linux-i586.rpm.

2. Do one of the following:

<table>
<thead>
<tr>
<th>IF</th>
<th>DO THIS</th>
</tr>
</thead>
</table>
| Installing jre-1_5_0_12-linux-i586-rpm.bin. | Copy the binary to a temporary directory on your system.  
Open a command shell and log into your system as root.  
Change directory to the temporary directory and run the binary by entering:  
jre-1_5_0_12-linux-i586-rpm.bin |
Install the MCCLI RPM Package

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

1. Obtain the dpnmccli-VERSION.rh7.i386.rpm package.
   
   Where VERSION is the specific version of the RPM package you are installing.

2. Open a command shell and log into your system as root.

3. Copy the RPM to a temporary directory.

4. Enter the following and follow the instructions displayed in your command shell:

   rpm -ivh dpnmccli-VERSION.rh7.i386.rpm

   Where VERSION is the specific version of the RPM package you are installing.

The following information appears in your command shell:

   rpm -ivh dpnmccli-VERSION.rh7.i386.rpm

   Preparing... ########################################################## [100%]
   1:dpnmccli ########################################################## [100%]

   Please run /usr/local/avamar/bin/avsetup_mccli to configure MCCLI
Run the avsetup_mccli Utility

avsetup_mccli is a utility that configures the MCCLI.

By default the MCCLI RPM installs essential files to the following locations:

- /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

Refer to your RPM documentation for instructions on installing the MCCLI application to a directory other than the default /usr/local/avamar location.

The following default paths are used by the MCCLI during command invocation:

- /usr/java/jre1.5.0_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 re-running the avsetup_mccli utility.

Additionally, avsetup_mccli also prompts you to enter values for all mccli global options (page 24). Entering these values during your interactive avsetup_mccli session automatically updates the mcclimcs.xml default options file (page 16).

Configure your MCCLI by performing the following:

1. Open a command shell and log into your system as user root.
2. Enter:
   
   avsetup_mccli
   
   The following information appears in your command shell:
   
   setting linux default
   Enter the location of your JRE 1.5 installation [/usr/java/jre1.5.0_12]:
   
3. Enter the directory where your Java Runtime Environment (JRE) is installed and press ENTER.
   
   The following information appears in your command shell:
   
   Enter the root directory of your Avamar installation [/usr/local/avamar]:
   
4. Enter the top-level (root) directory where your Avamar software is installed and press ENTER.
   
   The following information appears in your command shell:
   
   Enter the user data directory of your Avamar installation [~/.avamardata/var]:
   
5. Enter the directory where your user data is stored and press ENTER.
   
   The following information appears in your command shell:
   
   Configuring default local mcsprofile in /usr/local/avamar/lib/mcclimcs.xml
   Enter default mcs host name (mcsaddr) [avamar-1.example.com]:
   
6. Enter the Avamar server IP address or hostname as defined in corporate DNS and press ENTER.
   
   The following information appears in your 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 your command shell:
   ```
   Enter default userid on avamar-1.example.com (mcsuserid) [MCUser]:
   ```
8. Enter an existing Avamar administrative user account and press **ENTER**.
   The following information appears in your command shell:
   ```
   Enter password for MCUser (mcspasswd) [MCUser1]:
   ```
9. Enter the Avamar administrative user account password and press **ENTER**.
   The following information appears in your command shell:
   ```
   Avamar CLI has been configured correctly
   Type mccli command to use it
   ```
Additional Configuration Notes

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

**mccli.xml Preferences File**

The mccli.xml preferences file contains all user-modifiable parameters for the MCCLI application.

The factory 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, the factory 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 will be read and those settings will be used for that command session.

The mccli.xml file currently contains only two user-modifiable parameters:

- event_monitor_display_limit, which sets the maximum number of events to retrieve.
- mcs_config_file, which 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 will be 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 (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. One profile can be designated as the default profile to use if no MCS information is specified on the command line global options (page 24). 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.

**IMPORTANT:** 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 your mcclimcs.xml preferences file to account for those changes.
Behavior. The factory 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, the factory 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:

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

<MCSConfig default="local">
  <!-- Add Resource, Command, Options, Option nodes to match -->
  <Defaults>
    <Commands>
      <!-- the hierarchy in the mcclisyntax.xml file -->
      <!-- use the Value attribute to specify the default value -->
      <!-- Use the Value attribute to specify the default value -->
      <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>
</MCSConfig>
```

```xml
<!-- add more profiles if needed here -->
<!-- and set default to select default -->
```
**Practical Examples.** Consider the following activity resource setting:

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

This setting constrains the `mccli activity show` command (page 28) to only show active jobs, as if the `--active=true` option was supplied on the command line.

Consider the following client resource settings:

```xml
<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 (page 49) so that any new client will be enabled and have its page data port set to 29123 as if the `--enabled=true` and `--pageport=29123` options were supplied on the command line.
MCCLI COMMAND REFERENCE

This chapter documents all commands, features and functions provided by the Avamar Management Console Command Line Interface (MCCLI) application.

General Programming Notes

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 (page 22)
- A command (page 24)
- Global options for connecting to the MCS (page 24)
- One or more command options, which are specific to the resource and command combination used on a specific mccli command line.
- Display options (page 25) that control how the output is formatted.

These command line elements must be specified in this order or the mccli command will fail.
Data Types

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

<table>
<thead>
<tr>
<th>DATA TYPE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BOOLEAN</td>
<td>BOOLEAN values always express a binary condition: true or false. Valid values are the case-insensitive words <code>true</code> or <code>false</code>.</td>
</tr>
<tr>
<td>INTEGER</td>
<td>INTEGER values are always whole numbers.</td>
</tr>
<tr>
<td>STRING</td>
<td>STRING values contain plain text. <strong>IMPORTANT:</strong> 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 wild cards). Refer to <em>Pattern Matching Operators</em> (page 21) for additional information.</td>
</tr>
</tbody>
</table>

Default Values

If a command line argument has a default value, it shown inside parenthesis. For example, this BOOLEAN value defaults to the false condition:

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

Similarly, this 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 or not a particular argument is required for that particular command: optional arguments are enclosed by square brackets; required arguments are not.

For example, consider the following command line synopsis:

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

In the context of this particular 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 Operators

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

<table>
<thead>
<tr>
<th>OPERATOR</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asterisk (*)</td>
<td>Matches zero or more occurrences of any character until the first directory 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 directory. 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>.</td>
</tr>
<tr>
<td>Double asterisk (**)</td>
<td>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> directory structure, no matter how many sub-directory levels are encountered.</td>
</tr>
<tr>
<td>Question mark (?)</td>
<td>Matches one occurrence of any character. Conventional regex behavior.</td>
</tr>
<tr>
<td>Plus sign (+)</td>
<td>Unlike regex, the plus sign is not processed as a glob operator; the plus sign only matches a single occurrence of the plus sign.</td>
</tr>
<tr>
<td>Forward slash (/)</td>
<td>Patterns beginning with forward slash (/) are assumed to be absolute path designations for a single directory; recursive processing of subdirectories is turned off and that directory name is not matched anywhere else.</td>
</tr>
</tbody>
</table>
| [range of values] | Characters enclosed in square brackets and separated by a single hyphen (-) are interpreted as a range of values. This is conventional regex behavior. For example:  
  • [0-9] matches any single numeric character  
  • [a-z] matches any single lowercase alpha character |
| Pound sign (#) | In most cases, you can define multiple matching patterns in a text file and pass that file into the utility. This is generally easier than specifying multiple matches directly on the command line. However, when using a text file to pass in pattern matches, the pound sign (#) is interpreted as a comment if it appears at the beginning of a matching pattern. This will cause that entire pattern matching entry to be ignored. |
Output Description

Successful Completion. When an `mccli` command successfully completes, 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 (zero because the command successfully completed). The second element is the event code (23000 because the command successfully completed). The third element is the event code short description.

Depending on the exact 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 successfully complete, the output message will be in the same format but the return code will be one (1), and the event code will describe the error condition. For example:

1,22288,Dataset does not exist.

Global Command Line Syntax and Behavior

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

Synopsis

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

Resources

Each `mccli` command line must contain one and only one of the following resources:

- **activity**: Activity management resource. Refer to *Activity Resource Overview* (page 27) for additional information about the activity resource.
- **agent**: Client agent management resource. Refer to *Agent Resource Overview* (page 29) for additional information about the agent resource.
- **backup**: Backup management resource. Refer to *Backup Resource Overview* (page 30) for additional information about the backup management resource.
**checkpoint**  Checkpoint management resource. Refer to *Checkpoint Resource Overview* (page 42) for additional information about the checkpoint resource.

**client**  Client management resource. Refer to *Client Resource Overview* (page 47) for additional information about the client resource.

**dataset**  Dataset management resource. Refer to *Dataset Resource Overview* (page 82) for additional information about the dataset resource.

**domain**  Domain management resource. Refer to *Domain Resource Overview* (page 106) for additional information about the domain resource.

**dump**  **IMPORTANT:** The dump resource is strictly reserved for EMC internal use only. Refer to *Dump Resource Overview* (page 111) for additional information about the domain resource.

**event**  Event code information resource. Refer to *Event Resource Overview* (page 113) for additional information about the event resource.

**group**  Group management resource. Refer to *Group Resource Overview* (page 123) for additional information about the group resource.

**help**  Help information resource. Refer to *Help Resource Overview* (page 143) for additional information about the help resource.

**mcs**  Administrator server management resource. Refer to *MCS Resource Overview* (page 144) for additional information about the mcs resource.

**plugin**  Client plugin management resource. Refer to *Plugin Resource Overview* (page 153) for additional information about the plugin resource.

**retention**  Retention policy management resource. Refer to *Retention Resource Overview* (page 154) for additional information about the retention resource.

**schedule**  Schedule management resource. Refer to *Schedule Resource Overview* (page 163) for additional information about the schedule resource.

**server**  Server monitoring resource. Refer to *Server Resource Overview* (page 174) for additional information about the server resource.

**snapup**  Snapup management resource. Refer to *Snapup Resource Overview* (page 181) for additional information about the snapup management resource.

**IMPORTANT:** The snapup management resource has been deprecated in favor of the backup management resource (page 30).
Commands

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

Global Options

Each mccli command line can contain one or more of the following options, which are global in nature (they can be used with any resource class or command).

Typically, global options are persistently stored and read from the mccli.xml preferences file (page 16) and are only supplied on an actual mccli command line in cases where the persistent settings must be temporarily overridden.

---

### --mcsprofile=STRING

Specifies MCS profile name. If this option is supplied, settings stored in this profile are used and other global options supplied on the command line are ignored.

### --mcsaddr=STRING

Specifies MCS network name or IP address.

### --mcspasswd=STRING

Specifies --mcsuserid account password.

### --mcsport=INTEGER

Specifies data port used to contact MCS.

### --mcsuserid=STRING

Specifies Avamar user account that will be used to run the mccli command.

Command Options

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

--normalize

Produces output in a format that better supports parsing and comparison. This option affects any command that returns a date or time, or filesystem information such as capacity or file sizes. Refer to Default and Normalized Output. (page 25) for additional information.

--xml

Formats output as XML. This is useful for parsing output as part of a script.

Default and Normalized Output. The following table shows examples of various returned values expressed in both default and normalized formats:

<table>
<thead>
<tr>
<th>VALUE</th>
<th>DEFAULT FORMAT</th>
<th>NORMALIZED FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Absolute time</td>
<td>Expressed as conventional date/timestamp with local time zone. For example: 2008-09-12 11:00:17 PDT</td>
<td>Expressed as an integer representing Unix precision time format numerical (UTC milliseconds from the epoch). For example: 1189620017000</td>
</tr>
<tr>
<td>Elapsed time</td>
<td>Expressed as number of days, hours, minutes and seconds that have elapsed. For example: 12 days 17h:30m</td>
<td>Expressed as an integer representing total number of milliseconds that have elapsed. For example: 1099814000</td>
</tr>
<tr>
<td>Capacity</td>
<td>Expressed in bytes, MB, GB or TB, whichever is most correct. For example: 1.3 TB</td>
<td>Expressed as an integer representing total number of bytes. For example: 1464957140992</td>
</tr>
</tbody>
</table>

Return Codes

Each mccli command will return one of the following:

0    Command succeeded.

1    Command failed.
Global Event Codes

These event codes are global (they can potentially 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 respective `mccli` command.

22601  Server inactive.
23000  CLI command successfully completed.
23001  Arguments required by CLI command are either missing or empty.
23993  Attempt to read or write a file has failed.
23995  Invalid option specified on the CLI.
23996  Failed to connect to the administrator server.
23997  Conflicting arguments specified on the command line.
23998  CLI command failed.
23999  Unexpected command failure.
Activity Resource Overview

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

- `activity cancel` (page 27)
- `activity show` (page 28)

activity cancel

The `mccli activity cancel` command terminates (cancels) the specified backup, restore or validation activity.

If the specified activity has already completed, the command returns an event code indicating that the activity ID is invalid along with information indicating that the job has already completed.

Synopsis

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

Global Options

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

`--id=STRING` Specifies which activity to terminate (cancel). STRING must be a valid activity ID. This argument is required.

**TIP:** Use `mccli activity show` (page 28) to return a list of valid activity IDs, or use the activity ID returned by the backup, restore or validation command.

Display Options

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

Event Codes

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

The mccli activity show command lists backup, restore and validation activities with summary information, or detailed information for a specific activity. If viewing summary information for multiple activities, returned information can be filtered on an individual domain or client basis.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--active=Boolean(false) If set true, only currently running activities are returned.

--completed=Boolean(false) If set true, only completed activities are returned.

--domain=STRING Specifies Avamar server domain containing the client specified by the --name argument. If this option is supplied and --name is not supplied, all activities within that domain are shown.

--name=STRING Specifies client for which activities should be shown.

IMPORTANT: If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

--queued=Boolean(false) If set true, only queued activities are returned.

--verbose=Boolean(false) If set true, detailed (verbose) activity information is returned. If set false or not supplied, summary information is returned.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.
Agent Resource Overview

The mccli agent resource comprises a single command: agent show, which shows summary properties for all client agents.

agent show

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

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.
Backup Resource Overview

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

- `backup delete` (page 31)
- `backup edit` (page 33)
- `backup restore` (page 35)
- `backup show` (page 37)
- `backup validate` (page 40)
backup delete

The `mccli backup delete` command permanently removes (deletes) the specified backup from the server.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

`--created=STRING` Specifies date backup was created. STRING must be in the format of YYYY-MM-DD, where YYYY, MM and DD are calendar year, month and date, respectively.

`[--domain=STRING(/)]` Specifies Avamar server domain containing the client specified by the `--name` argument.

`--force=Boolean(false)` By default, if you attempt to delete a backup that has more than one retention type assigned to it, 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 set true, this checking is disabled and the backup will be deleted regardless of the number of retention types assigned to it.

`--labelNum=INTEGER` Specifies label number of backup to delete.

`--name=STRING` Specifies client from which this backup was originally taken. This argument is required. **IMPORTANT:** If a fully-qualified client name (for example, `/clients/MyClient`) is supplied, the `--domain` argument is ignored.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
### Event Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22236</td>
<td>Client does not exist.</td>
</tr>
<tr>
<td>22552</td>
<td>Backup does not exist.</td>
</tr>
<tr>
<td>22556</td>
<td>Changed backup retention.</td>
</tr>
<tr>
<td>22558</td>
<td>Multiple retention tags exist.</td>
</tr>
</tbody>
</table>
backup edit

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

- Directly specify a new expiration date
- Add additional days (d), weeks (w), months (m) or years that should be added to the existing backup expiration date
- Specify that the backup should never expire

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--created=STRING Specifies date backup was created. STRING must be in the format of YYYY-MM-DD, where YYYY, MM and DD are calendar year, month and date, respectively.

[--domain=STRING(/)] Specifies Avamar server domain containing the client specified by the --name argument.

--expiration=STRING Specifies new expiration date. STRING must be in one of the following formats:

    YYYY-MM-DD | +nn{D|W|M|Y} | NO_EXPIRATION

Where:

- YYYY-MM-DD is a specific year (YYYY), month (MM) and day (DD)
- +nn{D|W|M|Y} is the number (nn) of additional days (d), weeks (w), months (m) or years that should be added to the existing backup expiration date
- NO_EXPIRATION specifies that the backup should never expire.

--expiration and --retention are mutually exclusive.
--force=Boolean(false)  By default, if you attempt to edit 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 modified. If set true, this checking is disabled and the backup will be modified regardless of the number of retention types assigned to it.

--labelNum=INTEGER  Specifies label number of backup to change.

--name=STRING  Specifies client from which this backup was originally taken. This argument is required. **IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

--retention=STRING  Specifies one or more retention types that should be assigned to this backup. STRING must be one or more of the following:
- D or daily
- W or weekly
- M or monthly
- Y or yearly
- N or none
Both short form and long form retention type values are allowed and can be mixed. For example, all of the following are valid:
  --retention=D,weekly
  --retention=none
  --retention=Daily,W,monthly
--expiration and --retention are mutually exclusive.

Display Options
Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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.
backup restore

The mccli backup restore command restores files and folders (directories) to the specified client.

TIP: This command returns an activity ID, which can be passed to mccli activity show (page 28) in order to get status for this restore activity.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--cmd=STRING
   Specifies one or more optional plug-in commands. Multiple --cmd arguments can be supplied but each argument can only specify one plug-in command.
   For example, this is valid --cmd syntax:
   --cmd="verbose=5" --cmd="throttle=5"
   However, this is not valid --cmd syntax:
   --cmd="verbose=5 throttle=5"
   Refer to your Avamar System Administration Guide for a list of valid plug-in commands that can specified during a restore operation.

--data=STRING
   Specifies a single optional target directory (folder) or file to restore.
   IMPORTANT: Each --data option can only specify one target directory (folder) or file to restore; use multiple --data options to restore multiple targets.

--dest-client-domain=STRING
   Specifies Avamar server domain containing the alternative client specified by the --dest-client-name argument.
--dest-client-name=STRING  Used when performing a redirected restore to specify alternative destination client. You must also supply --dest-dir on any command line using this option.

--dest-dir=STRING  Specifies optional destination directory for the restored files or folders (directories).

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

--labelNum=INTEGER  Specifies which backup to use for this restore operation. This argument is required.  
**TIP:** Use mccli backup show (page 187) to return a list of valid backup label numbers.

--name=STRING  Specifies target client for this restore operation. STRING must be a valid client name. This argument is required.  
**TIP:** Use mccli client show (page 70) to return a list of valid client names.  
**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain option is ignored.

--plugin=INTEGER  Specifies plug-in ID for this restore operation. This argument is required.  
**TIP:** Use plugin show (page 153) to return a list of valid plug-in IDs.  
**IMPORTANT:** When performing a redirected restore, this argument should specify a plugin that is compatible with the alternative destination client, not the plugin that was originally used to perform the backup.

**Display Options**
Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

**Event Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>22236</td>
<td>Client does not exist.</td>
</tr>
<tr>
<td>22297</td>
<td>Request to restore is rejected.</td>
</tr>
<tr>
<td>22312</td>
<td>Client restore scheduled.</td>
</tr>
<tr>
<td>23009</td>
<td>Invalid plugin specified on the CLI.</td>
</tr>
</tbody>
</table>
backup show

The mccli backup show command returns all backups currently stored on the Avamar server for the specified client.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--after=STRING
Used with --before to specify a range of backups dates to return.
STRING must be in the format of YYYY-MM-DD, where YYYY is the four-digit year, MM is the two-digit calendar month and DD is the two-digit day of the month.

--before=STRING
Used with --after to specify a range of backups dates to return.
STRING must be in the format of YYYY-MM-DD, where YYYY is the four-digit year, MM is the two-digit calendar month and DD is the two-digit day of the month.

--dir=STRING(/)
Specifies initial directory of the backup from which to begin listing files and folders (directories).

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

--labelNum=INTEGER
Specifies a particular backup for which to show information.

--name=STRING
Specifies client from which to show backups. This argument is required.
IMPORTANT: If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

--recursive=BOOLEAN(false)
If set true, backup files and folders (directories) are recursively shown.
**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

**Event Codes**

22236 Client does not exist.

22504 Failed to retrieve the backups for a client.
Examples

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

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

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

```
Created         LabelNum Size
----------------------- -------- ----------
2006-01-31 15:51:30 PST  2  4767841280
2006-01-31 15:00:23 PST  1  4750878720
```

This command returns verbose file and directory information for a specific parent directory (C:/) within a specific client backup (the second backup taken from MyClient.example.com):

```
mccli backup show --name=clients/MyClient.example.com --verbose --labelNum=2 --dir=C:/
```

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

```
Type Date          Size          Owner                Group           Permissions     Name
---- ------------------- ------------- ------------------ --------------- ------------------ ---------------
Dir   2005-12-12 08:59:27 205,526,604 Administrators unknown  drwxrwxr-x     C:/Documents and Settings/
Dir   2006-01-17 11:33:54 1,601,595,257 Administrators unknown  dr-xr-x---     C:/Program Files/
Dir   2005-12-12 08:59:55 1,841 Administrator None          drwx-------    C:/RECYCLER/
Dir   2006-01-31 15:45:30 20,480 Administrators unknown     d-----x------   C:/System Volume Information/
Dir   2006-01-17 15:18:22 211 Administrators unknown        drwxrw---      C:/tmp/
Dir   2006-01-18 09:14:25 1,600,258,500 Administrators unknown  drwxrw---      C:/WINDOWS/
File  2005-12-09 15:21:15 211 Administrators unknown        -r-xr-x---     C:/boot.ini
File  2005-12-12 08:56:49 4,128 Administrators unknown        -rwxrwx---     C:/INFCACHE.1
File  2005-08-11 15:15:00 0 Administrators unknown          -rwxrw---      C:/IO.SYS
File  2005-08-11 15:15:00 0 Administrators unknown          -rwxrw---      C:/MSDOS.SYS
File  2005-08-04 03:00:00 47,564 Administrators unknown      -r-xr-x---     C:/NTDETECT.COM
File  2005-08-04 03:00:00 250,032 Administrators unknown     -r-xr-x---     C:/ntldr
```
backup validate

The `mccli backup validate` command initiates a validation of the specified backup.

**TIP:** This command returns an activity ID, which can be passed to `mccli activity show` (page 28) in order to get status for this validate activity.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--cmd=STRING
```

Specifies one or more optional plug-in commands. Multiple `--cmd` arguments can be supplied but each argument can only specify one plug-in command.

For example, this is valid `--cmd` syntax:

```
--cmd="verbose=5" --cmd="throttle=5"
```

However, this is not valid `--cmd` syntax:

```
--cmd="verbose=5 throttle=5"
```

Refer to your Avamar System Administration Guide for a list of valid plug-in commands that can specified during a validate operation.

```
--dest-client-domain=STRING
```

Used with `--dest-client-name` to validate a replicated backup (that is, a backup for a client residing within the `/REPLICATE` domain) on alternate client.

Specifies Avamar server domain containing the client specified by the `--dest-client-name` argument.
--dest-client-name=STRING

Used with --dest-client-domain to validate a replicated backup (that is, a backup for a client residing within the /REPLICATE domain) on alternate client. Specifies alternative target client for the validate operation. STRING must be a valid client name.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --dest-client-domain argument is ignored.

**IMPORTANT:** This argument will return an error if you supply anything other than a valid client name residing within the /REPLICATE domain.

--domain=STRING(/)

Specifies Avamar server domain containing the client specified by the --name argument.

--labelNum=INTEGER

Specifies which backup to use for this validate operation. This argument is required.

--name=STRING

Specifies target client for the validate operation. STRING must be a valid client name. This argument is required.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

--plugin=INTEGER

Specifies plug-in ID for this restore operation. This argument is required.

**TIP:** Use `plugin show` (page 153) to return a list of valid plug-in IDs.

Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22236</td>
<td>Client does not exist.</td>
</tr>
<tr>
<td>22298</td>
<td>Request to validate is rejected.</td>
</tr>
<tr>
<td>22315</td>
<td>Client validate scheduled.</td>
</tr>
<tr>
<td>23009</td>
<td>Invalid plugin specified on the CLI.</td>
</tr>
</tbody>
</table>
Checkpoint Resource Overview

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

- `checkpoint cancel-validate` (page 42)
- `checkpoint create` (page 43)
- `checkpoint delete` (page 44)
- `checkpoint show` (page 45)
- `checkpoint validate` (page 46)

checkpoint cancel-validate

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

Synopsis

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

Global Options

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Display Options

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

Event Codes

- 22615  A checkpoint validation was cancelled.
- 22616  No checkpoint validation is running.
checkpoint create

The mccli checkpoint create command creates a checkpoint.

NOTE: An MCS flush is performed as part of this checkpoint.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22608 An Avamar server checkpoint was successfully created.
22609 An error occurred creating an Avamar server checkpoint.
The `mccli checkpoint delete` command permanently removes (deletes) a checkpoint from the Avamar server.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--cptag=STRING   Specifies which checkpoint to delete. STRING must be a valid checkpoint ID. This argument is required.

TIP: Use `mccli checkpoint show` (page 45) to return a list of valid checkpoint IDs.
```

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**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 (verbose) information.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

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

If set true, detailed (verbose) checkpoint information is returned.

If set false or not supplied, summary information is returned.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

```
22531 Unexpected exception occurred.
```
checkpoint validate

The mccli checkpoint validate command validates a checkpoint.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--checktype=TYPE    Constrains checkpoint validation to one or more of the following checkpoint TYPES:

   full          Validation performed all checks.
   rolling       Validation was rolling HFS check.

--cptag=STRING     Specifies which checkpoint to validate. STRING must be a valid checkpoint ID. This argument is required.

TIP: Use mccli checkpoint show (page 45) to return a list of valid checkpoint IDs.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22612  Starting to validate an Avamar server checkpoint.

22617  The specified checkpoint was not found.
Client Resource Overview

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

- `client activate` (page 48)
- `client add` (page 49)
- `client backup-dataset` (page 54)
- `client backup-group-dataset` (page 56)
- `client backup-target` (page 58)
- `client delete` (page 60)
- `client edit` (page 61)
- `client invite` (page 66)
- `client load-bulk` (page 67)
- `client move` (page 68)
- `client retire` (page 69)
- `client show` (page 70)
- `client show-plugins` (page 73)
- `client snapup-dataset` (page 74)
- `client snapup-group-dataset` (page 76)
- `client snapup-target` (page 79)
- `client validate-bulk` (page 81)

**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 `dataset show` (page 103), `retention show` (page 161) and `plugin show` (page 153) commands, respectively.

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

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

**IMPORTANT:** The `mccli client activate` command has been deprecated in favor of `mccli client invite` (page 66).

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--domain=STRING(/)  Specifies domain containing client specified by the
  --name argument.

--name=STRING       Specifies which client to activate. STRING must be a
  valid client name. This argument is required.

TIP: Use `mccli client show` (page 70) to return a list
  of valid client names.

IMPORTANT: If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain
  argument is ignored.
```

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**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 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 enables policies to be subsequently defined for it even if the client has not yet been activated.

Synopsis

mccli client add [GLOBAL-OPTIONS] [--contact=STRING]
[--dataset=STRING] [--dataset-domain=STRING(/)] --domain=STRING(/)
[--email=STRING] [--enabled=BOOLEAN(false)]
[--encryption=STRING(None)] [--location=STRING] --name=STRING
[--override-dataset=BOOLEAN] [--override-encryption=BOOLEAN]
[--override-retention=BOOLEAN] [--overtime=BOOLEAN]
[--overtime-option=STRING] [--pageable=BOOLEAN]
[--pageaddr=STRING] [--pageport=STRING][--phone=STRING]
[--retention=STRING] [--retention-domain=STRING(/)]
[DISPLAY-OPTIONS]

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--contact=STRING.Specifies contact information for the new client.

--dataset=STRING.Specifies dataset that this client will use when performing on-demand backups and restores, or when the group dataset is overridden.

--dataset-domain=STRING(/). Specifies Avamar server domain containing the dataset specified by the --dataset argument.

--domain=STRING(/). Specifies domain where new client will reside.

--email=STRING.Specifies primary contact (responsible party) email address.

--enabled=BOOLEAN(false). If set true, client is eligible to immediately participate in on-demand and group backup and restore activities. If set false or not supplied, the client is disabled.
--encryption=STRING  Specifies encryption method that this client will use when performing on-demand backups and restores, or when the group encryption method is overridden.

STRING must be one of the following:

<table>
<thead>
<tr>
<th>Encryption Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>High</strong></td>
<td>Strongest available encryption setting for that specific client platform.</td>
</tr>
<tr>
<td><strong>Medium</strong></td>
<td>Medium strength encryption.</td>
</tr>
<tr>
<td><strong>None</strong></td>
<td>No encryption.</td>
</tr>
<tr>
<td><strong>AES 128-bit</strong></td>
<td>This setting deprecated in version 4.1.</td>
</tr>
<tr>
<td><strong>Axion</strong></td>
<td>This setting deprecated in version 4.1.</td>
</tr>
</tbody>
</table>

**NOTE:** The exact encryption technology and bit strength used for any given client/server connection is dependent on a number of factors, including the client platform and Avamar server version. Refer to your [Avamar Product Security Guide](#) for additional information.

--location=STRING  Specifies location information for the new client.

--name=STRING  Specifies new client name. This argument is required.

**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

--override-dataset=BOOLEAN  If set true, use the client dataset instead of group dataset for scheduled backups.

--override-encryption=BOOLEAN  If set true, use the encryption method specified by --encryption instead of group encryption method for scheduled backups.

--override-retention=BOOLEAN  If set true, use the retention policy specified by --retention instead of group retention policy for scheduled backups.

--overtime=BOOLEAN  If set true, client can exceed its backup window during scheduled backups.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--overtime-option=STRING</td>
<td>Specifies scheduled group backup overtime behavior. STRING must be one of the following:</td>
</tr>
<tr>
<td></td>
<td><strong>ALWAYS</strong>: Scheduled group backups are always allowed to run past the schedule duration setting.</td>
</tr>
<tr>
<td></td>
<td><strong>NEVER</strong>: Scheduled group backups are never allowed to run past the schedule duration setting.</td>
</tr>
<tr>
<td></td>
<td><strong>NEXT</strong>: Only the next scheduled group backup is allowed to run past the schedule duration setting.</td>
</tr>
<tr>
<td></td>
<td><strong>NEXT_SUCCESS</strong>: Scheduled group backups are allowed to run past the schedule duration setting until a successful backup is completed.</td>
</tr>
<tr>
<td></td>
<td>Default is <strong>NEXT_SUCCESS</strong>.</td>
</tr>
<tr>
<td>--pageable=BOOLEAN</td>
<td>If set true, the client can be paged for the purpose of initiating activation or picking up new backup/restore work.</td>
</tr>
<tr>
<td>--pageaddr=STRING</td>
<td>Specifies IP address that Avamar server can use to contact this client.</td>
</tr>
<tr>
<td>--pageport=STRING</td>
<td>Specifies data port that Avamar server can use to contact this client.</td>
</tr>
<tr>
<td>--phone=STRING</td>
<td>Specifies primary contact (responsible party) voice telephone information.</td>
</tr>
<tr>
<td>--retention=STRING</td>
<td>Specifies retention policy that this client will use when performing on-demand backups and restores, or when the group retention policy is overridden.</td>
</tr>
<tr>
<td>--retention-domain=STRING(/)</td>
<td>Specifies Avamar server domain containing the retention policy specified by the --retention argument.</td>
</tr>
</tbody>
</table>
Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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.

Notes

If --overtime-option is not supplied and --overtime=true is supplied, then --overtime-option is set to NEXT_SUCCESS.

If --overtime-option is not supplied and --overtime=false is supplied, then --overtime-option is set to NEVER.

If both --overtime-option and --overtime are supplied, then --overtime-option takes precedence.

Examples

This command adds a new client (MyClient):

```
mccli client add --name=MyClient
```

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

```
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>action</td>
<td>add</td>
</tr>
<tr>
<td>domain</td>
<td>/</td>
</tr>
<tr>
<td>node</td>
<td>MyClient</td>
</tr>
<tr>
<td>clientid</td>
<td>79a1042d7f5158c660fb7b863281f9787f8cb942</td>
</tr>
</tbody>
</table>

This command adds a new client (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>
  </Data>
</CLIOutput>
```
<Attribute>domain</Attribute><Value>/</Value></Row><Row><Attribute>node</Attribute><Value>MyClient</Value></Row><Row><Attribute>clientid</Attribute><Value>70279698b5f755b1bc2f41432b3b3471048478e7</Value></Row></Data></CLIOutput>
The `mccli client backup-dataset` command initiates an on-demand backup of a single client using the specific dataset. If a dataset is not specified on the command line, then the default dataset associated with that client is used.

**TIP:** This command returns an activity ID, which can be passed to `mccli activity show` (page 28) in order to get status for this backup activity.

### Synopsis

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

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

- **--cmd=STRING**
  Specifies one or more optional plug-in commands. Multiple `--cmd` arguments can be supplied but each argument can only specify one plug-in command.
  For example, this is valid `--cmd` syntax:
  ```
  --cmd="verbose=5" --cmd="throttle=5"
  ```
  However, this is not valid `--cmd` syntax:
  ```
  --cmd="verbose=5 throttle=5"
  ```
  Refer to your *Avamar System Administration Guide* for a list of valid plug-in commands that can specified during a backup operation.

- **--dataset=STRING**
  Specifies dataset to be used for this backup operation. If this argument is not supplied, the default dataset associated with that client is used.
  **TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.

- **--dataset-domain=STRING(/)**
  Specifies Avamar server domain containing the dataset specified by the `--dataset` argument.

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

Specifies which client is backed up (backed up). STRING must be a valid client name. This argument is required.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

**IMPORTANT:** If a fully-qualified client name (for example, `/clients/MyClient`) is supplied, the `--domain` argument is ignored.

Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

Event Codes

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

22236 Client does not exist.

22253 Client Adhoc Backup Request Error - Exception.

22305 Client backup scheduled.

22309 Invalid plugin specified on the CLI.

23003 Invalid dataset name specified on the CLI.
client backup-group-dataset

The `mccli client backup-group-dataset` command initiates an on-demand backup of a single client using the group dataset.

**NOTE:** It must be understood that this command initiates an on-demand client backup, not an on-demand group backup. Therefore, even if this client is a member of a group that is disabled and even if that group’s dataset is specified, the backup will still occur.

**IMPORTANT:** The client must already be a member of this group or the backup will fail.

**TIP:** This command returns an activity ID, which can be passed to `mccli activity show` (page 28) in order to get status for this backup activity.

### Synopsis

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

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

**--cmd=STRING**

Specifies one or more optional plug-in commands. Multiple **--cmd** arguments can be supplied but each argument can only specify one plug-in command. For example, this is valid **--cmd** syntax:

```
--cmd="verbose=5" --cmd="throttle=5"
```

However, this is not valid **--cmd** syntax:

```
--cmd="verbose=5 throttle=5"
```

Refer to your *Avamar System Administration Guide* for a list of valid plug-in commands that can specified during a backup operation.

**--domain=STRING(/)**

Specifies Avamar server domain containing the client specified by the **--name** argument.
--group-domain=STRING(/) Specifies Avamar server domain containing the group specified by the --group-name argument.

--group-name=STRING Specifies which group dataset to use for this backup. STRING must be a valid group name. This argument is required.

**TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

**IMPORTANT:** If a fully-qualified group name (for example, `/MyDomain/MyGroup`) is supplied, the --group-domain argument is ignored.

--name=STRING Specifies which client is backed up (backed up). STRING must be a valid client name. This argument is required.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

**IMPORTANT:** If a fully-qualified client name (for example, `/clients/MyClient`) is supplied, the --domain argument is ignored.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

### 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 one or more files or folders (directories) on the specified client.

**TIP:** This command returns an activity ID, which can be passed to mccli activity show (page 28) in order to get status for this backup activity.

### Synopsis

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

### Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

### Command Options

**--cmd=STRING**  
Specifies one ore more optional plug-in commands. Multiple --cmd arguments can be supplied but each argument can only specify one plug-in command. For example, this is valid --cmd syntax:
```
--cmd="verbose=5" --cmd="throttle=5"
```
However, this is not valid --cmd syntax:
```
--cmd="verbose=5 throttle=5"
```
Refer to your Avamar System Administration Guide for a list of valid plug-in commands that can specified during a backup operation.

**--domain=STRING(/)**  
Specifies Avamar server domain containing the client specified by the --name argument.

**--name=STRING**  
Specifies which client is backed up (backed up). STRING must be a valid client name. This argument is required.

**TIP:** Use mccli client show (page 70) to return a list of valid client names.

**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

**--plugin=INTEGER**  
Specifies plug-in ID.

**TIP:** Use plugin show (page 153) to return a list of valid plug-in IDs.
--target=STRING Specifies which files or folders (directories) to include in the backup.

At least one --target=STRING argument is required and more than one --target=STRING can be supplied on the same command line.

This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to Pattern Matching Operators (page 21) for additional information.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

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

22236 Client does not exist.

22253 Client Adhoc Backup Request Error - Exception.

22305 Client backup scheduled.

22309 Invalid plugin specified on the CLI.
client delete

The `mccli client delete` command permanently removes (deletes) a client and all its backups from the Avamar server.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--domain=STRING(/)  Specifies domain containing client specified by the
         --name argument.

--name=STRING     Specifies which client to delete. STRING must be a valid
         client name. This argument is required.
         **TIP:** Use `mccli client show` (page 70) to return a list
         of valid client names.
         **IMPORTANT:** If a fully-qualified client name (for
         example, /clients/MyClient) is supplied, the --domain
         argument is ignored.
```

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

- 22212  Client deleted.
- 22236  Client does not exist.
- 22240  Client delete failed.
The `mccli client edit` command modifies (edits) one or more properties for a specific client.

**IMPORTANT:** There are no default settings for the `mccli client edit` command. Entering this command without explicitly supplying options and values on the command line will have no effect on the specified resource.

### Synopsis

```
```

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

- **--activated=BOOLEAN**
  - If set true, client is assumed to have been previously activated.

- **--allow-cis=BOOLEAN**
  - If set true, users are allowed to initiate a backup of their client.

- **--allow-cis-fileselection=BOOLEAN**
  - If set true, users are allowed to make file selections when they initiate a backup of their client.

- **--contact=STRING**
  - Specifies contact information for client.

- **--dataset=STRING**
  - Specifies dataset that this client will use when performing on-demand backups and restores, or when the group dataset is overridden.

- **--dataset-domain=STRING(/)**
  - Specifies Avamar server domain containing the dataset specified by the `--dataset` argument.
--domain=STRING  Specifies domain containing client specified by the --name argument.

--email=STRING  Specifies primary contact (responsible party) email address.

--enabled=BOOLEAN  If set true, client is eligible to immediately participate in on-demand and group backup and restore activities. If set false, the client is disabled.

--encryption=STRING  Specifies encryption method that this client will use when performing on-demand backups and restores, or when the group encryption method is overridden. STRING must be one of the following:

<table>
<thead>
<tr>
<th>Encryption Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium</td>
<td>Medium strength encryption.</td>
</tr>
<tr>
<td>High</td>
<td>Strongest available encryption setting for that specific client platform.</td>
</tr>
<tr>
<td>None</td>
<td>No encryption.</td>
</tr>
<tr>
<td>AES 128-bit</td>
<td>This setting deprecated in version 4.1.</td>
</tr>
<tr>
<td>Axion</td>
<td>This setting deprecated in version 4.1.</td>
</tr>
</tbody>
</table>

**NOTE:** The exact encryption technology and bit strength used for any given client/server connection is dependent on a number of factors, including the client platform and Avamar server version. Refer to your Avamar Product Security Guide for additional information.

--location=STRING  Specifies location information for client.
**--name=STRING**

Specifies which client to modify (edit). STRING must be a valid client name. This argument is required.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the `--domain` argument is ignored.

**--new-name=STRING**

 Specifies new client name.

**--override-cis-retention=BOOLEAN**

If set true, this retention policy is used for all user-initiated backups of this client.

**--override-dataset=BOOLEAN**

If set true, use the client dataset instead of group dataset for scheduled backups.

**--override-encryption=BOOLEAN**

If set true, use the client encryption method instead of group encryption method for scheduled backups.

**--override-retention=BOOLEAN**

If set true, use the client retention policy instead of group retention policy for scheduled backups.

**--overtime=BOOLEAN**

If set true, client can exceed its backup window during backup activities.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--overtime-option=STRING</td>
<td>Specifies scheduled group backup overtime behavior.</td>
</tr>
<tr>
<td></td>
<td>STRING must be one of the following:</td>
</tr>
<tr>
<td>ALWAYS</td>
<td>Scheduled group backups are always allowed to run past the schedule duration setting.</td>
</tr>
<tr>
<td>NEVER</td>
<td>Scheduled group backups are never allowed to run past the schedule duration setting.</td>
</tr>
<tr>
<td>NEXT</td>
<td>Only the next scheduled group backup is allowed to run past the schedule duration setting.</td>
</tr>
<tr>
<td>NEXT_SUCCESS</td>
<td>Scheduled group backups are allowed to run past the schedule duration setting until a successful backup is completed.</td>
</tr>
<tr>
<td></td>
<td>Default is NEXT_SUCCESS.</td>
</tr>
<tr>
<td>--pageable=BOOLEAN</td>
<td>If set true, the client can be paged for the purpose of initiating activation or picking up new backup/restore work.</td>
</tr>
<tr>
<td>--phone=STRING</td>
<td>Specifies primary contact (responsible party) voice telephone information.</td>
</tr>
<tr>
<td>--retention=STRING</td>
<td>Specifies retention policy that this client will use when performing on-demand backups and restores, or when the group retention policy is overridden.</td>
</tr>
<tr>
<td>--retention-domain=STRING(/)</td>
<td>Specifies Avamar server domain containing the retention policy specified by the --retention argument.</td>
</tr>
</tbody>
</table>

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.
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.

Notes

If **--overtime** is not supplied and **--overtime=true** is supplied, then **--overtime** is set to NEXT_SUCCESS.

If **--overtime** is not supplied and **--overtime=false** is supplied, then **--overtime** is set to NEVER.

If both **--overtime** and **--overtime** are supplied, then **--overtime** takes precedence.
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 that client.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--domain=STRING(\/)  Specifies domain containing client specified by the
   --name argument.

--name=STRING  Specifies which client to invite. STRING must be a valid
   client name. This argument is required.

TIP: Use `mccli client show` (page 70) to return a list of valid client names.

IMPORTANT: If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.
```

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**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 load-bulk

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

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--file=STRING
```

Specifies full path and filename of the clients definition input file. This file must be in XML format. This argument is required. Refer to your *Avamar System Administration Guide* for additional information about clients definition files.

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

If set true, a list of clients and their load status is output to stdout.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

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

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

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/)  Specifies domain containing client specified by the --name argument.

--name=STRING  Specifies which client to move. STRING must be a valid client name. This argument is required.
TIP: Use mccli client show (page 70) to return a list of valid client names.
IMPORTANT: If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

--new-domain=STRING  Specifies the new domain for this client. This argument is required.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22515  Client was moved.
22236  Client does not exist.
22501  Failed to move client.
23019  Client move failed - still active.
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.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- **--domain=STRING(/)** Specifies domain containing client specified by the `--name` argument.
- **--expiration=STRING** Specifies expiration of backups for retired client. STRING must be one of the following:
  - `YYYY-MM-DD` to specify an explicit date
  - `+nn{D|W|M|Y}` to specify a duration from today (for example, `+4W` specifies 4 weeks from today)
  - `NO_EXPIRATION`
- **--name=STRING** Specifies which client to retire. STRING must be a valid client name. This argument is required.
  
  **TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

  **IMPORTANT:** If a fully-qualified client name (for example, `/clients/MyClient`) is supplied, the `--domain` argument is ignored.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22236</td>
<td>Client does not exist.</td>
</tr>
<tr>
<td>22506</td>
<td>Failed to retire a client.</td>
</tr>
<tr>
<td>22512</td>
<td>Client was retired.</td>
</tr>
</tbody>
</table>
client show

The mccli client show command lists one or more clients and their properties.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/)  Specifies domain containing client specified by the --name argument.
      NOTE: If --retired is supplied, this argument is ignored.

--name=STRING  Specifies a client name. If not supplied, all clients with the specified domain are listed.
      IMPORTANT: If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

--recursive=BOOLEAN(false)  If set true, command shows all clients in the domain and any sub-domains specified by the --domain=STRING argument.
      If set false or not supplied, command only shows clients within the specified domain (sub-domains are not examined).

--replicated=Boolean(false)  If set true, clients in the REPLICATE domain are shown.
      If set false or not supplied, clients in the REPLICATE domain are not shown.

--retired=BOOLEAN(false)  If set true, clients listed are retired clients.
      NOTE: If supplied, the --domain=STRING argument is ignored.
--verbose=BOOLEAN(false)

If --name is not supplied and this option is set false or not supplied, only the domain and client name are returned for each client. If set true, other properties are also returned.

**NOTE:** This option is only meaningful when the --name option is not supplied. When the --name option is supplied, the details for the client are always returned.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

### Event Codes

#### 22236

Client does not exist.

### Examples

This command returns a simple list of all clients within the clients domain:

```shell
mccli client show --domain=clients
```

This command return a detailed list of properties for MyClient.example.com:

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>MyClient</td>
</tr>
<tr>
<td>Fully Qualified Name</td>
<td>/clients/MyClient.example.com</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows XP Professional SP 1.0</td>
</tr>
<tr>
<td>Paging</td>
<td>Yes</td>
</tr>
<tr>
<td>Page Address</td>
<td>MyClient.example.com</td>
</tr>
<tr>
<td>Page Port</td>
<td>28002</td>
</tr>
<tr>
<td>Page Address Locked</td>
<td>No</td>
</tr>
<tr>
<td>Agent Version</td>
<td>3.5.0-95</td>
</tr>
<tr>
<td>CID</td>
<td>6e6241ae236c2ab58205f413f24d3d6e0</td>
</tr>
<tr>
<td>CID Assigned</td>
<td>2005-10-13 00:00:00 PDT</td>
</tr>
<tr>
<td>Activated</td>
<td>Yes</td>
</tr>
<tr>
<td>Activated Date</td>
<td>2005-10-13 12:28:13 PDT</td>
</tr>
<tr>
<td>Disabled</td>
<td>No</td>
</tr>
<tr>
<td>Agent Last Started</td>
<td>2005-11-02 06:00:00 PST</td>
</tr>
<tr>
<td>Last Check-in</td>
<td>2005-11-02 11:00:59 PST</td>
</tr>
<tr>
<td>Override Group Dataset</td>
<td>No</td>
</tr>
<tr>
<td>Dataset Name</td>
<td>Default Dataset</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Override Group Retention</td>
<td>No</td>
</tr>
<tr>
<td>Retention Policy</td>
<td>Default Retention</td>
</tr>
<tr>
<td>Allow Overtime</td>
<td>No</td>
</tr>
<tr>
<td>Restore Only</td>
<td>No</td>
</tr>
<tr>
<td>Encryption Method</td>
<td>High</td>
</tr>
<tr>
<td>Override Encryption</td>
<td>No</td>
</tr>
<tr>
<td>Allow client initiated backups</td>
<td>Yes</td>
</tr>
<tr>
<td>Allow file selection in client initiated backups</td>
<td>Yes</td>
</tr>
<tr>
<td>Override retention policy on client initiated backups</td>
<td>No</td>
</tr>
<tr>
<td>Contact Name</td>
<td>N/A</td>
</tr>
<tr>
<td>Contact Phone</td>
<td>N/A</td>
</tr>
<tr>
<td>Contact Email</td>
<td>N/A</td>
</tr>
<tr>
<td>Contact Location</td>
<td>N/A</td>
</tr>
<tr>
<td>Contact Notes</td>
<td>N/A</td>
</tr>
<tr>
<td>Member of Group</td>
<td>/Default Group</td>
</tr>
<tr>
<td>Plugin</td>
<td>Windows File System(3001)</td>
</tr>
<tr>
<td>Initial Install Date</td>
<td>2005-10-13 12:28:14 PDT</td>
</tr>
<tr>
<td>Last Version Registered</td>
<td>2005-10-13 12:28:14 PDT</td>
</tr>
<tr>
<td>Last Successful Backup</td>
<td>2005-11-01 22:00:41 PST</td>
</tr>
</tbody>
</table>
client show-plugins

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

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/) Specifies domain containing client specified by the --name argument.

--name=STRING Specifies client for which plug-ins are shown. STRING must be a valid client name. This argument is required.

TIP: Use mccli client show (page 70) to return a list of valid client names.

IMPORTANT: If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain argument is ignored.

--plugin=INTEGER Specifies plug-in ID.

TIP: Use plugin show (page 153) to return a list of valid plug-in IDs.

--verbose=BOOLEAN(false) If set true, detailed (verbose) plug-in information is returned.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22236 Client does not exist.

23009 Invalid plugin specified on the CLI.
**client snapup-dataset**

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

**TIP:** This command returns an activity ID, which can be passed to `mccli activity show` (page 28) in order to get status for this backup activity.

**IMPORTANT:** This command has been deprecated in favor of `mccli client backup-dataset` (page 54). Whenever possible, you should use `mccli client backup-dataset` instead of `mccli client snapup-dataset`.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- **--cmd=STRING**
  Specifies one or more optional plug-in commands. Multiple `--cmd` arguments can be supplied but each argument can only specify one plug-in command. For example, this is valid `--cmd` syntax:
  ```
  --cmd="verbose=5" --cmd="throttle=5"
  ```
  However, this is not valid `--cmd` syntax:
  ```
  --cmd="verbose=5 throttle=5"
  ```
  Refer to your *Avamar System Administration Guide* for a list of valid plug-in commands that can specified during a backup operation.

- **--dataset=STRING**
  Specifies dataset to be used for this backup operation. If this argument is not supplied, the default dataset associated with that client is used.
  **TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.
Display Options
Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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.
The `mccli client snapup-group-dataset` command initiates an on-demand backup of a single client using the group dataset.

**NOTE:** It must be understood that this command initiates an on-demand client backup, not an on-demand group backup. Therefore, even if this client is a member of a group that is disabled and even if that group’s dataset is specified, the backup will still occur.

**IMPORTANT:** The client must already be a member of this group or the backup will fail.

**TIP:** This command returns an activity ID, which can be passed to `mccli activity show` (page 28) in order to get status for this backup activity.

**IMPORTANT:** This command has been deprecated in favor of `mccli client backup-group-dataset` (page 56). Whenever possible, you should use `mccli client backup-group-dataset` instead of `mccli client snapup-group-dataset`.

### Synopsis

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

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.
Command Options

**--cmd=STRING**  
Specifies one or more optional plug-in commands. Multiple `--cmd` arguments can be supplied but each argument can only specify one plug-in command.

For example, this is valid `--cmd` syntax:

```
--cmd="verbose=5" --cmd="throttle=5"
```

However, this is not valid `--cmd` syntax:

```
--cmd="verbose=5 throttle=5"
```

Refer to your _Avamar System Administration Guide_ for a list of valid plug-in commands that can specified during a backup operation.

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

**--group-domain=STRING(/)**  
Specifies Avamar server domain containing the group specified by the `--group-name` argument.

**--group-name=STRING**  
Specifies which group dataset to use for this backup. STRING must be a valid group name. This argument is required.

**TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

**IMPORTANT:** If a fully-qualified group name (for example, /MyDomain/MyGroup) is supplied, the `--group-domain` argument is ignored.

**--name=STRING**  
Specifies which client is backed up. STRING must be a valid client name. This argument is required.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the `--domain` argument is ignored.

Display Options

Refer to _Display Options_ (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
Event Codes

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

22234  Group does not exist.

22236  Client does not exist.

22241  Client is not a member of group.

22253  Client Adhoc Backup Request Error - Exception.

22305  Client backup scheduled.

23003  Invalid dataset name specified on the CLI.
client snapup-target

The `mccli client snapup-target` command initiates an on-demand backup of a single file or folder (directory) on the specified client.

**TIP:** To backup more than one file or folder (directory), create a dataset, then use `mccli client snapup-dataset` (page 74) to perform the backup.

**TIP:** This command returns an activity ID, which can be passed to `mccli activity show` (page 28) in order to get status for this backup activity.

**IMPORTANT:** This command has been deprecated in favor of `mccli client backup-target` (page 58). Whenever possible, you should use `mccli client backup-target` instead of `mccli client snapup-target`.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- **--cmd=STRING**
  
  Specifies one or more optional plug-in commands. Multiple `--cmd` arguments can be supplied but each argument can only specify one plug-in command.
  
  For example, this is valid `--cmd` syntax:
  
  ```
  --cmd="verbose=5" --cmd="throttle=5"
  ```
  
  However, this is not valid `--cmd` syntax:
  
  ```
  --cmd="verbose=5 throttle=5"
  ```
  
  Refer to your *Avamar System Administration Guide* for a list of valid plug-in commands that can specified during a backup operation.

- **--domain=STRING(/)**
  
  Specifies Avamar server domain containing the client specified by the `--name` argument.
--name=STRING Specifies which client is backed up. STRING must be a valid client name. This argument is required.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the `--domain` argument is ignored.

--plugin=INTEGER Specifies plug-in ID.

**TIP:** Use `plugin show` (page 153) to return a list of valid plug-in IDs.

--target=STRING Specifies which file or folder (directory) to include in backups. This argument is required. This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to `Pattern Matching Operators` (page 21) for additional information.

**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.
client validate-bulk

The `mccli client validate-bulk` command validates a clients definition input file, which is used by the `mccli client load-bulk` command (page 67). This validation process ensures that the XML data is properly formatted and that the file will work with the `mccli client load-bulk` command.

**Synopsis**

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

**Global Options**

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- `--file=STRING` Specifies clients definition input file. This file must be in XML format. This argument is required. Refer to your Avamar System Administration Guide for additional information about clients definition files.
- `--verbose=BOOLEAN(false)` If set true, a list of clients and their load status is output to stdout.

**Display Options**

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

23006 CLI client bulk load encountered one or more errors.
Dataset Resource Overview

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

- `dataset add` (page 83)
- `dataset add-exclude` (page 84)
- `dataset add-include` (page 86)
- `dataset add-option` (page 88)
- `dataset add-target` (page 90)
- `dataset copy` (page 92)
- `dataset delete` (page 93)
- `dataset delete-exclude` (page 94)
- `dataset delete-include` (page 96)
- `dataset delete-option` (page 98)
- `dataset delete-target` (page 99)
- `dataset replace` (page 101)
- `dataset show` (page 103)
dataset add

The `mccli dataset add` command creates (adds) a new dataset. By default, the initial dataset will use all available source data plug-ins and will contain no explicit exclusion or inclusion entry entries.

Other dataset commands, such as `dataset add-exclude` (page 84), `dataset add-option` (page 88), `dataset add-include` (page 86), `dataset add-target` (page 90), `dataset delete-exclude` (page 94), `dataset delete-option` (page 98), `dataset delete-include` (page 96), `dataset delete-target` (page 99) and `dataset replace` (page 101) can be used to customize the dataset after it has been created.

Synopsis

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

Global Options

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

```
--alldata=BOOLEAN(true)
```
If set true, the initial dataset will use all available source data plug-ins and will contain no explicit exclusion or inclusion entry entries.

**TIP:** Set `--alldata=false` to create an empty dataset.

```
--domain=STRING(/)
```
Specifies Avamar server domain containing the dataset specified by the `--name` argument.

```
--name=STRING
```
Specifies new dataset name. This argument is required.

**IMPORTANT:** If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the `--domain` argument is ignored.

Display Options

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

Event Codes

- 22219 Dataset created.
- 23008 Dataset already exists.
dataset add-exclude

The `mccli dataset add-exclude` command adds an exclusion entry to an existing dataset.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--domain=STRING(/)  Specifies Avamar server domain containing the dataset specified by the --name argument.

--exclude=STRING   Specifies which files or folders (directories) to exclude from this dataset.
                  Either --exclude=STRING or --exclude-file=FILE must be supplied.
                  Multiple --exclude=STRING arguments are allowed and must be separated by whitespace.
                  This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to *Pattern Matching Operators* (page 21) for additional information.

--exclude-file=FILE Specifies the full path of a text FILE that contains one or more exclusion entries. Each exclusion entry must conform to allowable --exclude=STRING syntax and be on a single line.
                  Either --exclude=STRING or --exclude-file=FILE must be supplied.

--name=STRING      Specifies which dataset to modify. STRING must be a valid dataset name. This argument is required.
                  **TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.

                  **IMPORTANT:** If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the --domain argument is ignored.

--plugin=INTEGER   Specifies plug-in ID. This argument is required.
                  **TIP:** Use `plugin show` (page 153) to return a list of valid plug-in IDs.
**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**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 an existing dataset.

Synopsis

mccli dataset add-include [GLOBAL-OPTIONS] [--domain=STRING(/)]

{{--include=STRING [--include=STRING ... | --include-file=FILE]
--name=STRING --plugin=INTEGER [DISPLAY-OPTIONS]}}

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/)  Specifies Avamar server domain containing the dataset specified by the --name argument.

--include=STRING  Specifies which files or folders (directories) to add back into this dataset after an exclusion entry has excluded them.

Either --include=STRING or --include-file=FILE must be supplied.

Multiple --include=STRING arguments are allowed and must be separated by whitespace.

This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to Pattern Matching Operators (page 21) for additional information.

--include-file=FILE  Specifies the full path of a text FILE that contains one or more inclusion entries. Each include entry must conform to allowable --include=STRING syntax and be on a single line.

Either --include=STRING or --include-file=FILE must be supplied.

--name=STRING  Specifies which dataset to modify. STRING must be a valid dataset name. This argument is required.

TIP: Use mccli dataset show (page 103) to return a list of valid dataset names.

IMPORTANT: If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the --domain argument is ignored.

--plugin=INTEGER  Specifies plug-in ID. This argument is required.

TIP: Use plugin show (page 153) to return a list of valid plug-in IDs.
Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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 an existing dataset.

**Synopsis**

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

**Global Options**

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

--domain=STRING(/)  Specifies Avamar server domain containing the dataset specified by the --name argument.

--name=STRING  Specifies which dataset to modify. STRING must be a valid dataset name. This argument is required.

  **TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.

  **IMPORTANT:** If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the --domain argument is ignored.

--option=STRING  Specifies which plug-in command to add to this dataset. This argument is required.

  Multiple --option=STRING arguments are allowed and must be separated by whitespace.

  Refer to your `Avamar System Administration Guide` for a list of valid plug-in commands that can be added to a dataset.

--plugin=INTEGER  Specifies plug-in ID. This argument is required.

  **TIP:** Use `plugin show` (page 153) to return a list of valid plug-in IDs.

--value=STRING  Specifies a value for --option. This argument is required.

  Refer to your `Avamar System Administration Guide` for a list of valid plug-in commands that can be added to a dataset.

**Display Options**

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
### Event Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23009</td>
<td>Invalid plugin specified on the CLI.</td>
</tr>
<tr>
<td>22220</td>
<td>Dataset modified.</td>
</tr>
<tr>
<td>22288</td>
<td>Dataset does not exist.</td>
</tr>
</tbody>
</table>

### Notes

Plug-in commands and values, specified by the `--option=STRING` and `--value=STRING` options, respectively must occur in equal numbers. If multiple `--option=STRING` options are specified, an equal number of `--value=STRING` options must be supplied in order to specify values for all plug-in commands. Furthermore, the option-value pairing is positional (that is, 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 (directory) entry to an existing dataset.

Synopsis

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

Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

```
--domain=STRING/  Specifies Avamar server domain containing the dataset specified by the --name argument.

--name=STRING    Specifies which dataset to modify. STRING must be a valid dataset name. This argument is required.

  **TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.

  **IMPORTANT:** If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the
  --domain argument is ignored.

--plugin=INTEGER  Specifies plug-in ID. This argument is required.

  **TIP:** Use `plugin show` (page 153) to return a list of valid plug-in IDs.

--target=STRING   Specifies a target file or folder (directory) to add to this dataset. This argument is required.

  Either --target=STRING or --target-file=FILE must be supplied.

  Multiple --target=STRING arguments are allowed and must be separated by whitespace.

  This argument accepts regular expression (regex) pattern matching operators (also known as wild cards).
  Refer to *Pattern Matching Operators* (page 21) for additional information.

--target-file=FILE Specifies the full path of a text FILE that contains one or more target entries. Each include entry must conform to allowable --target=STRING syntax and be on a single line.

  Either --target=STRING or --target-file=FILE must be supplied.
Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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.
The `mccli dataset copy` command copies an existing dataset, creating a new dataset.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- **--domain=STRING(//)**
  Specifies Avamar server domain containing the dataset specified by the `--name` argument.

- **--name=STRING**
  Specifies which dataset to copy. STRING must be a valid dataset name. This argument is required.
  **TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.
  **IMPORTANT:** If a fully-qualified dataset name (for example, `/MyDomain/MyDataset`) is supplied, the `--domain` argument is ignored.

- **--new-domain=STRING(//)**
  Specifies Avamar server domain where the new dataset should be created.

- **--new-name=STRING**
  Specifies new dataset name. This argument is required.
  **IMPORTANT:** If a fully-qualified dataset name (for example, `/MyDomain/MyDataset`) is supplied, the `--new-domain` argument is ignored.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

- 22219  Dataset created.
- 22288  Dataset does not exist.
**dataset delete**

The `mccli dataset delete` command permanently removes (deletes) the specified dataset from the Avamar server.

---

**IMPORTANT:** You cannot delete a dataset if it is currently assigned to any client or group.

---

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

--domain=STRING[/] Specifies Avamar server domain containing the dataset specified by the --name argument.

--name=STRING Specifies which dataset to delete. STRING must be a valid dataset name. This argument is required.

**TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.

**IMPORTANT:** If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the --domain argument is ignored.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

22221   Dataset deleted.

22288   Dataset does not exist.
dataset delete-exclude

The `mccli dataset delete-exclude` command permanently removes (deletes) one or more existing exclusion entries from an existing dataset.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- `--all`
  Deletes all existing exclusion entries from this dataset.
  Either `--all`, `--exclude=STRING` or `--exclude-file=FILE` must be supplied.

- `--domain=STRING(/)`
  Specifies Avamar server domain containing the dataset specified by the `--name` argument.

- `--exclude=STRING`
  Specifies which exclusion entry to remove (delete) from this dataset.
  Either `--all`, `--exclude=STRING` or `--exclude-file=FILE` must be supplied.
  Multiple `--exclude=STRING` arguments are allowed and must be separated by whitespace.
  This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to *Pattern Matching Operators* (page 21) for additional information.

- `--exclude-file=FILE`
  Specifies the full path of a text FILE that contains one or more exclusion entries. Each exclusion entry must conform to allowable `--exclude=STRING` syntax and be on a single line.
  Either `--all`, `--exclude=STRING` or `--exclude-file=FILE` must be supplied.

- `--name=STRING`
  Specifies which dataset to modify. STRING must be a valid dataset name. This argument is required.
  **TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.
  **IMPORTANT:** If a fully-qualified dataset name (for example, `/MyDomain/MyDataset`) is supplied, the `--domain` argument is ignored.
--plugin=INTEGER    Specifies plug-in ID. This argument is required.

**TIP:** Use `plugin show` (page 153) to return a list of valid plug-in IDs.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

### 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.
The `mccli dataset delete-include` command permanently removes (deletes) one or more existing inclusion entries from an existing dataset.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

**--all**

Deletes all existing inclusion entries from this dataset.

Either `--all`, `--include=STRING` or `--include-file=FILE` must be supplied.

**--domain=STRING(/)**

Specifies Avamar server domain containing the dataset specified by the `--name` argument.

**--include=STRING**

Specifies which inclusion entry to remove (delete) from this dataset.

Either `--all`, `--include=STRING` or `--include-file=FILE` must be supplied.

Multiple `--include=STRING` arguments are allowed and must be separated by whitespace.

This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to *Pattern Matching Operators* (page 21) for additional information.

**--include-file=FILE**

Specifies the full path of a text FILE that contains one or more inclusion entries. Each include entry must conform to allowable `--include=STRING` syntax and be on a single line.

Either `--all`, `--include=STRING` or `--include-file=FILE` must be supplied.

**--name=STRING**

Specifies which dataset to modify. STRING must be a valid dataset name. This argument is required.

**TIP:** Use `mccli dataset show` (page 103) to return a list of valid dataset names.

**IMPORTANT:** If a fully-qualified dataset name (for example, `/MyDomain/MyDataset`) is supplied, the `--domain` argument is ignored.
---plugin=INTEGER Specifies plug-in ID. This argument is required.

**TIP:** Use `plugin show` (page 153) to return a list of valid plug-in IDs.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

### 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 \texttt{mccli dataset delete-option} command permanently removes (deletes) an existing plug-in command (option) from an existing dataset.

Synopsis

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

Global Options

Refer to \textit{Global Options} (page 24) for a list of GLOBAL-OPTIONS common to all \texttt{mccli} commands.

Command Options

\texttt{--domain=STRING(/)} Specifies Avamar server domain containing the dataset specified by the \texttt{--name} argument.

\texttt{--name=STRING} Specifies which dataset to modify. STRING must be a valid dataset name. This argument is required.
\textit{TIP:} Use \texttt{mccli dataset show} (page 103) to return a list of valid dataset names.
\textit{IMPORTANT:} If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the \texttt{--domain} argument is ignored.

\texttt{--option=STRING} Specifies which plug-in command to remove (delete) from this dataset. This argument is required.
Multiple \texttt{--option=STRING} arguments are allowed and must be separated by whitespace.

\texttt{--plugin=INTEGER} Specifies plug-in ID. This argument is required.
\textit{TIP:} Use \texttt{plugin show} (page 153) to return a list of valid plug-in IDs.

Display Options

Refer to \textit{Display Options} (page 25) for a list of DISPLAY-OPTIONS common to all \texttt{mccli} commands.

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 removes (deletes) a target file or folder (directory) from an existing dataset.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/) Specifies Avamar server domain containing the dataset specified by the --name argument.

--name=STRING Specifies which dataset to modify. STRING must be a valid dataset name. This argument is required.

TIP: Use mccli dataset show (page 103) to return a list of valid dataset names.

IMPORTANT: If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the --domain argument is ignored.

--plugin=INTEGER Specifies plug-in ID. This argument is required.

TIP: Use plugin show (page 153) to return a list of valid plug-in IDs.

--target=STRING Specifies which file or folder (directory) to remove (delete) from this dataset. This argument is required.

Either --target=STRING or --target-file=FILE must be supplied.

Multiple --target=STRING arguments are allowed and must be separated by whitespace.

This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to Pattern Matching Operators (page 21) for additional information.

--target-file=FILE Specifies the full path of a text FILE that contains one or more target entries. Each include entry must conform to allowable --target=STRING syntax and be on a single line.

Either --target=STRING or --target-file=FILE must be supplied.
Display Options
Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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 replace

The `mccli dataset replace` command replaces all existing exclusion and inclusion entries for an existing dataset with the entries supplied on the command line.

**Synopsis**

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

**Command Options**

- `--domain=STRING(/)` Specifies Avamar server domain containing the dataset specified by the `--name` argument.
- `--exclude=STRING` Specifies which files or folders (directories) to exclude from this dataset.
  Multiple `--exclude=STRING` arguments are allowed and must be separated by whitespace.
  This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to *Pattern Matching Operators* (page 21) for additional information.
- `--exclude-file=FILE` Specifies the full path of a text FILE that contains one or more exclusion entries. Each exclusion entry must conform to allowable `--exclude=STRING` syntax and be on a single line.
- `--include=STRING` Specifies which files or folders (directories) to add back into this dataset after an exclusion entry has excluded them.
  Either `--include=STRING` or `--include-file=FILE` must be supplied.
  Multiple `--include=STRING` arguments are allowed and must be separated by whitespace.
  This argument accepts regular expression (regex) pattern matching operators (also known as wild cards). Refer to *Pattern Matching Operators* (page 21) for additional information.
- `--include-file=FILE` Specifies the full path of a text FILE that contains one or more inclusion entries. Each include entry must conform to allowable `--include=STRING` syntax and be on a single line.
  Either `--include=STRING` or `--include-file=FILE` must be supplied.
dataset replace

Display Options
Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/)    Specifies Avamar server domain containing the dataset specified by the --name argument.

--name=STRING        Specifies a dataset name. STRING must be a valid dataset name. If not supplied, all datasets are listed with summary information. IMPORTANT: If a fully-qualified dataset name (for example, /MyDomain/MyDataset) is supplied, the --domain argument is ignored.

--recursive=BOOLEAN(false)    If set true, command shows all datasets in the domain and any sub-domains specified by the --domain=STRING argument.
    If set false or not supplied, command only shows datasets within the specified domain (sub-domains are not examined).

--verbose=BOOLEAN(false)    If set true, detailed dataset information is returned.
    If set false or not supplied, only dataset names are returned.
    If -name is supplied, detailed information for that specific dataset is returned.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22288  Dataset does not exist.
Examples

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

```
mcli dataset show
```

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unix Dataset</td>
<td>/</td>
</tr>
<tr>
<td>Windows Dataset</td>
<td>/</td>
</tr>
<tr>
<td>Default Dataset</td>
<td>/</td>
</tr>
<tr>
<td>Base Dataset</td>
<td>/</td>
</tr>
</tbody>
</table>

This command returns summary information for all datasets currently in use:

```
mcli dataset show --verbose=true
```

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Domain</th>
<th>Num Targets</th>
<th>Num Includes</th>
<th>Num Excludes</th>
<th>Num Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unix Dataset</td>
<td>/</td>
<td>4</td>
<td>0</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>Windows Dataset</td>
<td>/</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Default Dataset</td>
<td>/</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Base Dataset</td>
<td>/</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
This command returns detailed information for a single dataset:

```
mccli dataset show --name='Unix Dataset'
```

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Unix Dataset</td>
</tr>
<tr>
<td>ID</td>
<td>UNIX:SNAPID</td>
</tr>
<tr>
<td>Domain</td>
<td>/</td>
</tr>
<tr>
<td>ReadOnly</td>
<td>false</td>
</tr>
<tr>
<td>ALLDATA</td>
<td>true</td>
</tr>
<tr>
<td>Num Targets</td>
<td>4</td>
</tr>
<tr>
<td>Linux File System Target</td>
<td>ALL</td>
</tr>
<tr>
<td>Linux File System Target ID</td>
<td>1001</td>
</tr>
<tr>
<td>Solaris File System Target</td>
<td>ALL</td>
</tr>
<tr>
<td>Solaris File System Target ID</td>
<td>2001</td>
</tr>
<tr>
<td>HP-UX File System Target</td>
<td>ALL</td>
</tr>
<tr>
<td>HP-UX File System Target ID</td>
<td>4001</td>
</tr>
<tr>
<td>AIX File System Target</td>
<td>ALL</td>
</tr>
<tr>
<td>AIX File System Target ID</td>
<td>5001</td>
</tr>
<tr>
<td>Num Includes</td>
<td>0</td>
</tr>
<tr>
<td>Num Excludes</td>
<td>12</td>
</tr>
<tr>
<td>Linux File System Exclude</td>
<td>/tmp</td>
</tr>
<tr>
<td>Linux File System Exclude</td>
<td>/var/tmp</td>
</tr>
<tr>
<td>Linux File System Exclude</td>
<td>/usr/tmp</td>
</tr>
<tr>
<td>Linux File System Exclude</td>
<td>core</td>
</tr>
<tr>
<td>Linux File System Exclude</td>
<td>*cache.dat</td>
</tr>
<tr>
<td>Linux File System Exclude</td>
<td>*scan.dat</td>
</tr>
<tr>
<td>Solaris File System Exclude</td>
<td>/tmp</td>
</tr>
<tr>
<td>Solaris File System Exclude</td>
<td>/var/tmp</td>
</tr>
<tr>
<td>Solaris File System Exclude</td>
<td>/usr/tmp</td>
</tr>
<tr>
<td>Solaris File System Exclude</td>
<td>core</td>
</tr>
<tr>
<td>Solaris File System Exclude</td>
<td>*cache.dat</td>
</tr>
<tr>
<td>Solaris File System Exclude</td>
<td>*scan.dat</td>
</tr>
<tr>
<td>Num Options</td>
<td>0</td>
</tr>
<tr>
<td>Is Link</td>
<td>false</td>
</tr>
</tbody>
</table>
Domain Resource Overview

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

- `domain add` (page 107)
- `domain delete` (page 108)
- `domain edit` (page 109)
- `domain show` (page 110)
domain add

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

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

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

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22526 Failed directory add.
22527 Domain added.
22540 Invalid name character.
22541 Domain already exists.
22558 A domain or client with this name already exists.
The `mccli domain delete` command permanently removes (deletes) a domain from the Avamar server.

**IMPORTANT:** Deleting a domain deletes the domain and all sub-domains, clients and backups stored within that domain.

### Synopsis

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

### Command Options

- `--domain=STRING` Specifies parent domain. This argument is required.
- `--name=STRING` Specifies which domain to delete. STRING must be a valid domain name. This argument is required.

**TIP:** Use `mccli domain show` (page 110) to return a list of valid domain names.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

### 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 modifies (edits) one or more properties for a specific domain.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

**Command Options**

- `--contact=STRING` Specifies contact information for the domain.
- `--domain=STRING(/)` Specifies parent domain.
- `--email=STRING` Specifies primary contact (responsible party) email address.
- `--location=STRING` Specifies location information for the domain.
- `--name=STRING` Specifies which domain to edit. STRING must be a valid domain name. This argument is required. **TIP:** Use `mccli domain show` (page 110) to return a list of valid domain names.
- `--phone=STRING` Specifies primary contact (responsible party) voice telephone information.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

**Event Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22519</td>
<td>Domain modified.</td>
</tr>
<tr>
<td>22542</td>
<td>Domain does not exist.</td>
</tr>
</tbody>
</table>
domain show

The mccli domain show command lists (shows) properties for the specified domain.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/) Specifies parent domain under which domains and sub-domains are listed.

--name=STRING Specifies name of domain to delete. STRING must be a valid domain name.

--recursive=BOOLEAN(false) If set true, command examines all domains within the domain and any sub-domains. If set false or not supplied, command only examines the specified domain (sub-domains are not examined).

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22542 Domain does not exist.
Dump Resource Overview

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

- `dump clientcache` (page 111)
- `dump domaincache` (page 112)
- `dump jobcache` (page 112)

**dump clientcache**

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

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

**Synopsis**

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

**Global Options**

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
dump domaincache

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

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

**Synopsis**

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

**Global Options**

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

.dump jobcache

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

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

**Synopsis**

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

**Global Options**

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
Event Resource Overview

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

- `event ack` (page 114)
- `event get-info` (page 117)
- `event show` (page 118)
**event ack**

The `mccli event ack` command acknowledges one or more events.

**Synopsis**

```bash
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```bash
--after=STRING
```

Acknowledges events after this 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 time stamp.

```bash
--all=BOOLEAN(false)
```

Acknowledges all events.

```bash
--before=STRING
```

Acknowledges events before this 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 time stamp.

```bash
--category=STRING
```

Specifies event category. STRING must be one of the following:

- APPLICATION
- SECURITY
- SYSTEM
- USER

```bash
--domain=STRING
```

Specifies event domain

```bash
--exclude=STRING
```

Comma-separated list of event codes to exclude. **IMPORTANT:** This comma-separated list cannot contain spaces.

--include and --exclude are mutually exclusive.

```bash
--id=INTEGER
```

Unique event identifier
--include=STRING Comma-separated list of event codes to include. **IMPORTANT:** This comma-separated list cannot contain spaces.

--include and --exclude are mutually exclusive.

--range=STRING Specifies event ID range. 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.

--severity=STRING Specifies event severity. STRING must be one of the following:

- NODE
- NODE_FATAL
- OK
- PROCESS
- PROCESS_FATAL
- SYSTEM_FATAL
- USER
- USER_FATAL

--type=STRING Specifies event type. STRING must be one of the following:

- AUDIT
- DEBUG
- ERROR
- INFORMATION
- INTERNAL
- WARNING

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

**Event Codes**

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.

Output is a count of the events acknowledged.

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

This command acknowledges an event code (ID 148710):

```
mccli event ack --id=148710
```

0,23000,CLI command completed successfully.

Attribute    Value
------------- ----
events-acked 1
event get-info

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

Synopsis

```
mccli event get-info [GLOBAL-OPTIONS] --code=INTEGER
    [--description=BOOLEAN(false)] [--remedy=BOOLEAN(false)]
    [--summary=BOOLEAN(true)] [DISPLAY-OPTIONS]
```

Global Options

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

- `--code=INTEGER` Specifies event code number. This argument is required.

- `--description=BOOLEAN(false)` If set true, other properties are also returned, including the remedy and summary properties.
  If set false or not supplied, only the code is returned.

- `--remedy=BOOLEAN(false)` If set true, event remedy information is returned.
  If set false or not supplied, the event remedy is not returned.

- `--summary=BOOLEAN(true)` If set true or not supplied, event summary information is returned.
  If set false, the event summary is not returned.

Display Options

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
The `mccli event show` command returns event occurrence details.

**Synopsis**
```
mccli event show [GLOBAL-OPTIONS] [--after=STRING] [--before=STRING]
   [--category=STRING] [--domain=STRING] [--exclude=STRING]
   [--id=INTEGER] [--include=STRING] [--severity=STRING]
   [--type=STRING] [--unack=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

**Global Options**
Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

**Command Options**

--after=STRING  
Returns events after 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 time stamp.

--before=STRING  
Returns events before 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 time stamp.

--category=STRING  
Specifies event category. STRING must be one of the following:
   • APPLICATION
   • SECURITY
   • SYSTEM
   • USER

--domain=STRING  
Specifies event domain

--exclude=STRING  
Comma-separated list of event codes to exclude.  
**IMPORTANT:** This comma-separated list cannot contain spaces.

   --include and --exclude are mutually exclusive.

--id=INTEGER  
Unique event identifier

--include=STRING  
Comma-separated list of event codes to include.  
**IMPORTANT:** This comma-separated list cannot contain spaces.

   --include and --exclude are mutually exclusive.
### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

### 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 output for multiple event occurrences:

```bash
mccli event show
```

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

<table>
<thead>
<tr>
<th>ID</th>
<th>Date</th>
<th>Type</th>
<th>Code</th>
<th>Category</th>
<th>Severity</th>
<th>Domain</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>149897</td>
<td>2008-04-19</td>
<td>14:19:45</td>
<td>PDT</td>
<td>INFORMATION</td>
<td>578</td>
<td>SYSTEM</td>
<td>OK / hfscheckresults::merge, in checkpoint cp.20080419140741, this ((start 1176995040, nodestart 1176995040, nodefinish 0, valid false, totalerrors 0)) tried to merge with ((start 0, nodestart 0, nodefinish 0, valid false, totalerrors 0)) but the uniquestarttime fields differ</td>
</tr>
</tbody>
</table>
This command returns detailed information about a specific event occurrence (ID):

```
mccli event show --id=149897
```

0,23000,CLI command completed successfully.

```
Command completed successfully
```

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>149897</td>
</tr>
<tr>
<td>Date</td>
<td>2008-04-19 14:19:45 PDT</td>
</tr>
<tr>
<td>Type</td>
<td>INFORMATION</td>
</tr>
<tr>
<td>Code</td>
<td>578</td>
</tr>
<tr>
<td>Category</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>Severity</td>
<td>OK</td>
</tr>
<tr>
<td>Domain</td>
<td>/</td>
</tr>
<tr>
<td>Summary</td>
<td>hfscheckresults::merge, in checkpoint cp.20080419140741, this ((start 1176995040, nodestart 1176995040, nodefinish 0, valid false, totalerrors 0)) tried to merge with ((start 0, nodestart 0, nodedefinition 0, valid false, totalerrors 0)) but the uniquestarttime fields differ</td>
</tr>
<tr>
<td>SW Source</td>
<td>DPN:Unknown</td>
</tr>
<tr>
<td>For Whom</td>
<td>N/A</td>
</tr>
<tr>
<td>HW Source</td>
<td>node-10-0-54-249.example.com</td>
</tr>
<tr>
<td>Description</td>
<td>N/A</td>
</tr>
<tr>
<td>Remedy</td>
<td>N/A</td>
</tr>
<tr>
<td>Notes</td>
<td>N/A</td>
</tr>
</tbody>
</table>

```
<data><entry key="code" type="text" value="0578" version=""/></entry><entry key="type" type="text" value="INFO" version=""/></entry><entry key="time" type="text" value="21:19:45.93935" version=""/></entry><entry key="message" type="text" value="hfscheckresults::merge, in checkpoint cp.20080419140741, this ((start 1176995040, nodestart 1176995040, nodefinish 0, valid false, totalerrors 0)) tried to merge with ((start 0, nodestart 0, nodedefinition 0, valid false, totalerrors 0)) but the uniquestarttime fields differ" version=""/></entry><entry key="date" type="text" value="2008/04/19" version=""/></entry><entry key="thread" type="text" value="srvm-14070642#srv:265" version=""/></data>
```
This command shows when the maximum number of events displayed is reached:

```
mccli event show
```

INFO: The maximum number of events displayed limit reached. The maximum is 5000.
0,23000,CLI command completed successfully.

<table>
<thead>
<tr>
<th>ID</th>
<th>Date</th>
<th>Type</th>
<th>Code</th>
<th>Category</th>
<th>Severity</th>
<th>Domain</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>149997</td>
<td>2008-04-19 14:44:46 PDT</td>
<td>INFORMATION</td>
<td>578</td>
<td>SYSTEM</td>
<td>OK</td>
<td>/</td>
<td>hfscheckresults::merge, in checkpoint cp.20080419140741, this ((start 1176995040, nodestart 1176995040, nodefinish 0, valid false, totalerrors 0)) tried to merge with ((start 0, nodestart 0, nodefinish 0, valid false, totalerrors 0)) but the uniquestarttime fields differ</td>
</tr>
</tbody>
</table>
Group Resource Overview

The `mccli group` resource is used to create and manage groups and group policy on the Avamar server. Available commands are:

- `group add` (page 124)
- `group add-client` (page 126)
- `group backup` (page 128)
- `group copy` (page 129)
- `group delete` (page 131)
- `group edit` (page 132)
- `group export` (page 134)
- `group move-client` (page 135)
- `group remove-client` (page 137)
- `group show` (page 139)
- `group show-client-members` (page 141)
- `group snapup` (page 142)

**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 `dataset show` (page 103), `schedule show` (page 171), `retention show` (page 161) and `plugin show` (page 153) commands, respectively.

You can also view a list of valid datasets, retention policies and plug-ins using the Avamar Administrator Policy > Tools > Manage Datasets..., Manage Schedules..., Manage Retention Policies... and Manage Agents & Plug-ins... features, respectively.
group add

The mccli group add command creates (adds) a new group.

Synopsis

mccli group add [GLOBAL-OPTIONS] --dataset=STRING (Default Dataset) [--dataset-domain=STRING (/)] [--domain=STRING (/)] --enabled=BOOLEAN (false) --name=STRING --retention=STRING (Default Retention) [--retention-domain=STRING (/)] --schedule=STRING (Default Schedule) [--schedule-domain=STRING (/)] [DISPLAY-OPTIONS]

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--dataset=STRING (Default Dataset) Specifies which dataset to assign to this group. This argument is required.

--dataset-domain=STRING (/) Specifies Avamar server domain containing the dataset specified by the --dataset argument.

--domain=STRING (/) Specifies Avamar server domain containing the group specified by the --name argument.

--enabled=BOOLEAN (false) If set true, the group is eligible to immediately participate in scheduled backup activities. This argument is required.

--name=STRING Specifies new group name. This argument is required.

IMPORTANT: If a fully-qualified group name (for example, /MyDomain/MyGroup) is supplied, the --domain argument is ignored.

--retention=STRING (Default Retention) Specifies which retention policy to assign to this group. This argument is required.

--retention-domain=STRING (/) Specifies Avamar server domain containing the retention policy specified by the --retention argument.
**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

**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.
group add-client

The `mccli group add-client` command adds the specified client to the specified group.

Synopsis

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

Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

- `--client-domain=STRING(/)` Specifies domain of the client to add to the group.
- `--client-name=STRING` Specifies which client to add to this group. STRING must be a valid client name. This argument is required.
  **TIP:** Use `mccli client show` (page 70) to return a list of valid client names.
  **IMPORTANT:** If a fully-qualified client name (for example, `/MyDomain/MyClient`) is supplied, the `--client-domain` argument is ignored.
- `--domain=STRING(/)` Specifies Avamar server domain containing the group specified by the `--name` argument.
- `--name=STRING` Specifies group to which client will be added. STRING must be a valid group name. This argument is required.
  **TIP:** Use `mccli group show` (page 139) to return a list of valid group names.
  **IMPORTANT:** If a fully-qualified group name (for example, `/MyDomain/MyGroup`) is supplied, the `--domain` argument is ignored.

Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
### Event Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>22234</td>
<td>Group does not exist.</td>
</tr>
<tr>
<td>22236</td>
<td>Client does not exist.</td>
</tr>
<tr>
<td>22242</td>
<td>Client is already a member of this group.</td>
</tr>
<tr>
<td>22243</td>
<td>Client group membership successfully updated.</td>
</tr>
<tr>
<td>22269</td>
<td>Unable to add a client to a group.</td>
</tr>
</tbody>
</table>
group backup

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

Synopsis

mccli group backup [GLOBAL-OPTIONS] [--domain=STRING(/)]
--name=STRING [DISPLAY-OPTIONS]

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/) Specifies Avamar server domain containing the group specified by the --name argument.

--name=STRING Specifies which group to back up. STRING must be a valid group name. This argument is required.

TIP: Use mccli group show (page 139) to return a list of valid group names.

IMPORTANT: If a fully-qualified group name (for example, /MyDomain/MyGroup) is supplied, the --domain argument is ignored.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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.
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 (the new group will not have any clients assigned to it).

**IMPORTANT:** You must copy groups within the same domain. You cannot copy a group to another domain under any circumstances.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- `--domain=STRING(/)` Specifies Avamar server domain containing the group specified by the `--name` argument.
- `--name=STRING` Specifies which group to copy. STRING must be a valid group name. This argument is required.
  
  **TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

  **IMPORTANT:** If a fully-qualified group name (for example, `/MyDomain/MyGroup`) is supplied, the `--domain` argument is ignored.

- `--new-name=STRING` Specifies new group name. This argument is required.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
Event Codes

22207  New group created.
22233  Group already exists.
22234  Group does not exist.
22235  Group add failed.
22540  Invalid name character.
The `mccli group delete` command permanently removes (deletes) a group from the Avamar server.

**IMPORTANT:** A client must always be a member of at least one group. Therefore, if this group contains any clients that are not also members of at least one other group, those clients must be moved to other groups, using the `mccli group move-client` command (page 135), before this group can be deleted.

### Synopsis

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

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

- **--domain=STRING[/]** Specifies Avamar server domain containing the group specified by the **--name** argument.

- **--name=STRING** Specifies which group to delete. STRING must be a valid group name. This argument is required. **TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

  **IMPORTANT:** If a fully-qualified group name (for example, `/MyDomain/MyGroup`) is supplied, the **--domain** argument is ignored.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

### 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.
The `mccli group edit` command modifies (edits) one or more properties for the specified group.

**IMPORTANT:** There are no default settings for the `mccli group edit` command. Entering this command without explicitly supplying options and values on the command line will have no effect on the specified resource.

### Synopsis

```
mccli group edit [GLOBAL-OPTIONS] --dataset=STRING
    [--dataset-domain=STRING(/)] [--domain=STRING] --enabled=BOOLEAN
    --name=STRING --retention=STRING [--retention-domain=STRING(/)]
    --schedule=STRING [--schedule-domain=STRING(/)] [DISPLAY-OPTIONS]
```

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

- **--dataset=STRING**
  Specifies dataset assigned to this group. This argument is required.

- **--dataset-domain=STRING(/)**
  Specifies Avamar server domain containing the dataset specified by the `--dataset` argument.

- **--domain=STRING**
  Specifies Avamar server domain containing the group specified by the `--name` argument.

- **--enabled=BOOLEAN**
  If set true, the group is eligible to immediately participate in scheduled backup activities. This argument is required.

- **--name=STRING**
  Specifies which group to modify (edit). STRING must be a valid group name. This argument is required. **TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

  **IMPORTANT:** If a fully-qualified group name (for example, `/MyDomain/MyGroup`) is supplied, the `--domain` argument is ignored.

- **--retention=STRING**
  Specifies retention policy assigned to this group. This argument is required.
Display Options
Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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.
group export

The `mccli group export` command exports the specified 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 manner do not contain a list of client members.

**IMPORTANT:** The `mccli group export` command only supports exporting of REGULAR groups. It does not support exporting of VCENTER groups (that is, Proxy or Virtual Machine groups).

**Synopsis**

```
mccli group export [GLOBAL-OPTIONS] [--domain=STRING(/)] --file=STRING --name=STRING [--force=Boolean(false)] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--domain=STRING(/)  Specifies Avamar server domain containing the group specified by the --name argument.
--file=STRING      Specifies the name of export XML file to be created. This option is required.
--force=Boolean(false) If set true, forces overwrite of an existing XML file.
--name=STRING      Specifies group to be exported. This option is required.
```

**TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

**IMPORTANT:** If a fully-qualified group name (for example, /MyDomain/In-group) is supplied, the --domain argument is ignored.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

22234  Group does not exist.
**group move-client**

The mccli group move-client command moves a specified client from one group to another.

**Synopsis**

```plaintext
mccli group move-client [GLOBAL-OPTIONS] [--client-domain=STRING(/)]
   --client-name=STRING [-domain=STRING(/)] --name=STRING
   [--old-group-domain=STRING(/)]
   [--old-group-name=STRING(Default Group)] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

**Command Options**

- `--client-domain=STRING(/)`
  - Specifies domain to which client will be moved.

- `--client-name=STRING`
  - Specifies which client to move. This argument is required.

- `--domain=STRING(/)`
  - Specifies Avamar server domain containing the group specified by the `--name` argument.

- `--name=STRING`
  - Specifies group to which this client will be moved. STRING must be a valid group name. This argument is required.
  - **TIP:** Use mccli group show (page 139) to return a list of valid group names.
  - **IMPORTANT:** If a fully-qualified group name (for example, /MyDomain/MyGroup) is supplied, the `--domain` argument is ignored.

- `--old-group-domain=STRING(/)`
  - Specifies domain from which this client will be moved.

- `--old-group-name=STRING(Default Group)`
  - Specifies group from which this client will be moved.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.
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.
The `mccli group remove-client` command removes the specified client from the specified group.

**IMPORTANT:** A client must always be a member of at least one group. Therefore, if the client you want to remove is only a member of this group, it must be added to another group, using the `mccli group add-client` command (page 126), before it can be removed from this group.

Refer to your *Avamar System Administration Guide* for additional information about moving a client from one group to another.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--client-domain=STRING(/)  Specifies domain from which client will be removed.
--client-name=STRING      Specifies which client to remove. This argument is required.
--domain=STRING(/)        Specifies Avamar server domain containing the group specified by the --name argument.
--name=STRING             Specifies group from which to remove the client. STRING must be a valid group name. This argument is required.
```

**TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

**IMPORTANT:** If a fully-qualified group name (for example, /MyDomain/MyGroup) is supplied, the --domain argument is ignored.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
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 show

The mccli group show command returns information for all groups.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--clients=BOOLEAN(false) If set true, all clients belonging to this group are listed.

--domain=STRING(/) Specifies Avamar server domain containing the group specified by the --name argument.

--name=STRING Specifies a group name. If not supplied, all groups with the specified domain are listed. STRING must be a valid group name.

TIP: Use mccli group show (page 139) to return a list of valid group names.

IMPORTANT: If a fully-qualified group name (for example, /MyDomain/MyGroup) is supplied, the --domain argument is ignored.

--recursive=BOOLEAN(false) If set true, command shows all groups in the domain and any sub-domains specified by the --domain=STRING argument.

If set false or not supplied, command only shows groups within the specified domain (sub-domains are not examined).

--verbose=BOOLEAN(false) If set true, detailed (verbose) information is returned.

If set false or not supplied, only the group names are returned.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.
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
------------- ------
Default Group /
```

This command returns detailed information for all groups currently in use:

```
mccli group show --verbose=true
```

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

```
Group          Domain   Enabled   Dataset       Schedule       Retention Policy Encryption Method
------------- ------ ------ -------------- ------------- -------------- -----------------Default Group / false Default Dataset Default Schedule Default Retention High
```

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             /
Enabled            false
Dataset            Default Dataset
Schedule           Default Schedule
Retention Policy    Default Retention
Encryption Method   High
```
The `mccli group show-client-members` command returns all clients belonging to the specified group.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--domain=STRING(//)  Specifies Avamar server domain containing the group specified by the --name argument.

--name=STRING  Specifies group from which to show client members. STRING must be a valid group name. This argument is required.

**TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

**IMPORTANT:** If a fully-qualified group name (for example, `/MyDomain/MyGroup`) is supplied, the --domain argument is ignored.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

```
22234  Group does not exist.
```

**Examples**

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

```
mccli group show-client-members --name='Default Group'
```

```
0,23000,CLI command completed successfully.
command completed successfully
Group   Client
---------  -----------------
Default Group /clients/MyClient
```
The `mccli group snapup` command initiates an on-demand group backup.

**IMPORTANT:** This command has been deprecated in favor of `mccli group backup` (page 128). Whenever possible, you should use `mccli group backup` instead of `mccli group snapup`.

### Synopsis

```
mckill group snapup [GLOBAL-OPTIONS] [--domain=STRING(/)] --name=STRING [DISPLAY-OPTIONS]
```

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

- `--domain=STRING(/)` Specifies Avamar server domain containing the group specified by the `--name` argument.
- `--name=STRING` Specifies which group to back up. STRING must be a valid group name. This argument is required.

**TIP:** Use `mccli group show` (page 139) to return a list of valid group names.

**IMPORTANT:** If a fully-qualified group name (for example, /MyDomain/MyGroup) is supplied, the `--domain` argument is ignored.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

### 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.
Help Resource Overview

The `mccli help` resource comprises a single command: `mccli help`, which returns online help for specified resource, then exits.

help

The `mccli help` command returns online help for specified resource, then exits.

Synopsis

```
mccli help {activity | backup | checkpoint | client | dataset
  | event | group | mcs | retention | schedule | snapup | version}
```
MCS Resource Overview

The `mccli mcs` resource is used to view status and control various administrator server features and functions. Available commands are:

- `mcs import` (page 145)
- `mcs list` (page 147)
- `mcs reboot-proxy` (page 148)
- `mcs resume-scheduler` (page 149)
- `mcs scheduler-status` (page 150)
- `mcs stop` (page 151)
- `mcs suspend-scheduler` (page 151)
- `mcs waitforflushcomplete` (page 152)
The `mcs import` command accepts an XML file created by the `group export` command (page 134), and imports the group and its policy objects (that is, dataset, schedule, retention policy) into the target Avamar server. The group name will remain the same but a new target domain can be specified.

**Synopsis**

```
mcli mcs import [GLOBAL-OPTIONS] --file=STRING
    [--force=Boolean(false)] --target-domain=STRING [--prefix=STRING]
    [DISPLAY-OPTIONS]
```

**Global Options**

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mcli` commands.

**Command Options**

`--file=STRING`  
Specifies the name of import XML file. This option is required.

`--force=Boolean(false)`  
XML files created by the export command are digitally signed. If an XML file is modified or manually created, digital signature verification will fail during import.  
If set true, the XML will be force imported without validating the digital signature.  
If set false, an error is returned, and the file will not be imported.

`--prefix=STRING`  
Specifies a prefix string that will be affixed to the beginning of the imported object names. This is done to avoid conflicts and distinguish imported objects from existing ones that might otherwise have similar names.

`--target-domain=STRING`  
Specifies the domain on the target server where the group and its policy objects (that is, dataset, schedule, retention policy) will reside.  
Original domain hierarchy for the imported objects will not be preserved. In other words, all imported objects will reside at the top-most level of this domain.  
The specified target domain must already exist on the target server. If not, this command will return an error.  
In order to avoid conflicts, if a policy object with the same name already exists in the target domain, the newly imported policy object will have _1, _2, _3 and so forth appended to it.
Display Options
Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22540  Invalid name character.
mcs list

The `mcs list` command accepts an XML file created by the `group export` command (page 134) and lists all groups and group policy objects (that is, dataset, schedule, retention policy) within that file.

**Synopsis**

```
mccli mcs list [GLOBAL-OPTIONS] --file=STRING [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--file=STRING
```

Specifies the name of import XML file. This option is required.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
mcs reboot-proxy

The `mcs` command reboots VMware proxy clients.

Synopsis

`mcs reboot-proxy [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]`

Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mcs` commands.

Command Options

- `--all=Boolean(false)` If supplied, reboot all known Avamar proxy clients.
- `--name=STRING` Specifies one or more proxies to reboot.

STRING must be a fully-qualified Avamar proxy client name.

Multiple `--name` options can be supplied on the same command line.

Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mcs` commands.
The `mccli mcs resume-scheduler` command turns on the MCS scheduler so that scheduled operations will occur at the scheduled times.

**Synopsis**

```
mccli mcs resume-scheduler [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options (page 24)* for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to *Display Options (page 25)* for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**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 `mcs scheduler-status` command returns status of the MCS scheduler.

**Synopsis**

`mccli mcs scheduler-status [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]`

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

**Example**

```
mcsi mcs scheduler-status
0,23000,CLI command completed successfully.
command completed successfully
Enabled
-------
true
```
**mcs stop**

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

**Synopsis**

`mccli mcs stop [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]`

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Example**

```
mccli mcs stop
0,23000,CLI command completed successfully.
command completed successfully
```

**mcs suspend-scheduler**

The `mccli mcs suspend-scheduler` command turns off the MCS scheduler so that no scheduled operations will be performed until such time as the scheduler is re-enabled using the `mccli mcs resume-scheduler` command (page 149).

**Synopsis**

`mccli mcs suspend-scheduler [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]`

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**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 [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]`

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
Plugin Resource Overview

The `mccli plugin` resource is used to manage client plugins. Available commands are:

- `plugin show` (page 153)
- `plugin update` (page 153)

plugin show

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

**Synopsis**

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

**Global Options**

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

plugin update

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

**Event Codes**

<table>
<thead>
<tr>
<th>Event Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21010</td>
<td>Plugin catalog dynamically updated.</td>
</tr>
</tbody>
</table>
Retention Resource Overview

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

- retention add (page 155)
- retention copy (page 157)
- retention delete (page 158)
- retention edit (page 159)
- retention show (page 161)

Advanced Retention Calendar Units

All advanced retention options (options beginning with "--adv") require a string value in the following format:

+nnnD  Specifies number (nnn) of calendar days (D).
  For example, specifying +5D will retain five daily backups; specifying +13D will retain thirteen daily backups.

+nnW  Specifies number (nn) of calendar weeks (W).
  For example, specifying +1W will retain one week of daily or weekly backups; specifying +13W will retain thirteen weeks of daily or weekly backups.

+nnM  Specifies number (nn) of calendar months (M).
  For example, specifying +1M will retain one month of daily, weekly or monthly backups; specifying +9M will retain nine months of daily, weekly or monthly backups.

+nnY  Specifies number (nn) of calendar years (Y).
  For example, specifying +1Y will retain one year of daily, weekly, monthly or yearly backups; specifying +5Y will retain five years of daily, weekly, monthly or yearly backups.
retention add

The `mccli retention add` command creates (adds) a new retention policy.

If no options are specified, the new retention policy will have 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, these settings are not enabled (basic retention settings are used instead of advanced retention settings).

Synopsis

```bash
mccli retention add [GLOBAL-OPTIONS] [--adv-daily=STRING(+1D)]
[–adv-monthly=STRING(+1M)] [–adv-weekly=STRING(+1W)]
[–adv-yearly=STRING(+1Y)] [–basic=STRING(NO_EXPIRATION)]
[–domain=STRING(/)] --name=STRING [--override=BOOLEAN(false)]
[DISPLAY-OPTIONS]
```

Global Options

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

- `--adv-daily=STRING(+1W)` Specifies daily advanced retention setting. STRING must be `+nnnD`, `+nnW`, `+nnM` or `+nnY`. Refer to `Advanced Retention Calendar Units` (page 154) for additional information.

- `--adv-monthly=STRING(+1Y)` Specifies monthly advanced retention setting. STRING must be `+nnnD`, `+nnW`, `+nnM` or `+nnY`. Refer to `Advanced Retention Calendar Units` (page 154) for additional information.

- `--adv-weekly=STRING(+1M)` Specifies weekly advanced retention setting. STRING must be `+nnnD`, `+nnW`, `+nnM` or `+nnY`. Refer to `Advanced Retention Calendar Units` (page 154) for additional information.

- `--adv-yearly=STRING(+1Y)` Specifies yearly advanced retention setting. STRING must be `+nnnD`, `+nnW`, `+nnM` or `+nnY`. Refer to `Advanced Retention Calendar Units` (page 154) for additional information.
--basic=STRING(NO_EXPIRATION) Specifies basic retention (assigns a fixed expiration date).

STRING must be one of the following:
- YYYY-MM-DD to specify an explicit date
- +nn{D|W|M|Y} to specify a duration from today (for example, +4W specifies 4 weeks from today)
- NO_EXPIRATION

--domain=STRING(/) Specifies Avamar server domain containing the retention policy specified by the --name argument.

--name=STRING Specifies new retention policy name. This argument is required.

**IMPORTANT:** If a fully-qualified retention policy name (for example, /MyDomain/MyRetentionPolicy) is supplied, the --domain argument is ignored.

--override=BOOLEAN(false) If set true, basic retention settings are overridden in favor of advanced retention settings.

**NOTE:** This option only applies to scheduled backups.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

### Event Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>22216</td>
<td>Retention Policy created.</td>
</tr>
<tr>
<td>23005</td>
<td>Invalid Retention Policy specified on the CLI.</td>
</tr>
<tr>
<td>23011</td>
<td>Retention Policy already exists.</td>
</tr>
</tbody>
</table>
The `mccli retention copy` command copies an existing retention policy, creating a new retention policy.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--domain=STRING(/)  Specifies Avamar server domain containing the retention policy specified by the --name argument.

--name=STRING      Specifies which retention policy to copy. STRING must be a valid retention policy name. This argument is required.
                     **TIP:** Use `mccli retention show` (page 161) to return a list of valid retention policy names.
                     **IMPORTANT:** If a fully-qualified retention policy name (for example, /MyDomain/MyRetentionPolicy) is supplied, the --domain argument is ignored.

--new-domain=STRING Specifies optional new Avamar server destination domain for the copied retention policy.

--new-name=STRING   Specifies new retention policy name. This argument is required.
```

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

- 22216 Retention Policy created.
- 22289 Retention policy does not exist.
The `mccli retention delete` command permanently removes (deletes) a retention policy from the Avamar server.

**IMPORTANT:** You cannot delete a retention policy if it is currently assigned to any client or group.

### Synopsis

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

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

- **--domain=STRING(/)** Specifies Avamar server domain containing the retention policy specified by the `--name` argument.

- **--name=STRING** Specifies which retention policy to delete. STRING must be a valid retention policy name. This argument is required.

  **TIP:** Use `mccli retention show` (page 161) to return a list of valid retention policy names.

  **IMPORTANT:** If a fully-qualified retention policy name (for example, `/MyDomain/MyRetentionPolicy`) is supplied, the `--domain` argument is ignored.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

### Event Codes

- 22218 Retention Policy deleted.
- 22289 Retention policy does not exist.
The `mccli retention edit` command modifies (edits) one or more properties for the specified retention policy.

**IMPORTANT:** There are no default settings for the `mccli retention edit` command. Entering this command without explicitly supplying options and values on the command line will have no effect on the specified resource.

**Synopsis**

```
mccli retention edit [GLOBAL-OPTIONS] [--adv-daily=STRING] 
[--adv-monthly=STRING] [--adv-weekly=STRING] [--adv-yearly=STRING] 
[--basic=STRING] [--domain=STRING(/)] --name=STRING 
[--override=BOOLEAN] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

**--adv-daily=STRING(+1W)** Specifies daily advanced retention setting.

STRING must be `+nnnD`, `+nnW`, `+nnM` or `+nnY`. Refer to *Advanced Retention Calendar Units* (page 154) for additional information.

**--adv-monthly=STRING(+1Y)** Specifies monthly advanced retention setting.

STRING must be `+nnnD`, `+nnW`, `+nnM` or `+nnY`. Refer to *Advanced Retention Calendar Units* (page 154) for additional information.

**--adv-weekly=STRING(+1M)** Specifies weekly advanced retention setting.

STRING must be `+nnnD`, `+nnW`, `+nnM` or `+nnY`. Refer to *Advanced Retention Calendar Units* (page 154) for additional information.

**--adv-yearly=STRING(+1Y)** Specifies yearly advanced retention setting.

STRING must be `+nnnD`, `+nnW`, `+nnM` or `+nnY`. Refer to *Advanced Retention Calendar Units* (page 154) for additional information.
--basic=STRING Specifies basic retention (assigns a fixed expiration date).
STRING must be one of the following:
• YYYY-MM-DD to specify an explicit date
• +nn{D|W|M|Y} to specify a duration from today
  (for example, +4W specifies 4 weeks from today)
• NO_EXPIRATION

--domain=STRING(/) Specifies Avamar server domain containing the retention policy specified by the --name argument.

--name=STRING Specifies which retention policy to modify (edit).
STRING must be a valid retention policy name. This argument is required.
TIP: Use mccli retention show (page 161) to return a list of valid retention policy names.
IMPORTANT: If a fully-qualified retention policy name (for example, /MyDomain/MyRetentionPolicy) is supplied, the --domain argument is ignored.

--override=BOOLEAN If set true, basic retention settings are overridden in favor of advanced retention settings.
NOTE: This option only applies to scheduled backups.

Display Options
Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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 within the specified Avamar domain.

Synopsis

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

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--domain=STRING(/) Specifies Avamar server domain containing the retention policy specified by the --name argument.

--name=STRING Specifies a retention policy.
If not supplied, information for all retention policies are returned.
IMPORTANT: If a fully-qualified retention policy name (for example, /MyDomain/MyRetentionPolicy) is supplied, the --domain argument is ignored.

--recursive=BOOLEAN(false) If set true, command shows all retention policies in the domain and any sub-domains specified by the --domain=STRING argument.
If set false or not supplied, command only shows retention policies within the specified domain (sub-domains are not examined).

--verbose=BOOLEAN(false) If set true, detailed (verbose) information is returned.

Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22289 Retention policy 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:

```
mcli retention show --name='Default Retention'
```

```
0,23000,CLI command completed successfully.
Attribute  Value
----------------------
Name        Default Retention
Basic Expiration Date 90 Days
Override    Yes
Keep days of daily 1
Keep weeks of weekly 1
Keep months of monthly 1
Keep years of yearly 7
```

This command lists verbose information for the Minimal Retention policy:

```
NOTE: It is normal for the Minimal Retention policy to return fewer properties than other retention policies.
```

```
mcli retention show --name='Minimal Retention'
```

```
0,23000,CLI command completed successfully.
Attribute  Value
----------------------
Name        Minimal Retention
Basic Expiration Date 90 Days
Disabled    No
```

This command lists verbose information for the End User On-Demand Retention policy:

```
NOTE: It is normal for the End User On Demand Retention policy to return fewer properties than other retention policies.
Schedule Resource Overview

The `mccli schedule` resource is used to create and manage schedules on the Avamar server. Available commands are:

- `schedule add` (page 164)
- `schedule copy` (page 167)
- `schedule delete` (page 168)
- `schedule edit` (page 169)
- `schedule show` (page 171)
- `schedule show-timezones` (page 173)
schedule add

The mccli schedule add command creates (adds) a new schedule.

Either --days, --hours, --nth-day or --week must be specified to indicate the recurrence. Specifying --days creates a weekly schedule; specifying --hours creates a daily schedule; specifying either --nth-day or --week creates a monthly schedule.

If --duration is not specified, the duration for a daily schedule will default to one hour and the duration for a weekly schedule will default to eight hours.

The --start option is not applicable for daily schedules.

If the --start option is not specified for a weekly schedule, the default start time will be 10 pm with respect to the time zone that the schedule was created.

If the --tz option is not specified, the time zone is defaults to the local time zone.

Synopsis

mccli schedule add [GLOBAL-OPTIONS]
{--days=STRING | --hours=STRING | --nth-day=STRING | --week=STRING}
[--desc=STRING] [--domain=STRING(/)] [--duration=STRING]
--name=STRING [--start=STRING] [--tz=STRING] [DISPLAY-OPTIONS]

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

Command Options

--days=STRING
| --hours=STRING
| --nth-day=STRING
| --week=STRING

Supplying --days creates a schedule that will 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).

Supplying --hours creates a schedule that will run daily on the specified hours of the day. STRING must be in the range 0-23; separate multiple values with commas (for example, 4,10,18).

Supplying --nth-day creates a schedule that will run on the specified calendar day of each month. STRING must be a valid numerical calendar date between 1 and 28, or last.

Supplying --week creates a schedule that will run on the specified week each month. STRING must be first, second, third, fourth or last.

--desc=STRING

Specifies a short text description of this schedule.

--domain=STRING(/)

Specifies Avamar server domain containing the schedule specified by the --name argument.
--duration=STRING Specifies 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 --duration is not supplied, then duration for a daily schedule defaults to one hour and the duration for a weekly schedule defaults to eight hours.

--name=STRING Specifies new schedule name. This argument is required.

**IMPORTANT:** If a fully-qualified schedule name (for example, /MyDomain/MySchedule) is supplied, the --domain argument is ignored.

--start=STRING Specifies earliest schedule start time. STRING must be in the format of HH:MM, where HH is hours and MM minutes.

--start is not applicable for daily schedules.

If --start is not specified for a weekly schedule, the default start value defaults to 10pm with respect to the time zone that the schedule was created in.

--tz=STRING Specifies time zone.

If this option is not supplied, the local time zone is used.

**TIP:** Use schedule show-timezones (page 173) to return a list of valid retention time zone names.

### Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

### Event Codes

22213 Schedule created.

22248 Schedule already exists.

22540 Invalid name character.

22542 Domain does not exist.
Examples

This example creates a monthly schedule (monthly_2nd_Friday) that runs on the second Friday of each month.

```
mccli schedule add --week=second --days=Friday --name=monthly_2nd_Friday
```

This example creates a monthly schedule (my_monthly_sch) that runs on the 28th calendar day of each month.

```
mccli schedule add --name=my_monthly_sch --nth-day=28
```
schedule copy

The `mccli schedule copy` command copies an existing schedule, creating a new schedule.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- `--domain=STRING(/)` Specifies Avamar server domain containing the schedule specified by the `--name` argument.
- `--name=STRING` Specifies which schedule to copy. STRING must be a valid schedule name. This argument is required.
  
  **TIP:** Use `mccli schedule show` (page 171) to return a list of valid schedule names.
  
  **IMPORTANT:** If a fully-qualified schedule name (for example, `/MyDomain/MySchedule`) is supplied, the `--domain` argument is ignored.
- `--new-domain=STRING` Specifies optional new Avamar server destination domain for the copied scheduled.
- `--new-name=STRING` Specifies new schedule name. This argument is required.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

- 22213 Schedule created.
- 22249 Schedule does not exist.
schedule delete

The `mccli schedule delete` command permanently removes (deletes) a schedule from the Avamar server.

**IMPORTANT:** You cannot delete a schedule if it is currently assigned to any group or event profile.

**Synopsis**

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

**Global Options**

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

**--domain=STRING(/)** Specifies Avamar server domain containing the schedule specified by the `--name` argument.

**--name=STRING** Specifies which schedule to delete. STRING must be a valid schedule name. This argument is required.

**TIP:** Use `mccli schedule show` (page 171) to return a list of valid schedule names.

**IMPORTANT:** If a fully-qualified schedule name (for example, `/MyDomain/MySchedule`) is supplied, the `--domain` argument is ignored.

**Display Options**

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**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.
The `mccli schedule edit` command modifies (edits) one or more properties for the specified schedule.

**IMPORTANT:** There are no default settings for the `mccli schedule edit` command. Entering this command without explicitly supplying options and values on the command line will have no effect on the specified resource.

**Synopsis**

```
mccli schedule edit [GLOBAL-OPTIONS] {--days=STRING | --hours=STRING} 
    [--desc=STRING] [--domain=STRING(/)] [--duration=STRING] 
    --name=STRING [--start=STRING] [--tz=STRING] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

**Command Options**

- **--days=STRING**
- **| --hours=STRING**
- **| --nth-day=STRING**
- **| --week=STRING**

Supplying `--days` creates a schedule that will 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).

Supplying `--hours` creates a schedule that will run daily on the specified hours of the day. STRING must be in the range 0-23; separate multiple values with commas (for example, 4,10,18).

Supplying `--nth-day` creates a schedule that will run on the specified calendar day of each month. STRING must be a valid numerical calendar date between 1 and 28, or last.

Supplying `--week` creates a schedule that will run on the specified week each month. STRING must be first, second, third, fourth or last.

- **--desc=STRING**

  Specifies a short text description of this schedule.

- **--domain=STRING(/)**

  Specifies Avamar server domain containing the schedule specified by the `--name` argument.

- **--duration=STRING**

  Specifies 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.
Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22214 Schedule modified.

22249 Schedule does not exist.

Example

This example edits an existing monthly schedule (my_monthly_sch) to run on the last calendar day of each month.

mccli schedule edit --name=my_monthly_sch --nth-day=last
schedule show

The `mccli schedule show` command lists all schedules and detailed schedule information.

Synopsis

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

Global Options

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

```
--domain=STRING(/)
```
Specifies Avamar server domain containing the schedule specified by the `--name` argument.

```
--name=STRING
```
Specifies a schedule name. If not supplied, all schedules are listed with summary information.

**IMPORTANT:** If a fully-qualified schedule name (for example, `/MyDomain/MySchedule`) is supplied, the `--domain` argument is ignored.

```
--recursive=BOOLEAN(false)
```
If set true, command shows all schedules in the domain and any sub-domains specified by the `--domain=STRING` argument.
If set false or not supplied, command only shows schedules within the specified domain (sub-domains are not examined).

```
--verbose=BOOLEAN(false)
```
If set true, detailed (verbose) schedule information is returned.
If set false or not supplied, only schedule names are returned.
If `--name` is supplied, detailed information for that specific schedule is returned.

Display Options

Refer to `Display Options` (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

Event Codes

```
22249 Schedule 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     2008-09-20 10:00 PM
Start Time        10:00 PM Pacific Daylight Time
Backup Window Duration (hours) 8.0
Repeat            Weekly
Days of Week      Sun, Mon, Tue, Wed, Thu, Fri, Sat
Delay Start Until N/A
End Policy        No End Date
Description       N/A
```

The following tables describes `schedule show` output attributes and possible values:

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>VALUES</th>
<th>NORMALIZED VALUES</th>
<th>DESCRIPTION OF NORMALIZED VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next Run Time</td>
<td>2008-09-13 09:00 AM</td>
<td>1189699200000</td>
<td>Unix precision time format, represented as UTC milliseconds from the epoch.</td>
</tr>
<tr>
<td>Start Time</td>
<td>9:00 AM America/Los_Angeles</td>
<td>61200000</td>
<td>Number of milliseconds into a day starting at midnight in local timezone.</td>
</tr>
<tr>
<td>Delay Start Until</td>
<td>2008-09-12 12:19 PM PDT</td>
<td>1189624773776</td>
<td>Unix precision time format, represented as UTC milliseconds from the epoch.</td>
</tr>
<tr>
<td>End Policy</td>
<td>2008-12-10 11:59 PM PST</td>
<td>1228982399999</td>
<td>Unix precision time format, represented as UTC milliseconds from the epoch.</td>
</tr>
</tbody>
</table>
schedule show-timezones

The `mccli schedule show-timezones` command lists all valid time zones, which can be supplied with other `mccli schedule` commands using the `--tz` option.

**Synopsis**

```
mccli schedule show-timezones [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
Server Resource Overview

The `mccli server` resource is used to monitor various features and functions of the Avamar server. Available commands are:

- `server show-prop` (page 175)
- `server show-services` (page 178)
- `server show-util` (page 179)
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. Refer to your Avamar System Administration Guide for a thorough discussion of server properties.

If `mccli server show-prop` is run without command options, a summary of server properties is shown.

Synopsis

```
mccli server show-prop [GLOBAL-OPTIONS]
   {--module=STRING | --node=STRING | --node=STRING --partition=STRING}
   --verbose [DISPLAY-OPTIONS]
```

Global Options

Refer to Global Options (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

- `--module=STRING`  Specifies the Avamar server module from which to show properties.
  STRING must be a valid Avamar server module designator. Module designators are typically single-digit integers beginning with zero. For example, 0, 1, 2, and so forth.
  A value of `ALL` returns status for all modules.

- `--node=STRING`  Specifies the Avamar server node from which to show properties.
  STRING must be a valid Avamar server numeric node designator in the format of MODULE.NODE. For example, 0.0 is typically the first storage node in a multi-node server.
  A value of `ALL` returns status for all nodes.

- `--partition=STRING`  Specifies the Avamar server data partition from which to show properties.
  STRING must be a valid Avamar server partition designator. Partition designators are typically single-digit integers beginning with zero. For example, 0, 1, 2, and so forth.
  A value of `ALL` returns status for all partitions.

- `--node=STRING` must also be supplied.

- `--verbose`  If supplied, maximum information is returned.
Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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
```

```
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 2008-10-01 04:00:28 PDT
Last validated checkpoint 2008-09-30 22:00:28 PDT
System Name avamar-1
HFSAAddr 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):

```
mcli server show-prop --node=0.0 --partition=0
```

```
Attribute Value
-------------------------------- -------------------------
Node ID 0.0
State ONLINE
Runlevel fullaccess
Accessmode mhpu+0hpu+0hpu
Port 26000
Dispatcher 27000
Server uptime 0 days 18h:08m
Server utilization 0.1%
Number of stripes 22
Server version 4.1.0-113
Version v2.6.9-67.ELsmp
Node uptime 6 days 00h:07m
Total capacity 86.0 TB
Capacity used 6.4 TB
Load average 0.22
CPU % 1.9%
Disk reads 0.11/sec
```
Disk writes 2.21/sec
Network reads 2K/sec
Network writes 2K/sec
IP address 10.6.249.38
Mac address 00:13:72:59:8F:AE
Number of partitions 4

The following tables describes `server show-prop` output attributes and possible values:

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>VALUES</th>
<th>NORMALIZED VALUES</th>
<th>DESCRIPTION OF NORMALIZED VALUES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes protected quota</td>
<td>None</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Configured</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>License expiration</td>
<td>Never</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Not Set</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Time since Server initialization</td>
<td>0 days 20h:06m</td>
<td>72389000</td>
<td>Number of milliseconds ago.</td>
</tr>
<tr>
<td>Last checkpoint</td>
<td>2008-09-12 11:00:17 PDT</td>
<td>1189620017000</td>
<td>Unix precision time format, represented as UTC milliseconds from the epoch.</td>
</tr>
<tr>
<td></td>
<td>No checkpoints performed</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Last validated checkpoint</td>
<td>2008-09-11 23:00:16 PDT</td>
<td>1189576816000</td>
<td>Unix precision time format, represented as UTC milliseconds from the epoch.</td>
</tr>
<tr>
<td></td>
<td>No validated checkpoints.</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>Server uptime</td>
<td>0 days 20h:05m</td>
<td>72331000</td>
<td>Number of milliseconds ago.</td>
</tr>
<tr>
<td>Node uptime</td>
<td>12 days 17h:30m</td>
<td>1099814000</td>
<td>Number of milliseconds ago.</td>
</tr>
</tbody>
</table>
server show-services

The `mccli server show-services` command returns detailed properties for Avamar server services, similar to what is shown in the Avamar Administrator Services Administration feature. Refer to your *Avamar System Administration Guide* for a thorough discussion of server properties.

**Synopsis**

```
mccli server show-services [GLOBAL-OPTIONS] [--service=STRING] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--service=STRING
```

Specifies an individual service for which to show information. STRING must be a valid Avamar service name (for example, replicate).

If not supplied, all services are shown.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
server show-util

The `mccli server show-util` command returns detailed utilization information from the Avamar server, similar to what is shown in the Avamar Administrator Server Monitor. Refer to your Avamar System Administration Guide for a thorough discussion of server properties.

If `mccli server show-util` is run without command options, a summary of server utilization is shown.

**Synopsis**

```
mccli server show-util [GLOBAL-OPTIONS]
    [--module=STRING | --node=STRING] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--module=STRING
```

Specifies the Avamar server module from which to show utilization information. STRING must be a valid Avamar server module designator. Module designators are typically single-digit integers beginning with zero. For example, 0, 1, 2, and so forth.

```
--node=STRING
```

Specifies the Avamar server node from which to show utilization information. STRING must be a valid Avamar server numeric node designator in the format of MODULE.NODE. For example, 0.1 is typically the first storage node in a multi-node server.

A value of ALL returns status for all nodes.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Notes**

The `--module=STRING` and `--node=STRING` options are mutually exclusive.
Examples

This command returns a summary of server utilization:

```
mccli server show-util
```

<table>
<thead>
<tr>
<th>Node</th>
<th>State</th>
<th>CPU-Load</th>
<th>CPU-User</th>
<th>CPU-Sys</th>
<th>Network-Ping</th>
<th>Network-In</th>
<th>Network-Out</th>
<th>Disk-Reads</th>
<th>Disk-Writes</th>
<th>Disk-Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>ONLINE</td>
<td>0.00</td>
<td>0.0%</td>
<td>0.1%</td>
<td>0.00</td>
<td>1K/sec</td>
<td>1K/sec</td>
<td>0/sec</td>
<td>0.61/sec</td>
<td>0.9%</td>
</tr>
<tr>
<td>0.1</td>
<td>ONLINE</td>
<td>0.00</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0.00</td>
<td>1K/sec</td>
<td>2K/sec</td>
<td>0/sec</td>
<td>0.49/sec</td>
<td>0.8%</td>
</tr>
<tr>
<td>0.2</td>
<td>ONLINE</td>
<td>0.00</td>
<td>0.1%</td>
<td>0.0%</td>
<td>0.00</td>
<td>2K/sec</td>
<td>3K/sec</td>
<td>0/sec</td>
<td>0.77/sec</td>
<td>0.7%</td>
</tr>
<tr>
<td>0.3</td>
<td>OFFLINE</td>
<td>Unknown</td>
<td>Unknown</td>
<td>Unknown</td>
<td>NaN</td>
<td>NaN</td>
<td>0/sec</td>
<td>0/sec</td>
<td>0.0%</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>ONLINE</td>
<td>0.03</td>
<td>0.0%</td>
<td>0.0%</td>
<td>0.00</td>
<td>2K/sec</td>
<td>3K/sec</td>
<td>0/sec</td>
<td>0.59/sec</td>
<td>0.9%</td>
</tr>
<tr>
<td>0.5</td>
<td>ONLINE</td>
<td>0.00</td>
<td>0.1%</td>
<td>0.0%</td>
<td>0.00</td>
<td>1K/sec</td>
<td>1K/sec</td>
<td>0/sec</td>
<td>0.56/sec</td>
<td>0.6%</td>
</tr>
<tr>
<td>0.6</td>
<td>ONLINE</td>
<td>0.00</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0.00</td>
<td>1K/sec</td>
<td>1K/sec</td>
<td>0/sec</td>
<td>0.62/sec</td>
<td>0.4%</td>
</tr>
<tr>
<td>0.7</td>
<td>ONLINE</td>
<td>0.00</td>
<td>0.0%</td>
<td>0.1%</td>
<td>0.04</td>
<td>1K/sec</td>
<td>1K/sec</td>
<td>0/sec</td>
<td>0.64/sec</td>
<td>0.7%</td>
</tr>
</tbody>
</table>
Snapup Resource Overview

The `mccli snapup` resource is used to restore files and folders (directories) to clients, as well as manage backups on the Avamar server. Available commands are:

- `snapup delete` (page 182)
- `snapup edit` (page 183)
- `snapup restore` (page 185)
- `snapup show` (page 187)
- `snapup validate` (page 189)

**IMPORTANT:** The snapup management resource has been deprecated in favor of the backup management resource (page 30).
The `mccli snapup delete` command permanently removes (deletes) the specified backup from the server.

**Synopsis**

```
mccli snapup delete [--created=STRING] [--domain=STRING(/)]
[--labelNum=INTEGER] [--name=STRING]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

**Command Options**

- **--created=STRING**
  Specifies date backup was created. STRING must be in the format of YYYY-MM-DD, where YYYY, MM and DD are calendar year, month and date, respectively.

- **[--domain=STRING(/)]**
  Specifies Avamar server domain containing the client specified by the **--name** argument.

- **--labelNum=INTEGER**
  Specifies label number of backup to delete.

- **--name=STRING**
  Specifies client from which this backup was originally taken. This argument is required. **IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the **--domain** argument is ignored.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

**Event Codes**

- 22236  Client does not exist.
- 22552  Backup does not exist.
**snapup edit**

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

- Directly specify a new expiration date
- Add additional days (d), weeks (w), months (m) or years that should be added to the existing backup expiration date
- Specify that the backup should never expire

**Synopsis**

```bash
mccli snapup edit [--created=STRING] [--domain=STRING(/)]
[--expiration=STRING] [--labelNum=INTEGER] [--name=STRING]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all *mccli* commands.

**Command Options**

- **--created=STRING**
  Specifies date backup was created. STRING must be in the format of YYYY-MM-DD, where YYYY, MM and DD are calendar year, month and date, respectively.

- **[--domain=STRING(/)]**
  Specifies Avamar server domain containing the client specified by the **--name** argument.

- **--expiration=STRING**
  Specifies new expiration date. STRING must be in one of the following formats:
  
  YYYY-MM-DD | +nn{D|W|M|Y} | NO_EXPIRATION

  Where:
  
  - YYYY-MM-DD is a specific year (YYYY), month (MM) and day (DD)
  - +nn{D|W|M|Y} is the number (nn) of additional days (d), weeks (w), months (m) or years that should be added to the existing backup expiration date
  - NO_EXPIRATION specifies that the backup should never expire.

- **--labelNum=INTEGER**
  Specifies label number of backup to change.

- **--name=STRING**
  Specifies client from which this backup was originally taken. This argument is required.

  **IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the **--domain** argument is ignored.
Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

Event Codes

22236  Client does not exist.

22552  Backup does not exist.
The `mccli snapup restore` command initiates a restore of files and folders (directories) to the specified client.

**TIP:** This command returns an activity ID, which can be passed to `mccli activity show` (page 28) in order to get status for this restore activity.

### Synopsis

```
mccli snapup restore [GLOBAL-OPTIONS]
    [--cmd=STRING [--cmd=STRING] ...] [--data=STRING]
    [--dest-dir=STRING] [--domain=STRING(/)] --labelNum=INTEGER
    --name=STRING --plugin=INTEGER [DISPLAY-OPTIONS]
```

### Global Options

Refer to `Global Options` (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

- **--cmd=STRING**
  Specifies one ore more optional plug-in commands. Multiple `--cmd` arguments can be supplied but each argument can only specify one plug-in command. For example, this is valid `--cmd` syntax:
  ```
  --cmd="verbose=5" --cmd="throttle=5"
  ```
  However, this is not valid `--cmd` syntax:
  ```
  --cmd="verbose=5 throttle=5"
  ```
  Refer to your `Avamar System Administration Guide` for a list of valid plug-in commands that can specified during a restore operation.

- **--data=STRING**
  Specifies a single optional target directory (folder) or file to restore.
  **IMPORTANT:** Each `--data` option can only specify one target directory (folder) or file to restore; use multiple `--data` options to restore multiple targets.

- **--dest-dir=STRING**
  Specifies optional destination directory for the restored files or folders (directories).

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

- **--labelNum=INTEGER**
  Specifies which backup to use for this restore operation. This argument is required.
  **TIP:** Use `mccli snapup show` (page 187) to return a list of valid backup label numbers.
--name=STRING  Specifies target client for this restore operation. STRING must be a valid client name. This argument is required.
TIP: Use mccli client show (page 70) to return a list of valid client names.
IMPORTANT: If a fully-qualified client name (for example, /clients/MyClient) is supplied, the --domain option is ignored.

--plugin=INTEGER  Specifies plug-in ID for this restore operation. This argument is required.
TIP: Use plugin show (page 153) to return a list of valid plug-in IDs.

Display Options
Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

Event Codes

22236  Client does not exist.
22312  Client restore scheduled.
23009  Invalid plugin specified on the CLI.
snapup show

The `mccli snapup show` command returns all backups currently stored on the Avamar server for the specified client.

**Synopsis**

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

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

```
--after=STRING

--before=STRING

--dir=STRING(/)

--domain=STRING(/)

--labelNum=INTEGER

--name=STRING

--recursive=BOOLEAN(false)

--verbose=BOOLEAN(false)
```

**--after=STRING**

Used with `--before` to specify a range of backups dates to return.

STRING must be in the format of YYYY-MM-DD, where YYYY is the four-digit year, MM is the two-digit calendar month and DD is the two-digit day of the month.

**--before=STRING**

Used with `--after` to specify a range of backups dates to return.

STRING must be in the format of YYYY-MM-DD, where YYYY is the four-digit year, MM is the two-digit calendar month and DD is the two-digit day of the month.

**--dir=STRING(/)**

Specifies initial directory of the backup from which to begin listing files and folders (directories).

**--domain=STRING(/)**

Specifies Avamar server domain containing the client specified by the `--name` argument.

**--labelNum=INTEGER**

Specifies a particular backup for which to show information.

**--name=STRING**

Specifies client from which to show backups. This argument is required.

**IMPORTANT:** If a fully-qualified client name (for example, /clients/MyClient) is supplied, the `--domain` argument is ignored.

**--recursive=BOOLEAN(false)**

If set true, backup files and folders (directories) are recursively shown.

**--verbose=BOOLEAN(false)**

If set true, detailed (verbose) information is returned.
**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

22236  Client does not exist.

22504  Failed to retrieve the backups for a client.

**Examples**

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

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

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

<table>
<thead>
<tr>
<th>Created</th>
<th>LabelNum</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006-01-31 15:51:30 PST</td>
<td>2</td>
<td>4767841280</td>
</tr>
<tr>
<td>2006-01-31 15:00:23 PST</td>
<td>1</td>
<td>4750878720</td>
</tr>
</tbody>
</table>

This command returns verbose file and directory information for a specific parent directory (C:/) within a specific client backup (the second backup taken from MyClient.example.com):

```
mccli snapup show --name=clients/MyClient.example.com --verbose --labelNum=2 --dir=C:/
```

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

<table>
<thead>
<tr>
<th>Type</th>
<th>Date</th>
<th>Size</th>
<th>Owner</th>
<th>Group</th>
<th>Permissions</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dir</td>
<td>2005-12-12 08:59:27</td>
<td>205,526,604</td>
<td>Administrators unknown</td>
<td>drwxrwxr-x</td>
<td>C:/Documents and Settings/</td>
<td></td>
</tr>
<tr>
<td>Dir</td>
<td>2006-01-17 11:33:54</td>
<td>1,601,595,257</td>
<td>Administrators unknown</td>
<td>dr-xr-x---</td>
<td>C:/Program Files/</td>
<td></td>
</tr>
<tr>
<td>Dir</td>
<td>2005-12-12 08:59:55</td>
<td>1,841</td>
<td>Administrator None</td>
<td>drwx------</td>
<td>C:/RECYCLER/</td>
<td></td>
</tr>
<tr>
<td>Dir</td>
<td>2006-01-31 15:45:30</td>
<td>20,480</td>
<td>Administrators unknown</td>
<td>d--x-------</td>
<td>C:/System Volume Information/</td>
<td></td>
</tr>
<tr>
<td>Dir</td>
<td>2006-01-17 15:18:22</td>
<td>211</td>
<td>Administrators unknown</td>
<td>drwxrwx---</td>
<td>C:/tmp/</td>
<td></td>
</tr>
<tr>
<td>Dir</td>
<td>2006-01-18 09:14:25</td>
<td>1,600,258,500</td>
<td>Administrators unknown</td>
<td>drwxrwx---</td>
<td>C:/WINDOWS/</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>2005-12-09 15:21:15</td>
<td>211</td>
<td>Administrators unknown</td>
<td>-r-xr-x---</td>
<td>C:/boot.ini</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>2005-12-12 08:56:49</td>
<td>4,128</td>
<td>Administrators unknown</td>
<td>-rwxrwx---</td>
<td>C:/INFCACHE.1</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>2005-08-11 15:15:00</td>
<td>0</td>
<td>Administrators unknown</td>
<td>-rwxrwx---</td>
<td>C:/IO.SYS</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>2005-08-11 15:15:00</td>
<td>0</td>
<td>Administrators unknown</td>
<td>-rwxrwx---</td>
<td>C:/MSDOS.SYS</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>2005-08-04 03:00:00</td>
<td>47,564</td>
<td>Administrators unknown</td>
<td>-r-xr-x---</td>
<td>C:/NTDETECT.COM</td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>2005-08-04 03:00:00</td>
<td>250,032</td>
<td>Administrators unknown</td>
<td>-r-xr-x---</td>
<td>C:/ntldr</td>
<td></td>
</tr>
</tbody>
</table>
The `mccli snapup validate` command initiates a validation of the specified backup.

**TIP:** This command returns an activity ID, which can be passed to `mccli activity show` (page 28) in order to get status for this validate activity.

### Synopsis

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

### Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

### Command Options

- **--cmd=STRING** Specifies one or more optional plug-in commands. Multiple `--cmd` arguments can be supplied but each argument can only specify one plug-in command. For example, this is valid `--cmd` syntax:
  ```
  --cmd="verbose=5" --cmd="throttle=5"
  ```
  However, this is not valid `--cmd` syntax:
  ```
  --cmd="verbose=5 throttle=5"
  ```
  Refer to your *Avamar System Administration Guide* for a list of valid plug-in commands that can specified during a validate operation.

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

- **--labelNum=INTEGER** Specifies which backup to use for this validate operation. This argument is required.

- **--name=STRING** Specifies target client for the validate operation. STRING must be a valid client name. This argument is required.
  **TIP:** Use `mccli client show` (page 70) to return a list of valid client names.
  **IMPORTANT:** If a fully-qualified client name (for example, `/clients/MyClient`) is supplied, the `--domain` argument is ignored.
--plugin=INTEGER Specifies plug-in ID for this restore operation. This argument is required.

**TIP:** Use `plugin show` (page 153) to return a list of valid plug-in IDs.

### Display Options
Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all *mccli* commands.

### Event Codes
- **22236**  Client does not exist.
- **22312**  Client restore scheduled.
- **23009**  Invalid plugin specified on the CLI.
User Resource Overview

The `mccli user` resource is used to create and manage backup user accounts on the Avamar server. Available commands are:

- `user add` (page 192)
- `user authenticate` (page 195)
- `user delete` (page 197)
- `user edit` (page 199)
- `user show` (page 202)
- `user show-auth` (page 203)

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 `mccli user show-auth` (page 203), `client show` (page 70) and `domain show` (page 110) commands, respectively.
The `mccli user add` command creates (adds) a new backup user account to the specified client or domain.

**Synopsis**

```
mccli user add [GLOBAL-OPTIONS] [--authenticator=STRING(Axion)]
   --client-domain=STRING [--client-name=STRING]
   --name=STRING [--password=STRING] [--password-confirm=STRING]
   --role=STRING [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all mccli commands.

**Command Options**

```
--authenticator=String(Axion) Specifies authentication system that will be used to grant 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 your Avamar server; STRING is case-insensitive.

**TIP:** Use `mccli user show-auth` (page 203) to return a list of valid authentication systems.

```
--client-domain=STRING Specifies domain. STRING must be a valid domain name. This argument is required.

If both `--client-domain` and `--client-name` are supplied, user is added to that client.

If only `--client-domain` is supplied, user is added to that domain.

**TIP:** Use `mccli domain show` (page 110) to return a list of valid domain names.

```
--client-name=STRING Specifies client to which the new user account will be added. STRING must be a valid client name.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

```
--name=STRING Specifies user ID or name for this new user account. This argument is required.
Display Options

Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.
**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, use of the `--privilege` option is deprecated in favor of `--role`.

Beginning with version 4.0, use of the `Snapup` and `SnapupRestore` roles are deprecated in favor of `Backup` and `BackupRestore`, respectively.

**IMPORTANT:** You cannot add new user accounts to the MC_RETIRED domain or to any clients within that domain.

**Examples**

This command adds a new user account (jsmith) with Administrator privileges to /clients/MyDomain:

```bash
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"
      directory="/clients/MyDomain" name="jsmith" privilege="Administrator"/>
```

This command adds a new user account (jsmith) with backup privileges to /clients/MyClient:

```bash
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"
      directory="/clients/MyClient" name="jsmith" privilege="Back up Only"/>
```
user authenticate

The `mccli user authenticate` command verifies user authentication settings. This is useful for testing user names, passwords and authentication system settings prior to actually creating a new user account.

Synopsis

```
mccli user authenticate [GLOBAL-OPTIONS]
   [--authenticator=STRING(Axion)] --client-domain=STRING
   [--client-name=STRING] --name=STRING --password=STRING
   [DISPLAY-OPTIONS]
```

Global Options

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

Command Options

- `--authenticator=STRING(Axion)` Specifies 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 your Avamar server; STRING is case-insensitive.
  
  **TIP:** Use `mccli user show-auth` (page 203) to return a list of valid authentication systems.

- `--client-domain=STRING` Specifies domain where client or user account resides. STRING must be a valid domain name. This argument is required.
  
  **TIP:** Use `mccli domain show` (page 110) to return a list of valid domain names.

- `--client-name=STRING` Specifies client name where user account resides. STRING must be a valid client name.
  
  **TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

- `--name=STRING` Specifies user ID or name to authenticate.

- `--password=STRING` Specifies password that will be used to authenticate this user. This argument is required.

Display Options

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
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 removes (deletes) a user from the Avamar server.

**Synopsis**

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

**Command Options**

---

**--authenticator=String**

Specifies authentication system associated with this user account.

STRING must be either Axion or the name of another valid authentication system that has been configured for use with your Avamar server; STRING is case-insensitive.

**TIP:** Use `mccli user show-auth` (page 203) to return a list of valid authentication systems.

---

**--client-domain=STRING**

Specifies domain where client or user account resides. STRING must be a valid domain name. This argument is required.

If both `--client-domain` and `--client-name` are supplied, user is deleted from that client.

If only `--client-domain` is supplied, user is deleted from that domain.

**TIP:** Use `mccli domain show` (page 110) to return a list of valid domain names.

---

**--client-name=STRING**

Specifies client name where user account resides. STRING must be a valid client name.

**TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

---

**--name=STRING**

Specifies user ID or name to delete.

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.
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.
The `mccli user edit` command modifies (edits) one or more properties for a specific user.

**Synopsis**

```
mccli user edit [GLOBAL-OPTIONS] --authenticator=STRING
                --client-domain=STRING [--client-name=STRING]
                --name=STRING [--client-name=STRING] [--role=STRING] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- `--authenticator=String` Specifies authentication system associated with this user account. STRING must be either Axion or the name of another valid authentication system that has been configured for use with your Avamar server; STRING is case-insensitive. This argument is required.
  
  **TIP:** Use `mccli user show-auth` (page 203) to return a list of valid authentication systems.

- `--client-domain=STRING(/)` Specifies domain where client or user account resides. STRING must be a valid domain name. This argument is required.

  If both `--client-domain` and `--client-name` are supplied, user resides on that client.

  If only `--client-domain` is supplied, user reside in that domain.

  **TIP:** Use `mccli domain show` (page 110) to return a list of valid domain names.

- `--client-name=STRING` Specifies client name where user account resides. STRING must be a valid client name.

  **TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

- `--name=STRING` Specifies user ID or name for the user account. STRING must be a valid user name. This argument is required.

  **TIP:** Use `mccli user show` (page 202) to return a list of valid domain names.

- `--password=STRING` Specifies user password.
```
--password-confirm=STRING  Specifies password confirmation string.  
If --password=STRING is supplied, STRING must be the same value as  
--password=STRING. This is intended to prevent a single incorrect entry of the password.

--privilege=STRING  Specifies user privilege level.  
Beginning with version 4.0, use of this option is deprecated in favor of --role.

--role=STRING  Specifies user role.  
STRING must be one of the following:
  • ActivityOperator
  • Administrator
  • Auditor
  • Backup
  • BackupOperator
  • BackupRestore
  • BackupRestoreOperator
  • Restore
  • RestoreIgnoreFilePermissions
  • RestoreOperator

IMPORTANT: The RestoreIgnoreFilePermissions role is only allowed if you are using an external authentication system.
Refer to your Avamar System Administration Guide for additional information about roles.

Display Options
Refer to Display Options (page 25) for a list of DISPLAY-OPTIONS common to all mccli commands.

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, use of the `--privilege` option is deprecated in favor of `--role`.

Beginning with version 4.0, use of the Snapup and SnapupRestore roles are deprecated in favor of Backup and BackupRestore, respectively.

**IMPORTANT:** You cannot edit user accounts within the MC_RETIRED domain.
user show

The `mccli user show` command lists (shows) properties for users on an access list for a specified client or domain.

**Synopsis**

```
mccli user show [GLOBAL-OPTIONS] [--authenticator=String(Axion)]
[--client-domain=STRING(/)] [--client-name=STRING] [--name=STRING]
[--recursive=BOOLEAN(false)] [DISPLAY-OPTIONS]
```

**Global Options**

Refer to *Global Options* (page 24) for a list of GLOBAL-OPTIONS common to all `mccli` commands.

**Command Options**

- `--authenticator=String(Axion)` Specifies 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 your Avamar server; STRING is case-insensitive.

  **TIP:** Use `mccli user show-auth` (page 203) to return a list of valid authentication systems.

- `--client-domain=STRING(/)` Specifies domain where client resides.

- `--client-name=STRING` Specifies client name to examine for user access. STRING must be a valid client name.

  **TIP:** Use `mccli client show` (page 70) to return a list of valid client names.

- `--name=STRING` Specifies user ID or name.

- `--recursive=BOOLEAN(false)` If set true, command examines all domains within the domain and any sub-domains. If set false or not supplied, command only examines the specified domain (sub-domains are not examined).

**Display Options**

Refer to *Display Options* (page 25) for a list of DISPLAY-OPTIONS common to all `mccli` commands.

**Event Codes**

- 22542 Domain does not exist.
- 22547 User does not exist.
user show-auth

The \texttt{mccli user show-auth} command lists (shows) all authentication systems configured for use with your Avamar server.

\textbf{Synopsis}

\texttt{mccli user show-auth [GLOBAL-OPTIONS] [DISPLAY-OPTIONS]}

\textbf{Global Options}

Refer to \textit{Global Options} (page 24) for a list of GLOBAL-OPTIONS common to all \texttt{mccli} commands.

\textbf{Display Options}

Refer to \textit{Display Options} (page 25) for a list of DISPLAY-OPTIONS common to all \texttt{mccli} commands.
Version Resource Overview

The `mccli version` resource comprises a single command: `mccli version show`, which returns the version of `mccli` currently installed, then exits.

**version show**

The `mccli version show` command returns the version of `mccli` currently installed, then exits.

**Synopsis**

`mccli version show`