



ElectricAccelerator cmtool Reference and Users Guide

for version 7.0

Electric Cloud, Inc.
676 W. Maude Avenue
Sunnyvale, CA 94085
www.electric-cloud.com

Copyright © 2002 - 2013 Electric Cloud, Inc. All rights reserved.

Published 2/28/2013

Electric Cloud® believes the information in this publication is accurate as of its publication date. The information is subject to change without notice and does not represent a commitment from the vendor.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” ELECTRIC CLOUD, INCORPORATED MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any ELECTRIC CLOUD software described in this publication requires an applicable software license.

Copyright protection includes all forms and matters of copyrightable material and information now allowed by statutory or judicial law or hereinafter granted, including without limitation, material generated from software programs displayed on the screen such as icons, screen display appearance, and so on.

The software and/or databases described in this document are furnished under a license agreement or nondisclosure agreement. The software and/or databases may be used or copied only in accordance with terms of the agreement. It is against the law to copy the software on any medium except as specifically allowed in the license or nondisclosure agreement.

Trademarks

Electric Cloud, ElectricAccelerator, ElectricCommander, ElectricDeploy, ElectricInsight, and Electric Make are registered trademarks or trademarks of Electric Cloud, Incorporated.

Electric Cloud products—ElectricAccelerator, ElectricCommander, ElectricDeploy, ElectricInsight, and Electric Make—are commonly referred to by their “short names”—Accelerator, Commander, Deploy, Insight, and eMake—throughout various types of Electric Cloud product-specific documentation.

Other product names mentioned in this guide may be trademarks or registered trademarks of their respective owners and are hereby acknowledged.

Contents

Chapter 1: Introduction	1-1
cmtool: The ElectricAccelerator Command-line Tool	1-1
Logging In	1-1
Using cmtool	1-1
cmtool Commands	1-2
Examples for MySQL	1-2
Global Arguments (Optional)	1-2
Using runAgentCmd	1-3
Chapter 2: cmtool Command Reference	2-1
Agent Management	2-1
Build Management	2-2
Cluster Management	2-3
Reporting	2-4
User Management	2-4
cmtool Commands in Full Detail (by Group)	2-5
Agent Management	2-5
Build Management	2-19
Cluster Management	2-35
Reporting	2-49
User Management	2-52

Chapter 1: Introduction

cmtool: The ElectricAccelerator Command-line Tool

cmtool provides access to the Cluster Manager through a command-line interface instead of using the web interface. With **cmtool**, you can write Perl scripts to access Cluster Manager information or manage builds. Almost all ElectricAccelerator® operations and tasks can be implemented with **cmtool**—with the exception of a few reports that are generated only from the web interface.

cmtool is used primarily for build and agent management, including commands for build class management, agent testing, and adding comments automatically.

Logging In

If you use **cmtool** outside of a job, you *must* invoke the **cmtool login** command to log in to the server. After logging in, **cmtool** saves information about the login session for use in future **cmtool** invocations. If you run **cmtool** as part of an ElectricAccelerator job, you do not need to log in because `--cmtool` uses the login session (and credentials) for that job.

To log in to cmtool:

```
cmtool login <username> <password>
```

To specify a session file, use the `--sessionFile=<fileName>` option, so you can use the same session for subsequent **cmtool** invocations.

Using cmtool

An invocation of **cmtool** identifies the Cluster Manager to contact, using the `--server` command-line option, followed by a list of commands to execute. Certain commands may have optional or required arguments.

For example, the following invocation receives all build requests that ran fewer than 10 jobs and orders the list [that ran the build] by host name.

```
cmtool --server easerver getBuilds-filter "job_count <10" --order host_name
```

General syntax for **cmtool** command usage:

```
cmtool [optional global argument(s)] <command> <required arguments> [optional arguments]
```

Return Codes

- 0** = success (the command was correct; if no data meets the criteria, return is still 0)
- 1** = failure (command was invalid)

cmtool Commands

Commands are grouped into common usage sections for your convenience. All cmtool commands are listed in alphabetical order within each group. Each command is shown with its description.

- Agent Management
- Build Management
- Cluster Management
- Reporting
- User Management

See the next chapter, "[cmtool Command Reference](#)" on page 2-1 which begins with a grouped list of commands.

Click a command name to go to an expanded version of the command, including its required arguments, optional arguments, available fields (where applicable), and a usage example.

When you receive a “returned response”—most cmtool command responses are returned in XML format.

Another source of cmtool ‘help’ is its own online help, for example:

- `cmtool --help-commands` prints a list of all commands with a short description.
- `cmtool --help-fields` displays a list of fields for a command.

Examples for MySQL

Note that all database examples provided in this guide are specific to MySQL. If you use Oracle or MSSQL, use syntax that is appropriate for your respective database.

Global Arguments (Optional)

Global arguments supply general information quickly, including cmtool online help.

Note: Global arguments support using the “=” sign character.

Global Arguments	Description
<code>--help=[command]</code> <code>--help [command]</code>	Prints this message and exits. If a command is specified, prints the help text for that command.
<code>--help-commands</code>	Prints the list of available commands with a short description.
<code>--help-fields=<command></code> <code>--help-fields <command></code>	Displays a list of fields for a command—requires the <code><command></code> argument.
<code>--version</code>	Prints cmtool version number.
<code>--server=<hostname></code> <code>--server <hostname></code>	ElectricAccelerator server address. Defaults to the <code>ACCELERATOR_SERVER</code> environment variable. If this variable does not exist, default is to the localhost.
<code>--port=<port></code> <code>--port <port></code>	HTTP listener port on the ElectricAccelerator server. Defaults to port 8030.

Global Arguments	Description
<pre>--securePort=<securePort> --securePort <securePort></pre>	HTTPS listener port on the ElectricAccelerator server. Defaults to port 8031.
<pre>--secure</pre>	Uses HTTPS to communicate with the ElectricAccelerator server.
<pre>--timeout=<s> --timeout <s></pre>	cmtool waits for a response from the server for a specified amount of time. Timeout for server communication defaults to 180 seconds (3 minutes) if no other time is specified. After the timeout, cmtool exits but the server will continue to process the command.
<pre>--output=<style> --output <style></pre>	Set output style—default is to XML: 'xml' for an XML document 'csv' for comma separated values 'simple' for no formatting 'silent' for no output
<pre>--fields=<list> --fields <list></pre>	<i>List</i> is a comma separated list of fields to emit when using 'csv' or 'simple' output styles. Default is to <i>all</i> fields.
<pre>--sessionFile=<path> --sessionFile <path></pre>	Overrides the location where session information will be stored.

Using runAgentCmd

IMPORTANT: Exercise caution when using the `runAgentCmd` command. Electric Cloud recommends using this command for documented scenarios only or under the direction of Electric Cloud Technical Support.

The `runAgentCmd` command enables you to run agent commands against the cluster.

Use this format: `cmtool --cm=<cm> runAgentCmd "agent command to run"`

where `<cm>` is the IP address or name of your Cluster Manager.

Some of the possible reasons for using `runAgentCmd` include:

- setting agent-side breakpoints (see the Using Breakpoints topic in online help)
- configuring agent log rotation (see the Installation and Configuration Guide)
- getting and setting agent and EFS debug levels (Knowledge Base article KB-00020)
- configuring the stalled job killer (Knowledge Base article KB-00031)
- troubleshooting builds that appear to hang (Knowledge Base article KB-00036)

Chapter 2: cmtool Command Reference

Click on a command to go to more information for that command.

Agent Management

Commands	Description
changeAgentsEnabled	Changes the agent enabled status of one or more agents.
createAgentComment	Creates a new agent comment.
createResource	Creates a new resource definition.
createResourceComment	Creates a new resource comment.
deleteAgentComment	Deletes an agent comment.
deleteAgents	Deletes an agent, including all dependent records.
deleteResource	Deletes a resource definition.
deleteResources	Deletes multiple resource definitions.
deleteResourceComment	Deletes a resource comment.
getAgentComments	Retrieves a list of related agent comments.
getAgentPerformance	Retrieves the performance log of one or more agents.
getAgents	Retrieves a list of agents.
getAgentStatus	Retrieves the state of one or more agents.
getLsfInformation	Retrieves current information about the LSF interface.
getLsfJobs	Retrieves information about all jobs submitted to LSF.
getResource	Finds a resource with full detail by the resource ID number.

Commands	Description
<code>getResources</code>	Retrieves a list of all resources.
<code>modifyAgentComment</code>	Modifies an agent comment.
<code>modifyResource</code>	Modifies a resource definition.
<code>modifyResourceComment</code>	Modifies a resource comment.
<code>setAgentDebug</code>	Sets the agent debug level.

Build Management

Commands	Description
<code>createBuildClass</code>	Creates a build class.
<code>createBuildClassComment</code>	Creates a new build class comment.
<code>createBuildComment</code>	Creates a new build comment.
<code>deleteBuild</code>	Deletes a build, including all dependent records.
<code>deleteBuilds</code>	Deletes a set of builds, including all dependent records.
<code>deleteBuildClass</code>	Deletes a build class, including all dependent records.
<code>deleteBuildClasses</code>	Deletes a set of build classes, including all dependent records.
<code>deleteBuildClassComment</code>	Deletes a build class comment.
<code>deleteBuildComment</code>	Deletes a build comment.
<code>getBuild</code>	Finds a build with full detail by the build's ID number.
<code>getBuilds</code>	Retrieves a list of builds.
<code>getBuildComments</code>	Retrieves a list of related build comments.
<code>getBuildClass</code>	Finds a build class with full detail by its ID number.
<code>getBuildClasses</code>	Retrieves a list of build classes with limited detail.
<code>getBuildClassComments</code>	Retrieves a list of related build class comments.
<code>getBuildUserStats</code>	Retrieves a list of user build statistics, grouped by user name, IP address, or host name.
<code>modifyBuild</code>	Modifies a build.

Commands	Description
<code>modifyBuildClass</code>	Modifies a build class.
<code>modifyBuildClassComment</code>	Modifies a build class comment.
<code>modifyBuildComment</code>	Modifies a build comment.
<code>setDatabaseConfiguration</code>	Modifies database configuration settings.
<code>stopBuild</code>	Stops a build.

Cluster Management

Commands	Description
<code>createServerComment</code>	Creates a new server comment.
<code>deleteLicense</code>	Deletes a license.
<code>deleteMessage</code>	Deletes a specific message, including all dependent records.
<code>deleteMessages</code>	Deletes a set of messages, including all dependent records.
<code>deleteServerComment</code>	Deletes a server comment.
<code>exportData</code>	Exports Cluster Manager data to a file.
<code>getLicense</code>	Retrieves information about one license.
<code>getLicenses</code>	Retrieves all license data.
<code>getMessage</code>	Retrieves a particular message.
<code>getMessages</code>	Retrieves a list of messages.
<code>getResourceStats</code>	Retrieves resource usage statistics.
<code>getServer</code>	Retrieves the server configuration.
<code>getServerComments</code>	Retrieves a list of related server comments.
<code>getVersion</code>	Retrieves server version information.
<code>importData</code>	Imports Cluster Manager data from a file.
<code>importLicenseData</code>	Imports one or more licenses.
<code>logMessage</code>	Creates a custom message on the Cluster Manager Messages page.
<code>modifyServer</code>	Modifies the server configuration.

Commands	Description
<code>modifyServerComment</code>	Modifies a server comment.
<code>shutdownServer</code>	Stops the server.
<code>testAgents</code>	Tests one or more agents.

Reporting

Commands	Description
<code>createFilter</code>	Creates a named filter for a specific table.
<code>deleteFilter</code>	Deletes a named filter for a specific table.
<code>getCurrentServerLoad</code>	Retrieves information about the current resource load.
<code>getFilter</code>	Retrieves a named filter for a specific table.
<code>getFilters</code>	Retrieves a list of saved filters for the current user.
<code>modifyFilter</code>	Updates a named filter for a specific table.

User Management

Commands	Description
<code>addGroupMember</code>	Adds a user name to the member list for a specific group.
<code>changeOwnUser</code>	Modifies the currently logged-in user.
<code>createGroup</code>	Creates a new local group.
<code>createUser</code>	Creates a new local user.
<code>deleteGroup</code>	Deletes a local group.
<code>deleteUser</code>	Deletes a local user.
<code>getAccessEntries</code>	Retrieves permissions for all users and groups that were granted server access.
<code>getGroupMembers</code>	Retrieves a list of users in a specific group.
<code>getGroups</code>	Finds all groups known to the server. If "local" is true, returns local groups only. Returns a list of groupInfo elements.

Commands	Description
<code>getEffectivePermissions</code>	Retrieves the permissions for the currently logged-in user.
<code>getPermissions</code>	Retrieves permissions for a particular user or group.
<code>getUser</code>	Finds a specific user known to the server.
<code>getUsers</code>	Finds all users known to the server. If "local" is true, returns local users only.
<code>getUserSettings</code>	Retrieves settings for the currently logged-in user.
<code>login</code>	Logs in to the client with the appropriate credentials.
<code>logout</code>	Logs out of the client session.
<code>modifyGroup</code>	Modifies a local group.
<code>modifyUser</code>	Modifies a local user.
<code>removeGroupMember</code>	Deletes a user from a specific group member list.
<code>setBuildEndNotification</code>	Enables/disables notification when builds of this class end for the currently logged-in user.
<code>setPermissions</code>	Creates or modifies permissions for a user or group. The permissions are a space-separated list of permission names.
<code>setUserSettings</code>	Updates settings for the currently logged-in user.

cmtool Commands in Full Detail (by Group)

Agent Management

changeAgentsEnabled	
Description	Changes the agent enabled status of one or more agents.
Required arguments	<enabled> - Values are true/false

changeAgentsEnabled	
Optional arguments	<pre>--agentId <unique, internal number that can change; assigned by the Cluster Manager></pre> <pre>--agentName <name defined by the host where the agent resides [numbers and/or letters]></pre> <pre>--filter <SQL query used to limit the result set for a request. For a list of possible SQL values, see the getAgents command.></pre> <p>Note: There is a syntax difference between MySQL and Oracle/MSSQL for enclosing criteria when using <code>--filter</code> for specific strings—for MySQL, use double quotes; for Oracle/MSSQL, use single quotes.</p> <p>Note: If no agent name, agent ID, or filter is specified, all agents are changed.</p>
Usage example	<pre>cmtool changeAgentsEnabled false</pre> <p>Disables all agents in the cluster.</p> <pre>cmtool changeAgentsEnabled true --agentName linagent1</pre> <p>Enables the agent named "linagent1".</p> <pre>cmtool changeAgentsEnabled true --filter "agent_name LIKE 'winbuild1-%'"</pre> <p>This enables all agents with a name that begins with "winbuild1-".</p>

createAgentComment	
Description	Creates a new agent comment.
Required arguments	<p><text> - The text of the item.</p> <p>Either <code>agentId</code> or <code>agentName</code> must be specified.</p>
Optional arguments	<pre>--agentId <unique, internal number that can change; assigned by the Cluster Manager></pre> <pre>--agentName <name defined by the host where the agent resides [numbers and/or letters]></pre>
Usage example	<pre>cmtool createAgentComment --agentName linagent "Agent has been running great"</pre> <p>Creates a comment for an agent named "linagent".</p>

createResource	
Description	Creates a new resource definition. After creating a resource, ensure the server is configured to support resource management. You can use the "modifyServer" on page 2-47 command to enable resource management.
Required arguments	<p><i><resourceName></i> - This name is used on the eMake parameter: <code>--emake-resource</code>, and can be specified in a build class. It is used in the <code>ea_resource</code> table and also matches the resource requirement string for eMake.</p> <p><i><hostMasks></i> - This is a quote-enclosed, semi-colon delimited list of host name masks, used to identify the list of hosts that support a resource. "*" is the wildcard character.</p>
Optional arguments	<code>--description <a quote-enclosed text description for your reference only></code>
Usage example	<pre>cmtool createResource R29 "rs*; rt*" --description "rs or rt hosts"</pre> <p>Creates a new resource named R29 that only uses hosts whose names start with 'rs' or 'rt'.</p>

createResourceComment	
Description	Creates a new resource comment.
Required arguments	<p><i><resourceId></i> - A unique number that identifies each resource.</p> <p><i><text></i> - The text of the item.</p>
Optional arguments	None
Usage example	<pre>cmtool createResourceComment 2 "This resource identifies production servers"</pre> <p>Creates a comment for resource 2.</p>

deleteAgentComment	
Description	Deletes an agent comment.
Required arguments	<p><i><commentId></i> - The unique key that identifies a comment. Use <code>getAgentComments</code> to get a list of <code>commentId</code> numbers.</p> <p>Either <code>agentId</code> or <code>agentName</code> must be specified.</p>

deleteAgentComment	
Optional arguments	<pre>--agentId <unique, internal number that can change; assigned by the Cluster Manager></pre> <pre>--agentName <name defined by the host where the agent resides [numbers and/or letters]></pre>
Usage example	<pre>cmtool deleteAgentComment 1008 --agentId 14</pre> <p>Deletes comment 1008 from agent 14 (14 is the Cluster Manager internal ID for the agent). To find out what the appropriate comment ID is, use the <code>getAgentComments</code> command, which will list the comments attached to a particular agent.</p>

deleteAgents	
Description	Deletes one or more agents, including all dependent records.
Required arguments	None
Optional arguments	<pre>--agentId <unique, internal number that can change; assigned by the Cluster Manager></pre> <pre>--agentName <name defined by the host where the agent resides [numbers and/or letters]></pre> <pre>--filter <SQL query used to limit the result set for a request. For a list of possible SQL values, see the getAgents command.></pre>
Usage example	<pre>cmtool deleteAgents --agentName winbuild1</pre> <p>Deletes agent “winbuild1” and all associated comments.</p>

deleteResource	
Description	Deletes a resource definition.
Required arguments	<pre><resourceId></pre> <p>- A unique number that identifies each resource. Use the <code>getResources</code> command to get a list of resource IDs.</p>
Optional arguments	None
Usage example	<pre>cmtool deleteResource 3</pre> <p>Deletes the resource definition for resource 3.</p>

deleteResources	
Description	Deletes multiple resource definitions.
Required arguments	None
Optional arguments	<code>--filter</code> <i><SQL query used to limit the result set for a request. For a list of possible SQL values, see the <code>getResources</code> command.></i>
Usage example	<code>cmtool deleteResources</code> Deletes all resource definitions.

deleteResourceComment	
Description	Deletes a resource comment.
Required arguments	<i><resourceId></i> - A unique number that identifies each resource. <i><commentId></i> - The unique key that identifies a comment. Use the <code>getResourceComments</code> command to get a list of comment IDs.
Optional arguments	None
Usage example	<code>cmtool deleteResourceComment 3 49</code> Deletes comment 49 from resource 3.

getAgentComments	
Description	Retrieves a list of related agent comments, or a specific comment (by using the <code>--commentId</code> option).
Required arguments	None
Optional arguments	<code>--agentId</code> <i><unique, internal number that can change; assigned by the Cluster Manager></i> <code>--agentName</code> <i><name defined by the host where the agent resides [numbers and/or letters]></i> <code>--commentId</code> <i><unique key that identifies a comment></i> Note: Either <code>agentId</code> or <code>agentName</code> must be specified.

getAgentComments		
Available fields	Field output name	Description
	commentId	The unique key that identifies a comment.
	createTime	The time when the item was created.
	lastModifiedBy	The user who last modified the item.
	modifyTime	The time when the item was last modified.
	text	The text of the item.
Usage example	<pre>cmtool getAgentComments --agentName ahost-3</pre> Retrieves all comments for agent "ahost-3".	

getAgentPerformance		
Description	Retrieves the performance log of one or more agents.	
Required arguments	None	
Optional arguments	<pre>--agentId <unique, internal number that can change; assigned by the Cluster Manager> --agentName <name defined by the host where the agent resides [numbers and/or letters]> --agents <this can be a list of agents whose performance you want to see> --buildId <this further restricts the returned agents to those running a specific build ID> --status <true/false> <choose active or inactive agents (ok or not ok)> --enabled <true/false> <choose enabled or disabled agents only></pre>	
Available fields	Field output name	Description
	agentName	This is the name of the agent as it appears on the web page (product UI).
	result	This is the performance information of the agent.
Usage example	<pre>cmtool getAgentPerformance --agentName SOL1-1</pre> Returns the performance log of the agent named "SOL1-1".	

getAgents	
Description	Retrieves a list of agents.
Required arguments	None
Optional arguments	<p><code>--agentId</code> <<i>unique, internal number that can change; assigned by the Cluster Manager</i>></p> <p><code>--agentName</code> <<i>name defined by the host where the agent resides [numbers and/or letters]</i>></p> <p><code>--filter</code> <<i>SQL query used to limit the result set for a request. See the possible values below.</i>></p> <p>Note: There is a syntax difference between MySQL and Oracle/MSSQL for enclosing criteria when using <code>--filter</code> for specific strings—for MySQL, use double quotes; for Oracle/MSSQL, use single quotes.</p> <p><code>--maxResults</code> <<i>maximum number of elements to run from a query</i>></p> <p><code>--firstResult</code> <<i>starting index for the query result set</i>></p> <p>Note: <code>--firstResult</code> takes values beginning with “0”. A negative value indicates a record starting from the end of the set, counting backwards, so “-1” is the last record, “-2” is the next to last, and so on.</p> <p><code>--order</code> <<i>SQL order by clause, used to specify ordering for the query result set.</i>></p> <p><code>--profile</code> <<i>level of detail to return from a query. Possible values are details (all information) and info (reduced information set).</i>></p>

getAgents			
Available fields	Field output name	Description	SQL query name for --filter and --order
	a2aPort	The agent to agent protocol communication port.	a2a_port
	agentId	A unique, internal number assigned to each agent by the Cluster Manager; this number can change.	id
	agentName	A name defined by the host where the agent resides [numbers and/or letters].	agent_name
	agentVersion	The agent version string.	agent_version
	availableResults	This is a count of 'max' or 'first' results if --maxResults or --firstResult is specified.	N/A
	buildId	A unique number assigned by the Cluster Manager for each build.	current_build_id
	buildName	The build name that is the expanded build class tag.	N/A
	consolePort	The agent console port.	console_port
	efsVersion	The EFS version string.	efs_version
	enabled	The flag indicating if an agent is enabled or not.	enabled
	errorCount	The number of internal agent errors.	error_count
	hostName	The name of the machine where Electric Make was invoked.	host_name
	inPenaltyBox	A flag indicating Electric Make had a recent problem with this agent.	N/A
	ipAddress	The agent IP address.	ip_address
	lastErrorTime	The last time the agent experienced an error.	last_error_time

getAgents			
	lastPingTime	The last time the agent was pinged to determine its status.	last_ping_time
	platform	The operating system being used or supported. If an OS is specified for a build class, builds from other operating systems cannot affiliate themselves with this class.	platform
	port	The agent protocol communication port.	port
	restartCount	The number of agent restarts.	restart_count
	status	The agent status. 1= OK, but anything else is an error code.	status
	statusDetail	If the last status update resulted in an error, it contains the error string (or the "OK" string if no error occurred).	status_detail
	webPort	The agent web server port.	web_port
Usage example	<pre>cmtool getAgents --filter "agent_name like '%SOL%'"</pre> Retrieves a list of all agents whose names start with "SOL".		

getAgentStatus	
Description	Retrieves the state of one or more agents. By default, only active agents are returned. (Use <code>--status false</code> to list inactive agents.)
Required arguments	None

getAgentStatus		
Optional arguments	<pre>--agentId <unique, internal number that can change; assigned by the Cluster Manager> --agentName <name defined by the host where the agent resides [numbers and/or letters]> --agents <this can be a list of agents whose status you want to see> --buildId <this further restricts the returned agents to those running a specific build ID> --status <true/false> <choose active or inactive agents (ok or not ok)> --enabled <true/false> <choose enabled or disabled agents only></pre>	
Available fields	Field output name	Description
	agentName	This is the name of the agent as it appears on the web page (product UI).
	result	This is the text string that describes the current state of the agent.
Usage example	<pre>cmtool getAgentStatus --agentName SOL1-1</pre> <p>Returns the status of the agent named "SOL1-1".</p>	

getLsfInformation	
Description	Retrieves current information about the LSF interface. Note: LSF must be enabled to retrieve information.
Required arguments	None
Optional arguments	None

getLsfInformation		
Available fields	Field output name	Description
	clusterName	The name of the LSF grid cluster.
	lsfAvailable	"1" if LSF is available to the Cluster Manager.
	masterName	The LSF Master Host name.
	numPendingAgentJobs	The number of LSF jobs submitted by the Cluster Manager that are waiting to run.
	numRunningAgentJobs	The number of LSF jobs Cluster Manager submitted that are running now.
	statusMessage	A message.
Usage example	<pre>cmtool --output csv --fields getLsfInformation</pre> Retrieves a Boolean value to indicate whether LSF is available or not.	

getLsfJobs		
Description	Retrieves information about all jobs submitted to LSF.	
Required arguments	None	
Optional arguments	None	
Available fields	Field output name	Description
	agentHostName	The machine name where the agent is running.
	jobNumber	The job number referencing a batch job submitted to LSF.
	jobStatus	The current status of an LSF job.
	resourceRequest	A request to the resource manager for a particular type of agent.
	submitTime	The time the job was submitted to LSF.
Usage example	<pre>cmtool getLsfJobs</pre> Retrieves all LSF job information.	

getResource									
Description	Finds a resource with full detail by the resource ID number.								
Required arguments	<resourceId> - A unique number that identifies each resource. Use getResources to retrieve a list of resourceIds.								
Optional arguments	None								
Available fields	<table border="1"> <thead> <tr> <th>Field output name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>hostMasks</td> <td>This is a semi-colon delimited list of host name masks, used to identify the list of hosts that support a resource. "*" is the wildcard character.</td> </tr> <tr> <td>resourceId</td> <td>A unique number that identifies each resource.</td> </tr> <tr> <td>resourceName</td> <td>This name is used on the eMake parameter: --emake-resource, and can be specified in a build class.</td> </tr> </tbody> </table>	Field output name	Description	hostMasks	This is a semi-colon delimited list of host name masks, used to identify the list of hosts that support a resource. "*" is the wildcard character.	resourceId	A unique number that identifies each resource.	resourceName	This name is used on the eMake parameter: --emake-resource, and can be specified in a build class.
	Field output name	Description							
	hostMasks	This is a semi-colon delimited list of host name masks, used to identify the list of hosts that support a resource. "*" is the wildcard character.							
resourceId	A unique number that identifies each resource.								
resourceName	This name is used on the eMake parameter: --emake-resource, and can be specified in a build class.								
Usage example	<pre>cmtool getResource 7</pre> Retrieves resource 7.								

getResources	
Description	Retrieves a list of all resources.
Required arguments	None
Optional arguments	<pre>--filter <SQL query used to limit the result set for a request. See the possible values below.></pre> <p>Note: There is a syntax difference between MySQL and Oracle/MSSQL for enclosing criteria when using --filter for specific strings—for MySQL, use double quotes; for Oracle/MSSQL, use single quotes.</p> <pre>--maxResults <the maximum number of elements to run from a query></pre> <pre>--firstResult <the starting index for the query result set></pre> <p>Note: --firstResult takes values beginning with "0". A negative value indicates a record starting from the end of the set, counting backwards, so "-1" is the last record, "-2" is the next to last, and so on.</p> <pre>--order <SQL order by clause, used to specify ordering for the query result set.></pre> <pre>--profile <the level of detail to return from a query. Possible values are details (all information) and info (reduced information set).></pre>

getResources			
Available fields	Field output name	Description	SQL query name for --filter and --order
	availableResults	This is a count of 'max' or 'first' results if --maxResults or --firstResult is specified.	N/A
	hostMasks	This is a semi-colon delimited list of host name masks, used to identify the list of hosts that support a resource. "*" is the wildcard character.	host_masks
	resourceId	A unique number that identifies each resource.	id
	resourceName	This name is used on the eMake parameter: --emake-resource, and can be specified in a build class.	resource_name
Usage example	<pre>cmtool getResources --order resource_name</pre> Retrieves a list of resources ordered by the resource name.		

getResourceComments	
Description	Retrieves resource comments.
Required arguments	<resourceId> - A unique number that identifies each resource.
Optional arguments	--commentId
Usage example	<pre>cmtool getResourceComments 29</pre> Retrieves comments for resource 29.

modifyAgentComment	
Description	Modifies an agent comment.
Required arguments	<code><commentId></code> - The unique key that identifies a comment. <code><text></code> - The text of the item.
Optional arguments	<code>--agentId</code> <i><unique, internal number that can change; assigned by the Cluster Manager></i> <code>--agentName</code> <i><name defined by the host where the agent resides [numbers and/or letters]></i> Note: Either <code>agentId</code> or <code>agentName</code> must be specified.
Usage example	<code>cmtool modifyAgentComment 1037 "changed comment" --agentName SOL1-1</code> Changes comment number 1037 on agent SOL1-1 to "changed comment".

modifyResource	
Description	Modifies a resource definition.
Required argument	<code><resourceId></code> - A unique number that identifies each resource.
Optional arguments	<code>--hostMasks</code> <i><a semi-colon delimited list of host name masks, used to identify the list of hosts that support a resource. "*" is the wildcard character.></i> <code>--resourceName</code> <i><the unique name of the resource></i> <code>--description</code> <i><a text description for your reference only></i>
Usage example	<code>cmtool modifyResource 27 --hostMasks "SOL*; SRL*"</code> Sets the host masks for resource 27 to "SOL*; SRL*".

modifyResourceComment	
Description	Modifies a resource comment. Use <code>getResources</code> to retrieve a list of resource IDs.
Required arguments	<code><resourceId></code> - A unique number that identifies each resource. <code><commentId></code> - A unique key identifying a comment. Use <code>getResourceComments</code> to retrieve a list of comment IDs. <code><text></code> - The text of the item.
Optional arguments	None
Usage example	<code>cmtool modifyResourceComment 1 1015 "new xxx"</code> Changes comment 1015 for resource 1.

setAgentDebug							
Description	Sets the agent debug level. [see <code>getAgentStatus</code>] This command sends a message to the agent(s) in real time; therefore, the agents must be up and connected to the Cluster Manager to have any effect.						
Required arguments	<code><level></code> - Possible options are <i>all, commands, environment, fileinfo, log, other, profile, registry, requests, state, test, usage</i> , and <i>nothing</i> .						
Optional arguments	<p><code>--agentId <unique, internal number that can change; assigned by the ClusterManager></code></p> <p><code>--agentName <name defined by the host where the agent resides [numbers and/or letters]></code></p> <p><code>--status <true false> <to choose active or inactive agents (ok or not ok)></code></p> <p><code>--buildId <this further restricts the returned agents to those running a specific build ID></code></p> <p><code>--enabled <true false> <to choose enabled or disabled agents only></code></p> <p><code>--agents <host>[:<port>[:<agentKey>]] <to specify individual agents based on their host name and listening port></code></p>						
Available fields	<table border="1"> <thead> <tr> <th>Field output name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>agentName</td> <td>The name of the configured agent</td> </tr> <tr> <td>result</td> <td>The configuration result</td> </tr> </tbody> </table>	Field output name	Description	agentName	The name of the configured agent	result	The configuration result
	Field output name	Description					
	agentName	The name of the configured agent					
result	The configuration result						
Usage example	<pre>cmtool setAgentDebug profile --agentName SOLAgent-1</pre> <p>Sets SOLAgent-1's debug level to "profile".</p>						

Build Management

createBuildClass	
Description	Creates a build class.
Required arguments	<code><buildClassName></code> - A name assigned by the user for the build class

createBuildClass	
Optional arguments	<p><code>--tagDefinition</code> <a format string that defines the resulting build name></p> <p><code>--annotationLevels</code> <a comma-separated list of values chosen from: basic, history, file, lookup, and waiting></p> <p><code>--maxAgents</code> <maximum number of agents to request for this build></p> <p><code>--minAgents</code> <minimum number of agents required for this build to run></p> <p><code>--platform</code> <operating system being used/supported. If an OS is specified for a build class, builds from other operating systems cannot affiliate themselves with this class. Also, platform must be either Windows, Linux, or Solaris.></p> <p><code>--priority</code> <build priority level. Values can be Low, High, or the default Normal. When assigning resources, an optional priority boost value (number) can be selected to give a build class preference over other builds of the same priority level. Higher boost values correspond to greater preference. Priority value 120 is normal, 220 is high, 20 is low. Each value can be adjusted up or down by 1-10 to "boost" the priority.></p> <p><code>--annoUpload</code> <If set to true, the annotation file is uploaded to Cluster Manager. Values can be Y, N, y, n, yes, no, Yes, or No.></p> <p><code>--resourceRequest</code> <request to the resource manager for a particular type of agent. Value=the name of a pre-existing resource.></p>
Usage example	<pre>cmtool createBuildClass batch --minAgents 5 --maxAgents 12 --priority 35 --resourceRequest blades</pre>

createBuildClassComment	
Description	Creates a new build class comment.
Required arguments	<p><buildClassId> - A unique number assigned by the Cluster Manager for each build class. Use <code>getBuildClass</code> to retrieve a list of build class IDs.</p> <p><text> - The text of the item.</p>
Optional arguments	None
Usage example	<pre>cmtool createBuildClassComment 7 "This build class is for QA builds."</pre> <p>Creates a comment for build class 7.</p>

createBuildComment	
Description	Creates a new build comment.
Required arguments	<buildId> - A unique number assigned by the Cluster Manager for each build. Use <code>getBuilds</code> to retrieve a list of build IDs. <text> - The text of the item.
Optional arguments	None
Usage example	<code>cmtool createBuildComment 1044 "This is our gold build for release 7.0"</code> Creates a comment for build 1044.

deleteBuild	
Description	Deletes a build, including all dependent records.
Required arguments	<buildId> - A unique number assigned by the Cluster Manager for each build. Use <code>getBuilds</code> to retrieve a list of build IDs.
Optional arguments	None
Usage example	<code>cmtool deleteBuild 1037</code> Deletes build 1037.

deleteBuildClass	
Description	Deletes a build class, including all dependent records.
Required arguments	<buildClassId> - A unique number assigned by the Cluster Manager for each build class. Use <code>getBuildClasses</code> to retrieve a list of build class IDs.
Optional arguments	None
Usage example	<code>cmtool deleteBuildClass 7</code> Deletes build class 7.

deleteBuildClasses	
Description	Deletes a set of build classes, including all dependent records.
Required arguments	None
Optional arguments	<p><code>--filter <SQL query used to limit the result set for a request. For a list of possible SQL values, see the <code>getBuildClasses</code> command.></code></p> <p>Note: If no filter is provided, all build classes (except the default) will be deleted.</p>
Usage example	<pre>cmtool deleteBuildClasses --filter "max_agents >20"</pre> <p>Deletes all build classes with more than 20 maximum agents.</p>

deleteBuildClassComment	
Description	Deletes a build class comment.
Required arguments	<p><code><buildClassId></code> - A unique number assigned by the Cluster Manager for each build class.</p> <p><code><commentId></code> - The unique key that identifies a comment.</p> <p>Use <code>getBuildClassComments</code> to retrieve a list of build class comment IDs.</p>
Optional arguments	None
Usage example	<pre>cmtool deleteBuildClassComment 6 1018</pre> <p>Deletes comment 1018 for build class 6.</p>

deleteBuildComment	
Description	Deletes a build comment.
Required arguments	<p><code><buildId></code> - A unique number assigned by the Cluster Manager for each build.</p> <p><code><commentId></code> - The unique key that identifies a comment.</p> <p>Use <code>getBuildComments</code> to retrieve a list of comment IDs.</p>
Optional arguments	None
Usage example	<pre>cmtool deleteBuildComment 1037 1019</pre> <p>Deletes build comment 1019 for build 1037.</p>

deleteBuilds	
Description	<p>Deletes a set of builds, including all dependent records.</p> <p>It is important to remove build logs periodically so they do not fill up the Cluster Manager's available disk space. Uploaded annotation is also considered part of build logs, so remember to clean up build logs regularly if annotation is frequently uploaded to the Cluster Manager.</p> <p>You can also manage build logs using the Cluster Manager web interface. Select the Builds tab, and then create and run a "Builds by Date" filter to display the set of builds that you want to remove. Click Delete Filtered Builds to remove the build logs from disk and from the database.</p>
Required arguments	If no argument is provided, all builds will be deleted.
Optional arguments	<code>--filter <SQL query used to limit the result set for a request. For a list of possible SQL values, see the getBuilds command.></code>
Usage example	<pre>cmtool deleteBuilds --filter "start_time <date_sub(curdate(), interval 20 day) "</pre> <p>Deletes all builds more than 20 days old.</p> <p>Note: This example is valid for MySQL only. If you use Oracle or MSSQL, use syntax that is appropriate for your respective database.</p>

getBuild			
Description	Finds a build with full detail by the build's ID number.		
Required arguments	<code><buildId></code> - A unique number assigned by the Cluster Manager for each build. Use <code>getBuilds</code> to retrieve a list of build IDs.		
Optional arguments	None		
Available fields	Field output names		
	allocatedAgents buildClassId buildClassName buildId buildLogDir buildName commandLine conflicts cwd duration	effectiveAgentAlloc emakeVersion historyExists historyFile hostName ipAddress jobCount lastRequestTime maxAgents minAgents	osUserName platform priority resourceRequest result requestedAgents startTime userLabel userName waitTime
Usage example	<pre>cmtool getBuild 1000</pre> <p>Retrieves build 1000.</p>		

getBuilds	
Description	Retrieves a list of builds.
Required arguments	None
Optional arguments	<p><code>--filter</code> <i><SQL query used to limit the result set for a request. See the possible values below.></i></p> <p>Note: There is a syntax difference between MySQL and Oracle/MSSQL for enclosing criteria when using <code>--filter</code> for specific strings—for MySQL, use double quotes; for Oracle/MSSQL, use single quotes.</p> <p><code>--maxResults</code> <i><maximum number of elements to run from a query></i></p> <p><code>--firstResult</code> <i><starting index for the query result set></i></p> <p>Note: <code>--firstResult</code> takes values beginning with “0”. A negative value indicates a record starting from the end of the set, counting backwards, so “-1” is the last record, “-2” is the next to last, and so on.</p> <p><code>--order</code> <i><SQL order by clause, used to specify ordering for the query result set></i></p> <p><code>--profile</code> <i><level of detail to return from a query. Possible values are details (all information) and info (reduced information set).></i></p> <p>Note: You must set <code>--profile details</code> in order to print fields that are part of the details category.</p>

getBuilds			
Available fields	Field output name	Description	SQL query name for --filter and --order
	allocatedAgents	The number of currently assigned agents for this build.	N/A
	availableResults	This is a count of 'max' or 'first' results if --maxResults or --firstResult is specified.	N/A
	buildClassId	A unique number assigned by the Cluster Manager for each build class.	build_class_id
	buildClassName	A name assigned by the user for the build class.	build_class_name
	buildId	A unique number assigned by the Cluster Manager for each build.	id
	buildLogDir	The directory containing uploaded build logs.	N/A
	buildName	The build name that is the expanded build class tag.	build_name
	commandLine	The original command-line invocation of Electric Make.	command_line
	conflicts	The number of conflicts in the build.	conflicts
	cwd	The current working directory where Electric Make was invoked.	cwd
	duration	The number of milli-seconds the build has been running. Note: duration for running builds is always 0.	duration

getBuilds			
effectiveAgentAlloc	The effective agent allocation percentage. 100% means eMake had all the hosts it needed all the time, while a lesser percentage means eMake had the hosts it needed for that percent of time. Note: effective_agent_alloc for running builds is always 0.	effective_agent_alloc	
emakeVersion	The Electric Make version used for this build.	emake_version	
historyExists	True means the history file existed and was used by the build.	history_exists	
historyFile	The name of the Electric Make history file.	history_file	
hostName	The name of the machine where Electric Make was invoked.	host_name	
ipAddress	The IP address of the machine where Electric Make was invoked.	ip_address	
jobCount	The total number of jobs that ran for the build. Note: job_count for running builds is always 0.	job_count	
lastRequestTime	The last time Electric Make requested agents for this build.	N/A	
maxAgents	The maximum number of agents to request for this build.	max_agents	
minAgents	The minimum number of agents required for this build to run.	min_agents	

getBuilds			
	osUserName	The OS-level name for the user who started Electric Make.	os_user_name
	platform	The operating system being used/supported. If an OS is specified for a build class, builds from other operating systems cannot affiliate themselves with this class.	platform
	priority	The build priority level. When assigning resources, an optional priority boost value can be selected to give a build class preference over other builds of the same priority level. Higher boost values correspond to greater preference.	priority
	resourceRequest	A request to the resource manager for a particular type of agent.	resource_request
	result	The build result code. -1 means the build is still running, 0-254 are actual exit codes, 256 means the build timed out, and 257 means the build was stopped.	result
	requestedAgents	The number of agents Electric Make requested.	N/A
	startTime	The time the build was started.	start_time
	userLabel	The user-supplied label (via the eMake command-line), attached to the build.	user_label

getBuilds			
	userName	The unique name of the user.	user_name
	waitTime	The number of seconds Electric Make was stalled because it had to wait for agents. Note: wait_time for running builds is always 0.	wait_time
Usage example	<pre>cmtool --output simple --fields "startTime,buildName,userId,duration" getBuilds --filter "duration >10000"</pre> <p>Returns the start time, build name, userid, and duration of all builds that ran more than 10 seconds.</p>		

getBuildComments		
Description	Retrieves a list of related build comments.	
Required arguments	<buildId> - A unique number assigned by the Cluster Manager for each build.	
Optional arguments	--commentId <unique key that identifies a comment>	
Available fields	Field output name	Description
	commentId	The unique key that identifies a comment.
	createTime	The time when the item was created.
	lastModifiedBy	The last time the agent experienced an error.
	modifyTime	The time when the item was last modified.
	text	The text of the item.
Usage example	<pre>cmtool getBuildComments 1000 --commentId 1039</pre> <p>Retrieves comment 1039 for build 1000.</p>	

getBuildClass							
Description	Finds a build class with full detail by its ID.						
Required arguments	<buildClassId> - A unique number assigned by the Cluster Manager for each build class. Use <code>getBuildClasses</code> to retrieve a list of build class IDs.						
Optional arguments	None						
Available fields (see <code>get-BuildClasses</code> for field descriptions)	<table border="1"> <thead> <tr> <th colspan="3">Field output names</th> </tr> </thead> <tbody> <tr> <td>annotationLevels annoUpload buildClassId buildClassName</td> <td>defaultClass maxAgents minAgents notifyOnBuildEnd</td> <td>platform priority resourceRequest tagDefinition</td> </tr> </tbody> </table>	Field output names			annotationLevels annoUpload buildClassId buildClassName	defaultClass maxAgents minAgents notifyOnBuildEnd	platform priority resourceRequest tagDefinition
Field output names							
annotationLevels annoUpload buildClassId buildClassName	defaultClass maxAgents minAgents notifyOnBuildEnd	platform priority resourceRequest tagDefinition					
Usage example	<pre>cmtool getBuildClass 1</pre> <p>Retrieves build class 1.</p>						

getBuildClasses	
Description	Retrieves a list of build classes with limited detail.
Required arguments	None
Optional arguments	<p><code>--filter</code> <SQL query used to limit the result set for a request. See the possible values below.></p> <p>Note: There is a syntax difference between MySQL and Oracle/MSSQL for enclosing criteria when using <code>--filter</code> for specific strings—for MySQL, use double quotes; for Oracle/MSSQL, use single quotes.</p> <p><code>--maxResults</code> <maximum number of elements to run from a query></p> <p><code>--firstResult</code> <starting index for the query result set></p> <p>Note: <code>--firstResult</code> takes values beginning with “0”. A negative value indicates a record starting from the end of the set, counting backwards, so “-1” is the last record, “-2” is the next to last, and so on.</p> <p><code>--order</code> <SQL order by clause, used to specify ordering for the query result set.></p> <p><code>--profile</code> <level of detail to return from a query. Possible values are <i>details</i> (all information) and <i>info</i> (reduced information set).></p>

getBuildClasses			
Available fields	Field output name	Description	SQL query name for --filter and --order
	annotationLevels	Annotation choices to include in the annotation file. Possible values are basic, history, file, lookup, and waiting.	annotation_levels
	annoUpload	If set to true, the annotation file is uploaded to Cluster Manager.	anno_upload
	availableResults	This is a count of 'max' or 'first' results if --maxResults or --firstResult is specified.	N/A
	buildClassId	A unique number assigned by the Cluster Manager for each build class.	id
	buildClassName	A name assigned by the user for the build class.	build_class_name
	defaultClass	If set, this is the default build class and cannot be deleted.	default_class
	maxAgents	The maximum number of agents to request for this build.	max_agents
	minAgents	The minimum number of agents required for this build to run.	min_agents
	notifyOnBuildEnd	If set to true, the currently logged-in user will receive an email when the build is finished.	notify_on_build_end
	platform	The operating system being used/supported. If an OS is specified for a build class, builds from other operating systems cannot affiliate themselves with this class.	platform

getBuildClasses			
	priority	The build priority level. When assigning resources, an optional priority boost value can be selected to give a build class preference over other builds of the same priority level. Higher boost values correspond to greater preference.	priority
	resourceRequest	A request to the resource manager for a particular type of agent.	resource_request
	tagDefinition	A format string that defines the resulting build name.	tag_definition
Usage example	<pre>cmtool getBuildClasses --filter "min_agents <5"</pre> Retrieves a list of build classes that require less than 5 agents.		

getBuildClassComments		
Description	Retrieves a list of related build class comments.	
Required arguments	<i><buildClassId></i> - A unique number assigned by the Cluster Manager for each build class. Use <code>getBuildClasses</code> to retrieve a list of build class IDs.	
Optional arguments	<code>--commentId</code> <i><unique key that identifies a comment></i>	
Available fields	Field output name	Description
	commentId	The unique key that identifies a comment.
	createTime	The time when the item was created.
	lastModifiedBy	The last time the agent experienced an error.
	modifyTime	The time when the item was last modified.
	text	The text of the item.
Usage example	<pre>cmtool getBuildClassComments 12</pre> Retrieves all build class comments for build class 12.	

getBuildUserStats			
Description	Retrieves a list of user build statistics, grouped by user name, IP address, or host name.		
Required arguments	<groupBy> - Possible values are <code>hostName</code> , <code>ipAddress</code> , and <code>userName</code> .		
Optional arguments	<p><code>--filter</code> <SQL query used to limit the result set for a request. See the possible values below.></p> <p>Note: There is a syntax difference between MySQL and Oracle/MSSQL for enclosing criteria when using <code>--filter</code> for specific strings—for MySQL, use double quotes; for Oracle/MSSQL, use single quotes.</p> <p><code>--order</code> <SQL order by clause, used to specify ordering for the query result set.></p>		
Available fields	Field output name	Description	SQL query name for <code>--filter</code> and <code>--order</code>
	<code>duration</code>	The total number of milliseconds of all builds, filtered by the value specified in the <code>groupBy</code> argument.	<code>duration</code>
	<code>entryName</code>	The value specified in the <code>groupBy</code> argument. If <code>groupBy</code> is "userName", the entry name is the user name.	
	<code>numOfBuilds</code>	The number of builds.	
	<code>waitTime</code>	The number of seconds Electric Make was stalled because it had to wait for agents.	<code>wait_time</code>
	<code>workload</code>	The total number of seconds used by the agents for all of the filtered builds.	<code>workload</code>
Usage example	<pre>cmtool getBuildUserStats hostName --filter "duration >30000" --order "waitTime desc, entryName asc"</pre> <p>Retrieves build user statistics for builds longer than 30 seconds, grouped by host name and ordered by wait time in a descending order and by entry name (in this case host name) in an ascending order.</p>		

modifyBuild	
Description	Modifies a build.
Required arguments	<buildId> - A unique number assigned by the Cluster Manager for each build. <priority> - The build priority level. Value can be Low or Normal, but NOT High.
Optional arguments	None
Usage example	<code>cmtool modifyBuild 1137 20</code> Changes build 1137 to priority 20.

modifyBuildClass	
Description	Modifies a build class.
Required arguments	<buildClassId> - A unique number assigned by the Cluster Manager for each build class.
Optional arguments	<p><code>--buildClassName</code> <name assigned by the user for the build class></p> <p><code>--tagDefinition</code> <format string that defines the resulting build name></p> <p><code>--annotationLevels</code> <a comma-separated list of possible values are: basic, history, file, lookup, and waiting></p> <p><code>--maxAgents</code> <maximum number of agents to request for this build></p> <p><code>--minAgents</code> <minimum number of agents required for this build to run></p> <p><code>--platform</code> <the operating system being used/supported. If an OS is specified for a build class, builds from other operating systems cannot affiliate themselves with this class. The value can be Windows, Linux, or Solaris.></p> <p><code>--priority</code> <the build priority level. When assigning resources, an optional priority boost value can be selected to give a build class preference over other builds of the same priority level. Higher boost values correspond to greater preference. Value can be Low, High, or the default Normal.></p> <p><code>--annoUpload</code> <if set to true, the annotation file is uploaded to Cluster Manager. Values can be Y, N, y, n, Yes, No, yes, no.></p> <p><code>--resourceRequest</code> <request to the resource manager for a particular type of agent. Value is the name of the pre-existing resource.></p>
Usage example	<code>cmtool modifyBuildClass 1 --annoupload true</code> Changes build class 1 to upload annotation files.

modifyBuildComment	
Description	Modifies a build comment.
Required arguments	<p><buildId> - A unique number assigned by the Cluster Manager for each build.</p> <p><commentId> - The unique key that identifies a comment.</p> <p><text> - The text of the item.</p>
Optional arguments	None
Usage example	<code>cmtool modifyBuildComment 16975 1137 "This is not a usable build"</code>

modifyBuildClassComment	
Description	Modifies a build class comment.
Required arguments	<p><buildClassId> - A unique number assigned by the Cluster Manager for each build class.</p> <p><commentId> - The unique key that identifies a comment.</p> <p><text> - The text of the item.</p>
Optional arguments	None
Usage example	<code>cmtool modifyBuildClassComment 1037 1129 "This is a low-priority class"</code>

setDatabaseConfiguration	
Description	Modifies database configuration settings
Required arguments	<p><databaseName> - The database instance name.</p> <p><databaseType> - The database type. Possible values are mysql, oracle, and sqlserver.</p> <p><hostName> - Machine name where the database is installed.</p> <p><port> - Database port number.</p> <p><userName> - Unique name of the user that is used to access the database.</p> <p><password> - Secret value used to identify an account for a particular user.</p>
Optional arguments	None

stopBuild	
Description	Stops a running build. (This command has no effect on completed builds.)
Required arguments	<buildId> - A unique number assigned by the Cluster Manager for each build. Use <code>getBuilds --filter "result <0"</code> to retrieve a list of running builds.
Optional arguments	None
Usage example	<code>cmtool stopBuild 16937</code>

Cluster Management

createServerComment	
Description	Creates a new server comment. Server comments are displayed on the Home page of the Cluster Manager machine.
Required arguments	<text> - The text of the item.
Optional arguments	None
Usage example	<code>cmtool createServerComment "cluster needs more servers to handle production builds"</code>

deleteLicense	
Description	Deletes a license.
Required arguments	<productName> - ElectricAccelerator, the name of the license. <featureName> - Feature name of the license, which is currently the "server".
Optional arguments	None
Usage example	<code>cmtool deleteLicense ElectricAccelerator Server</code> Deletes the license stored in the server.

deleteMessage	
Description	Deletes a specific message, including all dependent records. Messages are listed in the Cluster Manager interface Messages tab and generally are notifications about issues with agents or the Cluster Manager.
Required arguments	<code><messageId></code> - The numeric value that uniquely identifies each message.
Optional arguments	None
Usage example	<code>cmtool deleteMessage 501</code>

deleteMessages	
Description	Deletes a set of messages, including all dependent records.
Required arguments	None
Optional arguments	<code>--filter <SQL query used to limit the result set for a request. For a list of possible SQL values, see the <code>getMessages</code> command.></code>
Usage example	<pre>cmtool deleteMessages --filter "create_time <date_sub(curdate(), interval 200 day) "</pre> <p>Removes all messages more than 200 days old.</p> <p>Note: This example is valid for MySQL only. If you use Oracle or MSSQL, use syntax that is appropriate for your respective database.</p>

deleteServerComment	
Description	Deletes a server comment.
Required arguments	<code><commentId></code> - The unique key that identifies a comment.
Optional arguments	None
Usage example	<code>cmtool deleteServerComment 1396</code>

exportData	
Description	Exports Cluster Manager data to a file.
Required arguments	<p><fileName> - The filename or path to export to. If you use a filename, the destination is the current working directory of the Java process, for example, /opt/ecloud/i686_Linux or C:\ECloud\i686_win32. If you use a path, the Cluster Manager Java user must have execute and write access to the destination path.</p> <p>Note: This is a full database dump and it may take an extended period of time to complete depending on the size of the database.</p>
Optional arguments	None
Usage example	<code>cmtool exportData fileabc</code>

getLicense	
Description	Retrieves information for one license.
Required arguments	<p><productName> - ElectricAccelerator, the name of the license.</p> <p><featureName> - Feature name of the license.</p>
Optional arguments	None
Usage example	<code>cmtool getLicense ElectricAccelerator Server</code>

getLicenses	
Description	Retrieves all license data.
Required arguments	None
Optional arguments	None
Usage example	<code>cmtool getLicenses</code>

getMessage											
Description	Retrieves a particular message.										
Required arguments	<messageId> - The numeric value that uniquely identifies each message.										
Optional arguments	None										
Available fields (see get-Messages for field name descriptions)	<table border="1"> <thead> <tr> <th colspan="2">Field output name</th> </tr> </thead> <tbody> <tr> <td>agentId</td> <td>createTime</td> </tr> <tr> <td>agentName</td> <td>messageId</td> </tr> <tr> <td>buildId</td> <td>severity</td> </tr> <tr> <td>buildName</td> <td>text</td> </tr> </tbody> </table>	Field output name		agentId	createTime	agentName	messageId	buildId	severity	buildName	text
Field output name											
agentId	createTime										
agentName	messageId										
buildId	severity										
buildName	text										
Usage example	<pre>cmtool --output csv --fields buildId,severity,text getMessages -- filter "text like '%I/O%'"</pre> <p>Lists all messages in the Cluster Manager that contain the string 'I/O'.</p>										

getMessages	
Description	Retrieves a list of messages
Required arguments	None
Optional arguments	<p><code>--filter</code> <SQL query used to limit the result set for a request. See the possible values below.></p> <p>Note: There is a syntax difference between MySQL and Oracle/MSSQL for enclosing criteria when using <code>--filter</code> for specific strings—for MySQL, use double quotes; for Oracle/MSSQL, use single quotes.</p> <p><code>--maxResults</code> <maximum number of elements to run from a query></p> <p><code>--firstResult</code> <starting index for the query result set></p> <p>Note: <code>--firstResult</code> takes values beginning with "0". A negative value indicates a record starting from the end of the set, counting backwards, so "-1" is the last record, "-2" is the next to last, and so on.</p> <p><code>--order</code> <SQL order by clause, used to specify ordering for the query result set.></p> <p><code>--profile</code> <level of detail to return from a query. Possible values are details (all information) and info (reduced information set).></p>

getMessages			
Available fields	Field output name	Description	SQL query name for --filter and --order
	agentId	A unique, internal number assigned to each agent by the Cluster Manager; this number can change.	N/A
	agentName	A name defined by the host where the agent resides [numbers and/or letters].	agent_name
	buildId	A unique number assigned by the Cluster Manager for each build.	build_id
	buildName	The build name that is the expanded build class tag.	N/A
	createTime	The time when the item was created.	create_time
	messageId	The numeric value that uniquely identifies each message.	id
	severity	The severity level of the event: Info, Warning, or Error. For --filter and --order, use the following numerical values: 1 = Info 2 = Warning 3 = Error	severity
	text	The text of the item.	text
Usage example	cmtool getMessage 47		

getResourceStats	
Description	Retrieves resource usage statistics.
Required arguments	None

getResourceStats	
Optional arguments	<p><code>--filter</code> <SQL query used to limit the result set for a request. See the possible values below.></p> <p>Note: There is a syntax difference between MySQL and Oracle/MSSQL for enclosing criteria when using <code>--filter</code> for specific strings—for MySQL, use double quotes; for Oracle/MSSQL, use single quotes.</p> <p><code>--maxResults</code> <maximum number of elements to run from a query></p> <p><code>--firstResult</code> <starting index for the query result set></p> <p>Note: <code>--firstResult</code> takes values beginning with “0”. A negative value indicates a record starting from the end of the set, counting backwards, so “-1” is the last record, “-2” is the next to last, and so on.</p> <p><code>--order</code> <SQL order by clause, used to specify ordering for the query result set.></p> <p><code>--profile</code> <level of detail to return from a query. Possible values are <i>details</i> (all information) and <i>info</i> (reduced information set).></p>

getResourceStats			
Available fields	Field output name	Description	SQL query name for --filter and --order
	agentClusterShortage	The difference between the maximum number of agents requested by all builds and the number of agents that were assigned.	agent_cluster_shortage
	agentDemand	The total maximum number of requests for agents by all running builds.	agent_demand
	agentLicenseShortage	The difference between the maximum request for agents by all builds and the number of agents the license allows.	agent_license_shortage
	agentsAvailable	The total number of active agents in the cluster.	agents_available
	agentsInUse	The total number of agents assigned to builds.	agents_in_use
	availableResults	This is a count of 'max' or 'first' results if --maxResults or --firstResult is specified.	N/A
	buildsDuration	The average amount of time the current builds have been running.	builds_duration
	buildsRunning	Average number of simultaneous builds running during a specific time period.	builds_running
	createTime	The time when the item was created.	create_time
	duration	The number of milliseconds the build has been running.	duration

getResourceStats			
	resourceName	This name is used on the eMake parameter: <code>--emake-resource</code> , and can be specified in a build class. It is used in the <code>ea_resource</code> table and also matches the resource requirement string for eMake.	resource_name
	resourceStatId	The resource ID number that uniquely identifies every resource.	id
Usage example	<pre>cmtool getResourceStats --maxResults 100 --order "id desc" --filter "resource_name='Cluster' "</pre> <p>Retrieves the 100 most current resource statistic records for the entire cluster.</p>		

getServer	
Description	Retrieves server configuration.
Required arguments	None
Optional arguments	None

getServer		
Available fields	Field output name	Description
	agentAllocationPolicy	Defined as either <i>exclusive</i> or <i>shared</i> .
	agentLockTimerSec	When jobs run beyond this number of seconds, the agent should be locked.
	badAgents	The number of enabled agents with a bad status.
	disabledAgents	The number of disabled agents.
	emailInterval	The number of minutes between email notifications.
	emailItemLimit	Maximum number of messages per email notification.
	goodAgents	The number of enabled agents with a good status.
	logDaysToKeep	The number of days to keep message log entries.
	lsfAvailable	True if LSF is available to the Cluster Manager.

getServer		
	mailFrom	The value to use in the From header element.
	mailPrefix	The string used to prefix subject lines.
	maxAgents	The maximum number of agents to request for this build.
	maxClockSkew	Specify the maximum clock skew (in seconds) allowed between the Electric Make client and agents in the cluster.
	minAgents	The minimum number of agents required for this build to run.
	preemptionPolicy	The allocation preemption policy.
	priority	The build priority level. When assigning resources, an optional priority boost value can be selected to give a build class preference over other builds of the same priority level. Higher boost values correspond to greater preference.
	resourceManagerType	The type of resource manager that Cluster Manager should employ.
	resourceStatInterval	In minutes, the interval to collect stats on resource usage.
	resourceStatKeep	The number of minutes of resource usage statistics to keep.
	runningBuilds	The number of incomplete builds in the system.
Usage example	cmtool getServer	

getServerComments	
Description	Retrieves a list of related server comments.
Required arguments	None
Optional arguments	--commentId <unique key that identifies a comment>

getServerComments		
Available fields	Field output name	Description
	commentId	The unique key that identifies a comment.
	createTime	The time when the item was created.
	lastModifiedBy	The last time the agent experienced an error.
	modifyTime	The time when the item was last modified.
	text	The text of the item.
Usage example	<pre>cmtool getServerComments</pre> Returns all comments related to the server.	

getVersion		
Description	Retrieves server version information.	
Required arguments	None	
Optional arguments	None	
Available fields	Field output name	Description
	label	The Electric Cloud build label for the server.
	protocolVersion	The server protocol version.
	schemaVersion	The server database schema version.
	version	The string identifying a component version.
Usage example	<pre>cmtool getVersion</pre>	

importData	
Description	Imports Cluster Manager data from a file.
Required arguments	<p><fileName> - The name of the file to import. The file's path is relative to the current working directory of the Java process, for example, /opt/ecloud/i686_Linux or C:\ECloud\i686_win32.</p> <p>Note: Because this imports a full database dump, be advised of the following:</p> <ul style="list-style-type: none"> - The import may take an extended period of time to complete depending on the size of the database. - You must manually delete any old/unused agents from the agents list. - You must update the license file after import if it previously expired.
Optional arguments	None
Usage example	<code>cmtool importData fileabc</code>

importLicenseData	
Description	Imports one or more licenses.
Required arguments	<licenseFile> - Name of the file containing the license with the path.
Optional arguments	None
Usage example	<code>cmtool importLicenseData ./license.xml</code>

logMessage	
Description	Creates a custom message on the Cluster Manager Messages page.
Required arguments	"message text"
Optional arguments	<p>--severity <severity> - Possible values are: Debug, Info, Warning, Error (or 0, 1, 2, 3).</p> <p>--buildId <buildId> - The message applies to the specified build only.</p> <p>--agentName <agentName> - The message applies to the specified agent name only.</p> <p>Note: If --buildId and --agentName are on the same line, the message is applied to build and the agent name.</p>
Usage example	<code>cmtool logMessage "some text"</code>

modifyServer	
Description	Modifies the server configuration.
Required arguments	None
Optional arguments	<p><code>--priority</code> <<i>The default build priority level. When assigning resources, an optional priority boost value can be selected to give a build class preference over other builds of the same priority level. Higher boost values correspond to greater preference.</i>></p> <p><code>--emailInterval</code> <<i>number of minutes between email notifications</i>></p> <p><code>--emailItemLimit</code> <<i>maximum number of messages per email notification</i>></p> <p><code>--agentAllocationPolicy</code> <<i>exclusive or shared</i>></p> <p><code>--preemptionPolicy</code> <<i>the allocation preemption policy</i>></p> <p><code>--maxClockSkew</code> <<i>specify the maximum clock skew (in seconds) allowed between the Electric Make client and agents in the cluster</i>></p> <p><code>--maxAgents</code> <<i>maximum number of agents to request for this build</i>></p> <p><code>--minAgents</code> <<i>minimum number of agents required for this build to run</i>></p> <p><code>--resourceManagerType</code> <<i>values= none, ea, lsf, cloud, prioritypool to define what Cluster Manager should employ</i>></p> <p><code>--mailFrom</code> <<i>value to use in the From header element</i>></p> <p><code>--mailPrefix</code> <<i>string used to prefix subject lines</i>></p> <p><code>--logDaysToKeep</code> <<i>the number of days to keep message log entries</i>></p> <p><code>--resourceStatInterval</code></p> <p><code>--resourceStatKeep</code></p> <p><code>--wideDeepAllocationPolicy</code><<i>deep or wide - Deep means the agent allocation algorithm favors assigning more agents on the same host to a build. Wide means the algorithm favors assigning more agents from different hosts. If wide, be sure --agentAllocationPolicy is set to shared.</i>></p>
Usage example	<pre>cmtool modifyServer --mailFrom "cm@ourhost.com" --mailPrefix "cm message:"</pre> <p>Changes the mail "from" and mail prefix values used for mail notifications sent by the server.</p>

modifyServerComment	
Description	Modifies a server comment.
Required arguments	<p><<i>commentId</i>> - The unique key that identifies a comment.</p> <p><<i>text</i>> - The text of the item.</p>

modifyServerComment	
Optional arguments	None
Usage example	<code>cmtool modifyServerComment 1178 "Server is fine"</code>

shutdownServer	
Description	Stops the server. Note: Use with caution, it works!
Required arguments	None
Optional arguments	<code>--restart <true false></code>
Usage example	<code>cmtool shutdownServer</code>

testAgents	
Description	Instructs the Cluster Manager to contact each active agent and update its status.
Required arguments	None
Optional arguments	<code>--agentId<unique, internal number that can change; assigned by the Cluster Manager></code> <code>--agentName <name defined by the host where the agent resides [numbers and/or letters]></code> <code>--filter <SQL query used to limit the result set for a request. For a list of possible SQL values, see the getAgents command.></code>
Usage example	<code>cmtool testAgents --filter "agent_name like '%bl%'"</code> This command contacts all agents whose name contains 'bl' and updates their status.

Reporting

createFilter	
Description	Creates a named filter for a specific table. Note: Non-global filters are stored by user ID; therefore, the same name can be used by more than one user.
Required arguments	<i><tableName></i> - A short string that uniquely identifies the table being filtered. Note: Possible table names are: ec_agent, ec_build, ec_build_class, ec_filter, ec_message, ec_resource, ec_resource_stat. <i><filterName></i> - A short string that uniquely identifies the filter. <i><filterQuery></i> - A SQL order by clause for the associated table.
Optional arguments	<code>--order <SQL order by clause, used to specify ordering for the query result set></code> <code>--global <if true, this is a globally visible filter></code>
Usage example	<pre>cmtool createFilter ec_agents linuxAgents ""platform = 'linux' " --global true</pre> Creates a global filter that selects only Linux agents.

deleteFilter	
Description	Deletes a named filter for a specific table.
Required arguments	<i><tableName></i> - A short string that uniquely identifies the table being filtered. Note: Possible table names are: ec_agent, ec_build, ec_build_class, ec_filter, ec_message, ec_resource, ec_resource_stat. <i><filterName></i> - A short string that uniquely identifies the filter.
Optional arguments	<code>--global <if true, this is a globally visible filter></code>
Usage example	<pre>cmtool deleteFilter ec_agents linuxAgents --global true</pre>

getCurrentServerLoad	
Description	Retrieves information about the current resource load.
Required arguments	None

getCurrentServerLoad		
Optional arguments	None	
Available fields	Field output name	Description
	agentsAvailable	The total number of active agents in the cluster.
	agentClusterShortage	The difference between the maximum number of agents requested by all builds and the number of agents that were assigned.
	agentDemand	The total maximum number of requests for agents by all running builds.
	agentLicenseShortage	The difference between the maximum request for agents by all builds and the number of agents the license allows.
	agentsInUse	The total number of agents assigned to builds.
	buildsDuration	The average amount of time the current builds have been running.
	buildsRunning	Average number of simultaneous builds running during a specific time period.
	createTime	The time when the item was created.
	duration	The number of milli-seconds the build has been running.
	resourceName	This name is used on the eMake parameter: <code>--emake-resource</code> , and can be specified in a build class. It is used in the <code>ea_resource</code> table and also matches the resource requirement string for eMake.
	resourceStatId	The resource ID number that uniquely identifies every resource.
Usage example	<code>cmtool getCurrentServerLoad</code>	

getFilter	
Description	Retrieves a named filter for a specific table.
Required arguments	<p><code><tableName></code> - A short string that uniquely identifies the table being filtered.</p> <p>Note: Possible table names are: <code>ec_agent</code>, <code>ec_build</code>, <code>ec_build_class</code>, <code>ec_filter</code>, <code>ec_message</code>, <code>ec_resource</code>, <code>ec_resource_stat</code>.</p> <p><code><filterName></code> - A short string that uniquely identifies the filter.</p>

getFilter	
Optional arguments	<code>--global</code> <i><if true, this is a globally visible filter></i>
Usage example	<code>cmtool getFilter ec_agent agentFilter</code>

getFilters	
Description	Retrieves a list of saved filters for the current user.
Required arguments	None
Optional arguments	<p><code>--filter</code> <i><SQL query used to limit the result set for a request. For a list of possible SQL values, see the getAgents command.></i></p> <p><code>--maxResults</code> <i><maximum number of elements to run from a query></i></p> <p><code>--firstResult</code> <i><starting index for the query result set></i></p> <p>Note: <code>--firstResult</code> takes values beginning with "0". A negative value indicates a record starting from the end of the set, counting backwards, so "-1" is the last record, "-2" is the next to last, and so on.</p> <p><code>---order</code> <i><SQL order by clause, used to specify ordering for the query result set></i></p>
Usage example	<pre>cmtool getFilters --filter "table_name = 'ec_agent' && user_name is null"</pre> <p>Retrieves a list of all global filters for the agent table.</p>

modifyFilter	
Description	Updates a named filter for a specific table.
Required arguments	<p><i><tableName></i> - A short string that uniquely identifies the table being filtered.</p> <p>Note: Possible table names are: ec_agent, ec_build, ec_build_class, ec_filter, ec_message, ec_resource, ec_resource_stat.</p> <p><i><filterName></i> - A short string that uniquely identifies the filter.</p> <p><i><filterQuery></i> - A SQL order by clause for the associated table.</p>
Optional arguments	<p><code>--order</code> <i><SQL order by clause, used to specify ordering for the query result set></i></p> <p><code>--global</code> <i><if true, this is a globally visible filter. This parameter is required for global filters.></i></p>
Usage example	<code>cmtool modifyFilter ec_agent agentFilter "id 750" --order agent_name</code>

User Management

addGroupMember	
Description	Adds a user name to the member list for a specific group.
Required arguments	<i><groupName></i> - The unique name of the group. <i><userName></i> - The unique name of the user.
Optional arguments	None
Usage example	<code>cmtool addGroupMember DevGroupA ec123</code> Adds user 'ec123' to group DevGroupA.

changeOwnUser	
Description	Modifies the settings for the currently logged-in user.
Required arguments	<i><userName></i> - The unique name of the user.
Optional arguments	<code>--fullUserName <real world name of the user></code> <code>--email <the associated user email address></code> <code>--password <secret value used to identify an account for a particular user></code> <code>--passwordFile <path to password file, if --password is also specified, --passwordFile overrides its value in the command line></code>
Usage example	<code>cmtool ec123 --fullUserName "Mary Smith"</code>

createGroup	
Description	Creates a new local group.
Required arguments	<i><groupName></i> - The unique name of the group.
Optional arguments	None
Usage example	<code>cmtool createGroup DevGroupA</code>

createUser	
Description	Creates a new local user.
Required arguments	<code><userName></code> - The unique name of the user. <code><password></code> - The secret value used to identify an account for a particular user.
Optional arguments	<code>--fullUserName <real world name of the user></code> <code>--email <the associated user email address></code> <code>--passwordFile <path to password file, if --password is also specified, --passwordFile overrides its value in the command line></code>
Usage example	<pre>cmtool createUser ec123 psword --fullUserName "Bob Smith" --email "ec123@ourhost.com"</pre> <p>Creates a new user named “ec123” whose real-world name is Bob Smith; with “psword” as his password.</p> <p>Note: If you do not wish to expose passwords on the command line, you can omit the password from the example above. Press the Enter key after typing the command string (without the password) and you will be prompted for the password.</p>

deleteGroup	
Description	Deletes a local group.
Required arguments	<code><groupName></code> - The unique name of the group.
Optional arguments	None
Usage example	<pre>cmtool deleteGroup DevGroupA</pre> <p>Removes the 'DevGroupA' group from the Cluster Manager.</p>

deleteUser	
Description	Deletes a local user.
Required arguments	<code><userName></code> - The unique name of the user.
Optional arguments	None
Usage example	<pre>cmtool deleteUser ec123</pre>

getAccessEntries		
Description	Retrieves permissions for all users and groups that were granted server access.	
Required arguments	None	
Optional arguments	None	
Available fields	Field output name	Description
	entityName	A user or group name in an access entry.
	permissions	The list of permission flags for a particular entity.
Usage example	cmtool getAccessEntries	

getGroupMembers		
Description	Retrieves a list of users in a specific group.	
Required arguments	<groupName> - The unique name of the group.	
Optional arguments	None	
Available fields	Field output name	Description
	userName	The unique name of the user.
Usage example	cmtool getGroupMembers Retrieves a list of user name elements.	

getGroups	
Description	Finds all groups known to the server. If "local" is true, returns local groups only.
Required arguments	<userName> - The unique name of the user.
Optional arguments	--local <if true, returns local users only>

getGroups		
Available fields	Field output name	Description
	groupName	The unique name of the group.
	mutable	True if the associated user or group record is modifiable.
	providerName	The human-readable name configured for the directory provider of a specific user or group.
Usage example	<pre>cmtool getGroups</pre> Returns a list of <code>groupInfo</code> elements.	

getEffectivePermissions			
Description	Retrieves the permissions for the currently logged-in user.		
Required arguments	None		
Optional arguments	None		
Available field	Field output name	Description	
	permissions	The list of permission flags for a particular entity	
Available permission flags	Permission names		
	AgentsDelete AgentsRead AgentsWrite BuildsDelete BuildsRead BuildsWrite ClassesDelete ClassesRead ClassesWrite	EMakeImpersonate EMakeInvoke MaintenanceDelete MaintenanceRead MaintenanceWrite MessageLogDelete MessageLogRead MessageLogWrite	ReportsDelete ReportsRead ReportsWrite ResourcesDelete ResourcesRead ResourcesWrite ServerAccess UserModify
Usage example	<pre>cmtool getEffectivePermissions</pre> Retrieves the permissions for the currently logged-in user.		

getPermissions		
Description	Retrieves permissions for a particular user or group.	
Required arguments	< <i>principalType</i> > - Value = user or group. < <i>entityName</i> > - A user or group name in an access entry.	
Optional arguments	None	
Available field	Field output name	Description
	permissions	The list of permission flags for a particular entity
Available permission flags	Permission names	
	AgentsDelete AgentsRead AgentsWrite BuildsDelete BuildsRead BuildsWrite ClassesDelete ClassesRead ClassesWrite	EMakeImpersonate EMakeInvoke MaintenanceDelete MaintenanceRead MaintenanceWrite MessageLogDelete MessageLogRead MessageLogWrite
Usage example	cmtool getPermissions group DevGroupA Retrieves permissions for group DevGroupA.	

getUser	
Description	Finds a specific user known to the server.
Required arguments	< <i>userName</i> > - The unique name of the user.
Optional arguments	None

getUser		
Available fields	Field output name	Description
	email	The associated user email address.
	fullUserName	The real world name of the user.
	groupName	The unique name of the group.
	mutable	True if the associated user or group record is modifiable.
	providerName	The human-readable name configured for the directory provider of a specific user or group.
	userName	The unique name of the user.
Usage example	<pre>cmtool getUser ec123</pre> Retrieves the attributes for user ec123.	

getUsers		
Description	Finds all users known to the server. If "local" is true, returns local users only.	
Required arguments	None	
Optional arguments	<pre>--pattern <a wildcard pattern for a user name where "*" matches any character or SQL "like" string. If LDAP is set up for getting users, the * is the preferred wildcard, as % is not understood by LDAP (this limits the result set to records in the local database).></pre> <pre>--local <if true, returns local users only></pre>	
Available fields	Field output names	
(see <code>getUser</code> for field descriptions)	email fullUserName	mutable providerName userName
Usage example	<pre>cmtool getUsers --pattern ec*</pre> Retrieves information on all user IDs that begin with 'ec'.	

getUserSettings	
Description	Retrieves settings for the currently logged-in user.
Required arguments	None
Optional arguments	None
Usage example	<code>cmtool getUserSettings</code>

login					
Description	Logs in to the client with the appropriate credentials and creates a session file in the users home directory, which allows subsequent calls to cmtool to connect to the cluster manager.				
Required arguments	<p><i><userName></i> - The unique name of the user.</p> <p><i><password></i> - The secret value used to identify an account for a particular user.</p>				
Optional arguments	<code>--passwordFile <path to password file, if --password is also specified, --passwordFile overrides its value in the command line></code>				
Available fields	<table border="1"> <thead> <tr> <th>Field output name</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>sessionId</td> <td>This is a session "cookie."</td> </tr> </tbody> </table>	Field output name	Description	sessionId	This is a session "cookie."
	Field output name	Description			
sessionId	This is a session "cookie."				
Usage example	<p><code>cmtool login ec123 bobs</code></p> <p>Logs in a user named "ec123" whose password is "bobs".</p> <p>Note: If you do not wish to expose passwords on the command line, you can omit the password from the example above. Press the Enter key after typing the command string (without the password) and you will be prompted for the password.</p>				

logout	
Description	Logs out of the client session.
Required arguments	None
Optional arguments	None

modifyGroup	
Description	Modifies a local group.
Required arguments	<i><groupName></i> - The unique name of the group.
Optional arguments	<code>--newName <new group name></code>
Usage example	<code>cmtool modifyGroup DevGroupA --newName GroupDevA</code>

modifyUser	
Description	Modifies a local user.
Required arguments	<i><userName></i> - The unique name of the user.
Optional arguments	<code>--fullUserName <the real world name of the user></code> <code>--email <the associated user email address></code> <code>--password <secret value used to identify an account for a particular user></code> <code>--passwordFile <path to password file, if --password is also specified, --passwordFile overrides its value in the command line></code>
Usage example	<code>cmtool modifyUser ec123 --fullUserName "Mary Smith"</code>

removeGroupMember	
Description	Deletes a user name from a specific group member list.
Required arguments	<i><groupName></i> - The unique name of the group. <i><userName></i> - The unique name of the user.
Optional arguments	None
Usage example	<code>cmtool removeGroupMember DevGroupA ec123</code>

setBuildEndNotification	
Description	Enables/disables notification when builds of this class end for the currently logged-in user.
Required arguments	< <i>buildClassId</i> > - A unique number assigned by the Cluster Manager for each build class. Use <code>getBuildClasses</code> to retrieve a list of build class IDs. < <i>enabled</i> > - Set this to true to enable notification and to false to disable it.
Optional arguments	None
Usage example	<code>cmtool setBuildEndNotification 1 true</code> Enables build 'end notification' for build class 1.

setPermissions																															
Description	Creates or modifies permissions for a user or group. The permissions are a space-separated list of permission names.																														
Required arguments	< <i>principalType</i> > - Value = user or group. < <i>entityName</i> > - A user or group name in an access entry. < <i>permissions</i> > - The list of permission flags for a particular entity.																														
Optional arguments	None																														
Available permission flags	<table border="1"> <thead> <tr> <th colspan="3">Permission names</th> </tr> </thead> <tbody> <tr> <td>AgentsDelete</td> <td>EMakeImpersonate</td> <td>ReportsDelete</td> </tr> <tr> <td>AgentsRead</td> <td>EMakeInvoke</td> <td>ReportsRead</td> </tr> <tr> <td>AgentsWrite</td> <td>MaintenanceDelete</td> <td>ReportsWrite</td> </tr> <tr> <td>BuildsDelete</td> <td>MaintenanceRead</td> <td>ResourcesDelete</td> </tr> <tr> <td>BuildsRead</td> <td>MaintenanceWrite</td> <td>ResourcesRead</td> </tr> <tr> <td>BuildsWrite</td> <td>MessageLogDelete</td> <td>ResourcesWrite</td> </tr> <tr> <td>ClassesDelete</td> <td>MessageLogRead</td> <td>ServerAccess</td> </tr> <tr> <td>ClassesRead</td> <td>MessageLogWrite</td> <td>UserModify</td> </tr> <tr> <td>ClassesWrite</td> <td></td> <td></td> </tr> </tbody> </table>	Permission names			AgentsDelete	EMakeImpersonate	ReportsDelete	AgentsRead	EMakeInvoke	ReportsRead	AgentsWrite	MaintenanceDelete	ReportsWrite	BuildsDelete	MaintenanceRead	ResourcesDelete	BuildsRead	MaintenanceWrite	ResourcesRead	BuildsWrite	MessageLogDelete	ResourcesWrite	ClassesDelete	MessageLogRead	ServerAccess	ClassesRead	MessageLogWrite	UserModify	ClassesWrite		
Permission names																															
AgentsDelete	EMakeImpersonate	ReportsDelete																													
AgentsRead	EMakeInvoke	ReportsRead																													
AgentsWrite	MaintenanceDelete	ReportsWrite																													
BuildsDelete	MaintenanceRead	ResourcesDelete																													
BuildsRead	MaintenanceWrite	ResourcesRead																													
BuildsWrite	MessageLogDelete	ResourcesWrite																													
ClassesDelete	MessageLogRead	ServerAccess																													
ClassesRead	MessageLogWrite	UserModify																													
ClassesWrite																															
Usage example	<code>cmtool setPermissions user ec123 "BuildsRead AgentsRead"</code> Restricts user ec123 to read-only privileges for builds and agents.																														

setUserSettings	
Description	Updates settings for the currently logged-in user.
Required arguments	< <i>watchMessages</i> > - Indicates whether you want to receive notifications when messages of the specified notification level arrive. Values are Y, N, y, n, Yes, No, yes, or no.
Optional arguments	--notificationLevel < <i>value can be either Info, Warning, or Error</i> >
Usage example	<pre>cmtool setUserSettings true --notificationLevel Info</pre> <p>Sets the current user to receive notifications for 'Info' level messages.</p>

