Framework Management Layer User's Guide

How to Use the mlcmd Utility

5/3/2025

How to Use the mlcmd Utility

Contents

You can use the **mlcmd** command-line utility to :

- Query the status of hosts, applications, or solutions.
- Start, stop, and gracefully stop applications and solutions.
- Send a custom command to an application.

Installing the Utility

If you installed Solution Control Server 8.5.100.41 and earlier, the **mlcmd** utility is already installed, and is located in the same folder in which Solution Control Server was installed.

Staring with SCS 8.5.100.42 release, the **mlcmd** utility is not installed by default with Solution Control Server. You can install it separately by following the instructions in the section "Solution Control Server Utilities' of the *Framework Deployment Guide*. After the utilities are installed, the **mlcmd** utility is stored in the location you specified during the installation.

Starting in 8.1.2, you can install the Solution Control Server utilities without installing Solution Control Server itself.

Using the Utility

All **mlcmd** command parameters are made in a single command. The general syntax is as follows:

```
mlcmd <mandatory parameters> <optional parameters> <command parameter>
```

You must provide all the mandatory parameters and one operation parameter. The parameters are listed in the following table. Starting in release 8.1, you must authenticate yourself with **mlcmd** by logging in to the utility. If authentication is successful, you can use the utility as part of operations. The parameters are listed in the following table.

> **Important**
>
> The **mlcmd** utility does not support issuing the `switchapp` command to Configuration Server Proxy.

[+] Show table

Parameter	Description
COMMON PARAMETERS	
-help	Prints the version of the utility and its usage.
-cshost *hostname*	Mandatory. Configuration Server host name.

Parameter	Description
-csport *portnumber*	Mandatory. Configuration Server port number.
-csuser *username*	Mandatory. Username of user.
-cspassword *password*	Mandatory. The password of the user.
-csappname *application_name*	Mandatory. Application name of the interface used to access Management Framework (that is, Genesys Administrator or Genesys Administrator Extension).
-scshost *SCS_host_name*[a]	Optional. Solution Control Server host name.
-scsport *SCS_port_name*[a]	Optional. Solution Control Server port number.
-timeout *seconds*	Optional. Specifies the amount of time (in seconds) that the utility will wait for SCS to perform the requested action. If SCS does not respond within the specified time, the utility will return an error. Minimum value permitted = 10 seconds. Default value = 60 seconds. If the specified value is less than 10, the command will use the default value.
-secure	Optional. Specifies that a secure connection should be used by clients when connecting to SCS.
-cert *certificate*	Optional. Use only if -secure is used. On Windows, specifies the security certificate thumbprint; on UNIX, specifies the path to the host's security certificate.
-key *key*	Use only if **-secure** is used. On UNIX, specifies the path to the file with the private key; not used on Windows.
-ca-cert *ca-cert*	Use only if **-secure** is used. On UNIX, specifies the path to the file with the CA certificate; not used on Windows.
COMMAND PARAMETERS	
WARNING! Specify only one of the following command parameters and any associated sub-parameters in a single invocation of the utility command.	
-getallalarms	Requests the object type, DBID, status, DBID of the Alarm Condition, time it was triggered, time it expires, and message text of all active alarms.
-getallappstatus	Requests the DBID, status, and runmode of all applications.
-getappstatus *Application Name* | *DBID* [-usedbid]	Requests the status of the application specified by *Application Name* (the default), or the *DBID* if usedbid is specified.
-getappstatus-runmode *Application Name* | *DBID* [-usedbid]	Requests the status and runmode of the application specified by **Application Name** (the default), or the **DBID** if usedbid is specified.
-gethoststatus *Host Name* | *DBID* [-usedbid]	Requests the status of the host specified by *Host Name* (the default), or the *DBID* if usedbid is

Parameter	Description
	specified.
-getsolstatus *Solution Name* |*DBID* [-usedbid]	Requests the status of the solution specified by *Solution Name* (the default), or the *DBID* if usedbid is specified.
-clear-app-alarms *Application Name* | *DBID* [-usedbid]	Clears all active alarms raised by the Application specified by the *Application Name* (the default), or the *DBID* if usedbid is specified.
-clear-cond-alarms (-dbid *Alarm Condition DBID*) | (-name *Alarm Condition Name*)	Clears all active alarms raised on behalf of the Alarm Condition with a DBID specified by *Alarm Condition DBID* or with the name specified by *Alarm Condition Name*.
-startapp *Application Name* | *DBID* [-usedbid]	Starts the application with the specified *Application Name* (the default), or the *DBID* if usedbid is specified.
-start-initapp *Application Name* | *DBID* [-usedbid]	Starts the application with the specified *Application Name* (the default), or the *DBID* if usedbid is specified, without waiting for the status to change to Initializing/Running.
-stopapp *Application Name* | *DBID* [-usedbid]	Stops the application with the specified *Application Name* (the default), or the *DBID* if usedbid is specified.
-switchapp *Application Name* | *DBID* [-usedbid]	Switches over the application with the specified *Application Name* (the default), or the *DBID* if usedbid is specified.
-stopapp-graceful *Application Name* | *DBID* [-usedbid]	Gracefully stops the application with the specified *Application Name* (the default), or the *DBID* if usedbid is specified.
-startsol *Solution Name* | *DBID* [-usedbid]	Starts the solution with the specified *Solution Name* (the default), or the *DBID* if usedbid is specified.
-stopsol *Solution Name* | *DBID* [-usedbid]	Stops the solution with the specified *Solution Name* (the default), or the *DBID* if usedbid is specified.
-stopsol-graceful *Solution Name* | *DBID* [-usedbid]	Gracefully stops the solution with the specified *Solution Name* (the default), or the *DBID* if usedbid is specified.
-get-app-performance {*Application Name* | *DBID* [-usedbid]} [-result *xml file name*]	Requests information about a given process of an application with the specified *Application Name* (the default), or the *DBID* if usedbid is specified. The information, including CPU Usage for each thread, is stored in an XML file named *application name*_Performance_*time stamp*.xml, if *xml file name* is not specified, or <xml file name>.xml if it is.
CUSTOM COMMAND PARAMETERS	
Name | *DBID* [-usedbid]	Sends custom commands to the application with the specified *Application Name* (the default), or the *DBID* if usedbid is specified. Must be used with one or both of the **-custom-subcommand**

Parameter	Description
	and **-custom-data parameters**.
-custom-subcommand *subcommand* where *subcommand* is a number -custom-data *path* where *path* is the path to a data file	Use only with the **-custom-command** option. Both of these options specify the content of a custom data packet as follows: • If the **-custom-subcommand** option is used, the *subcommand* value is converted to a 4 B value (network byte order) and added to the custom data packet. • If the **-custom-data** option is used, the contents of the file given by *path* is added to the custom data packet. The custom data packet is then sent to the application with a custom command.
-custom-response *path*	Use only with the **-custom-command** option. Specifies the path to the file in which the response to the custom command is to be stored. If this parameter is not used, the response is discarded.
-custom-print-response	Use only with the **-custom-command** option. Specifies that the first 4 B of the custom response must be converted from network byte order into a decimal format and sent to the standard output.

[a] If **-scshost** and **-scsport** are not specified, and if authentication is successful, **mlcmd** will retrieve the SCS connection parameters from Configuration Server. If there is more than one SCS, the utility will select the first SCS it finds, and retrieve the host and port information from the application object for that SCS.

Utility Output

For all parameters, the utility returns a numeric code when it has finished, regardless of whether execution was successful. This code can then be used in downstream processing as necessary. See Return Codes for a full list of return codes.

If any errors occur when processing this utility, a log message is generated and sent to **stderr**. Output is never sent to **stdout** unless the **-help** or **-custom-print-response** parameters are specified.

The parameter **get-app-performance** enables you to output CPU Usage data in an XML file. See XML Data Output for Information Query.

Return Codes

A zero-value (0) or a positive two-digit return code indicates that processing was completed successfully. If the command included one of the parameters used to retrieve the status of a host, application, or solution, the return code indicates the status. A negative return code (or on UNIX, a positive value in the range of 247 to 255) indicates that processing did not complete successfully. Success and failure return codes are given in the following tables.

[+] Show tables

Success Codes Returned by mlcmd Utility

Parameter Used	Code	Description
-gethoststatus	0	Host status is UNKNOWN
	1	Host status is DISCONNECTED
	2	Host status is RUNNING
	3	Host status is UNAVAILABLE
	4	Host status is UNREACHABLE
-getappstatus	0	Application status is UNKNOWN
	1	Application status is STOPPED
	2	Application status is STOP_TRANSITION
	3	Application status is STOP_PENDING
	4	Application status is START_TRANSITION
	5	Application status is START_PENDING
	6	Application status is RUNNING
	7	Application status is INITIALIZING
	8	Application status is SERVICE_UNAVAILABLE
	9	Application status is SUSPENDING
	10	Application status is SUSPENDED
-getsolstatus	0	Solution status is UNKNOWN
	1	Solution status is STOPPED
	2	Solution status is STOP_PENDING
	3	Solution status is START_PENDING
	4	Solution status is STOP_TRANSITION
	5	Solution status is START_TRANSITION
	6	Solution status is RUN_PENDING

	7	Solution status is RUNNING
	8	Solution status is STARTED
-getappstatus-runmode	0	Execution completed successfully
	The code returned by -getappstatus-runmode is a combination of the application status and the runmode. See the following table "Codes Returned Using -getappstatus-runmode Parameter" to separate the two elements.	
All others not mentioned previously	0	Execution completed successfully

Codes Returned Using -getappstatus-runmode Parameter

	0	1	2	3	4	5	6	7	8	9	10
RunMode	**Application Status (see -get_appstatus in the table above)**										
0 (PRIMARY)	0[a]	1	2	3	4	5	6	7	8	9	10
1 (BACKUP)	32	33	34	35	36	37	38	39	40	41	42
2 (EXIT)	64	65	66	67	68	69	70	71	72	73	74

[a] Execution completed successfully.

Error Codes Returned by mlcmd Utility

All but UNIX	UNIX	
Code		**Description**
-1	255	Cannot connect to Configuration Server and/or SCS.
-2	254	Unexpected disconnection from Configuration Server and/or SCS.
-3	253	Timeout expired.
-4	252	No specified object found.
-5	251	Management Layer cannot execute the operation.
-6	250	Incorrect parameters specified.
-7	249	Internal utility error.
-8	248	Command execution failed.
-9	247	User does not have correct permissions to execute the command.

XML Data Output for Information Query

When you use the **get-app-performance** command, the results are stored in an XML file named as

follows:

- Default: **<*application name*>_Performance_<timestamp>.xml**
- User specifies name in command: **<*User-specified name*>.xml**

The format of the record in the XML file is as follows:

[+] Show format

```
<AppName>                          <!—The Application name-->
  <getprocessinfo>
    <process>
      <pid>94236</pid>
      <execname>MessageServer.exe</execname>
      <vmsize>4243456</vmsize>
      <pctcpu>0.00</pctcpu>
      <priority>normal</priority>
      <cmdline>cmdline</cmdline>
      <threads>
      <!-- This section can be absent if application is on host with old version of LCA -->
        <thread>
          <tid>1st thread ID</tid>
          <pctcpu>1st thread CPU usage</pctcpu>
        </thread>
        <thread>
          <tid>2nd thread ID</tid>
          <pctcpu>2nd thread CPU usage</pctcpu>
        </thread>
        ...
        <thread>
          <tid>Nth thread ID</tid>
          <pctcpu>Nth thread CPU usage</pctcpu>
        </thread>
      </threads>
    </process>
  <getprocessinfo>
  <ProcSizeInBytes/>
</AppName>
```