



CLI Commands

This chapter describes the Cisco Connected Mobile Experiences (Cisco CMX) CLI commands:

- [cmxctl Commands, page 1-1](#)
- [cmxctl config Commands, page 1-18](#)
- [cmxos Commands, page 1-26](#)



Note

In the context of Cisco CMX, in order to access the CLI, use either the console to the virtual machine (VM) or an appliance or the Secure Shell (SSH) to the server when it has IP connectivity.

You can use the following options to manipulate the output and to get help for the commands:

- **-v** or **--verbose**—Provides a verbose output.
- **-q** or **--quiet**—Suppresses all output.
- **--help**—Shows the help text available in the system, for example, **cmxos --help**.

cmxctl Commands

This section lists the **cmxctl** commands that you can use to perform debug, enable, and start tasks:

- [cmxctl checklogs, page 1-2](#)
- [cmxctl debug, page 1-3](#)
- [cmxctl disable, page 1-4](#)
- [cmxctl dump, page 1-5](#)
- [cmxctl enable, page 1-6](#)
- [cmxctl jobs, page 1-7](#)
- [cmxctl node, page 1-9](#)
- [cmxctl restart, page 1-10](#)
- [cmxctl start, page 1-11](#)
- [cmxctl status, page 1-13](#)
- [cmxctl stop, page 1-15](#)
- [cmxctl version, page 1-17](#)

cmxctl checklogs

To check logs and generate a report, use the **cmxctl checklogs** command.

cmxctl checklogs

Syntax Description This command has no arguments or keywords.

Command Default None

Usage Guidelines After a report is generated, the specific log that shows the error can be viewed for additional details. For example, /opt/cmx.var.log/cmxjobs.log.3 has 108 errors, use the command **more /opt/cmx.var.log/cmxjobs.log.3** to view the corresponding file.

Examples The following example shows how to check logs and generate a report:

```
[root@server]# cmxctl checklogs

*****
Checking /opt/cmx/var/log/cmxjobs.log.3 for errors..
/opt/cmx/var/log/cmxjobs.log.3 has 108 errors
*****
Checking /opt/cmx/var/log/system-cron.log for errors..
/opt/cmx/var/log/system-cron.log has 0 errors
*****
Checking /opt/cmx/var/log/cmxjobs.log for errors..
/opt/cmx/var/log/cmxjobs.log has 81 errors
*****
Checking /opt/cmx/var/log/collectd.log for errors..
/opt/cmx/var/log/collectd.log has 0 errors
*****
Checking /opt/cmx/var/log/consul.log for errors..
/opt/cmx/var/log/consul.log has 0 errors
*****
Checking /opt/cmx/var/log/qless-py-worker.log for errors..
/opt/cmx/var/log/qless-py-worker.log has 0 errors
*****
Checking /opt/cmx/var/log/influxdb.log for errors..
/opt/cmx/var/log/influxdb.log has 0 errors
*****
Checking /opt/cmx/var/log/cmxjobs.log.4 for errors..
/opt/cmx/var/log/cmxjobs.log.4 has 108 errors
*****
```

cmxctl debug

To create a debug tarball in the current directory, use the **cmxctl debug** command.

cmxctl debug

Syntax Description	This command has no arguments or keywords.
Command Default	None
Usage Guidelines	The debug tarball that is created will be approximately 300 MB in size, and takes at 90 seconds to complete.
Examples	The following example shows how to create a debug tarball in the current directory:

```
[root@server]# cmxctl debug

running locally
Dumping debug information...
[localhost] Executing task 'dump_config'
cp: cannot stat `/opt/cmx/share/upgrade.answers': No such file or directory
[localhost] Executing task 'dump_state'
running 'ps aux'
running 'ifconfig -a'
running 'cmxctl status'
running 'ulimit -a'
running 'ps -u root,postgres -o %cpu,%mem,cmd'
running 'netstat -o -n -a'
running 'df -h'
running 'ntpdate -d 172.19.28.250'
running 'consul members'
[localhost] Executing task 'dump_apis'
getting /api/config/v1/clusters
getting /api/config/v1/nodes
[localhost] Executing task 'dump_hosts'
pinging configuration.service.consul
pinging location.service.consul
pinging 6379.cache.service.consul
pinging 6380.cache.service.consul
pinging 6381.cache.service.consul
pinging database.service.consul
pinging analytics.service.consul
pinging halo.service.consul
Done.
```

cmxctl disable

To disable a service, use the **cmxctl disable** command.

```
cmxctl disable { analytics | agent | cache_6379 | cache_6380 | cache_6381 | cassandra |
configuration | confd | consul | database | haproxy | location | matlabengine | metrics |
nmsplb | influxdb | iodocs | qlesspyworker }
```

Syntax Description		
	analytics	Performs analytics on calculated location data.
	agent	Manages Cisco CMX system lifecycle. starts, stops, and monitors all the services running in Cisco CMX.
	cache_6379	Caches the service used by location service.
	cache_6380	Caches the service used by analytics service.
	cache_6381	Caches the service used by analytics service.
	cassandra	Enables cassandra database service used by the location service for historical data.
	configuration	Configures nodes and clusters.
	confd	Internal service.
	consul	Internal service.
	database	Enables the database service used by analytics and configuration service.
	haproxy	Enables the TCP or HTTP load balancer gateway to all service APIs.
	location	Enables location service to compute location.
	matlabengine	Provides access point heatmap for location service.
	metrics	Collects system metrics.
	nmsplb	Enables the load balancer service used for distributing Network Mobility Services Protocol (NMSP) messages to location services.
	influxdb	Enables database services used for storing statistics from various services.
	iodocs	Enables online document service for REST API offered by various services.
	qlesspyworker	Internal service.

Command Default None

Examples The following example shows how to disable the cassandra database service:

```
[root@server]# cmxctl disable cassandra

Done
The nodeagent service is currently running with PID: 31776
Stopping cassandra process...
Done
Successfully shutdown cassandra Process.
```

cmxctl dump

To create a configuration tarball in the current directory, use the **cmxctl dump** command.

cmxctl dump

Syntax Description This command has no arguments or keywords.

Command Default None

Examples The following example shows how to create a configuration tarball in the current directory:

```
[root@server]# cmxctl dump

running locally
Dumping configuration information...
[localhost] Executing task 'dump_config'
Done.
```

cmxctl enable

To enable a service, use the **cmxctl enable** command.

```
cmxctl enable {analytics | agent | cache_6379 | cache_6380 | cache_6381 | cassandra |
configuration | confd | consul | database | haproxy | location | matlabengine | metrics |
nmsplb | influxdb | iodocs | qlesspyworker}
```

Syntax Description

analytics	Performs analytics on calculated location data.
agent	Manages Cisco CMX system lifecycle. starts, stops, and monitors all the services running in Cisco CMX.
cache_6379	Caches the service used by location service.
cache_6380	Caches the service used by analytics service.
cache_6381	Caches the service used by analytics service.
cassandra	Enables cassandra database service used by the location service for historical data.
configuration	Configures nodes and clusters.
confd	Internal service.
consul	Internal service.
database	Enables the database service used by analytics and configuration services.
haproxy	Enables the TCP or HTTP load balancer gateway to all service APIs.
location	Enables location service to compute location.
matlabengine	Provides access point heatmap for location service.
metrics	Collects system metrics.
nmsplb	Enables the load balancer service used for distributing Network Mobility Services Protocol (NMSP) messages to location services.
influxdb	Enables database services used for storing statistics from various services.
iodocs	Enables online document service for REST API offered by various services.
qlesspyworker	Internal service.

Command Default

None

Examples

The following example shows how to enable analytics service:

```
[root@server]# cmxctl enable analytics

The nodeagent service is not running.
Agent is not running, starting it now.
Starting nodeagent Process...
Retrying..
Done
Started nodeagent service with PID: 31027
```

cmxctl jobs

To configure recurring background jobs, use the **cmxctl jobs** command.

cmxctl jobs {**cancel** *jobname* | **list** | **run** *jobname* | **runnow** *jobname*}

Syntax Description		
cancel		Cancels a scheduled job.
<i>jobname</i>		Name of the job.
list		Lists all the scheduled jobs.
run		Runs a job at a specified time.
runnow		Triggers a one-time run of the job.

Command Default None

Usage Guidelines The Apache Cassandra database stores location history data. Pruning should be performed to maintain disk usage. Cisco CMX 10.2 introduces the option to prune database size. The default disk-pruning task runs at an interval of 90 days.

You can also use the **cmxctl jobs runnow cleanupcassandra** command to run an on-demand job of cleaning up the Cassandra database, which is a normal scheduled task that runs once every two days.

Examples The following example shows how to run a background job:

```
[root@server]# cmxctl jobs run LocationIndexCleanup

submitted the job, verify using cmxctl jobs list.
```

cmxctl metrics notification

To generate notification metrics for a Cisco Connected Mobile Experiences (Cisco CMX) file, use the **cmxctl metrics notification** command.

cmxctl metrics notification

Syntax Description This command has no keywords or arguments.

Command Default None

Usage Guidelines The **notification** keyword provides metrics for notification.

Examples The following example shows how to generate metrics for a Cisco CMX file:

```
[root@server]# cmxctl metrics notification
```

```
+-----+-----+-----+-----+-----+
-----+
|           EndPoint           | Success | Failure | SuccessRate |
FailureRate |
+=====+=====+=====+=====+=====+
=====+
+-----+-----+-----+-----+-----+
-----+
```


cmxctl node

To manage node installation, use the **cmxctl node** command.

```
cmxctl node {addswap | configure / install | reinstall / sslmode {disable | enable} / uninstall |
upgrade | verify}
```

Syntax Description		
addswap		Adds a 10 GB swap space to the node.
configure		Confirms node specifications.
install		Installs Cisco Connected Mobile Experiences (Cisco CMX) for the first time.
reinstall		Reinstalls the existing installation.
sslmode disable		Disables Secure Sockets Layer (SSL).
sslmode enable		Enables SSL.
uninstall		Uninstalls the software.
upgrade		Upgrades the Cisco CMX from a URL or a file. Upgrading using a URL consists of two steps: <ol style="list-style-type: none"> 1. Downloading the file (Red Hat Package Manager [RPM])—Requires HTTP or FTP link to download the image. 2. Installing the RPM.
verify		Verifies node configuration.

Command Default None

Examples The following example shows how to manage node installation:

```
[root@server]# cmxctl node sslmode enable

enabling ssl
ssl enabled
```

cmxctl restart

To restart a Cisco Connected Mobile Experiences (Cisco CMX) service, use the **cmxctl restart** command.

```
cmxctl restart { analytics | agent | cache_6379 | cache_6380 | cache_6381 | cassandra |
configuration | confd | consul | database | haproxy | location | matlabengine | metrics |
nmsplb | influxdb | iodocs | qllesspyworker }
```

Syntax Description		
	analytics	Performs analytics on calculated location data.
	agent	Manages Cisco CMX system lifecycle. starts, stops, and monitors all the services running in Cisco CMX.
	cache_6379	Caches the service used by location service.
	cache_6380	Caches the service used by analytics service.
	cache_6381	Caches the service used by analytics service.
	cassandra	Enables cassandra database service used by the location service for historical data.
	configuration	Configures nodes and clusters.
	confd	Internal service.
	consul	Internal service.
	database	Enables the database service used by analytics and configuration services.
	haproxy	Enables the TCP or HTTP load balancer gateway to all service APIs.
	location	Enables location service to compute location.
	matlabengine	Provides access point heatmap for location service.
	metrics	Collects system metrics.
	nmsplb	Enables the load balancer service used for distributing Network Mobility Services Protocol (NMSP) messages to location services.
	influxdb	Enables database services used for storing statistics from various services.
	iodocs	Enables online document service for REST API offered by various services.
	qllesspyworker	Internal service.

Command Default None

Examples

The following example shows how to restart a Cisco CMX service:

```
[root@server bin]# cmxctl restart database

Done
The nodeagent service is currently running with PID: 16718
Stopping postgres Process...
Successfully shutdown postgres Process.
Starting postgres Process...
Done
Started postgres service with PID: 25702
Exception while notifying CE
```

cmxctl start

To start a Cisco Connected Mobile Experiences (Cisco CMX) service, use the **cmxctl start** command.

```
cmxctl start { analytics | agent | cache_6379 | cache_6380 | cache_6381 | cassandra |
configuration | confd | consul | database | haproxy | location | matlabengine | metrics |
nmsplb | influxdb | iodocs | qlesspyworker }
```

Syntax Description		
	analytics	Performs analytics on calculated location data.
	agent	Manages Cisco CMX system lifecycle. starts, stops, and monitors all the services running in Cisco CMX.
	cache_6379	Caches the service used by location service.
	cache_6380	Caches the service used by analytics service.
	cache_6381	Caches the service used by analytics service.
	cassandra	Enables cassandra database service used by the location service for historical data.
	configuration	Configures nodes and clusters.
	confd	Internal service.
	consul	Internal service.
	database	Enables the database service used by analytics and configuration services.
	haproxy	Enables the TCP or HTTP load balancer gateway to all service APIs.
	location	Enables location service to compute location.
	matlabengine	Provides access point heatmap for location service.
	metrics	Collects system metrics.
	nmsplb	Enables the load balancer service used for distributing Network Mobility Services Protocol (NMSP) messages to location services.
	influxdb	Enables database services used for storing statistics from various services.
	iodocs	Enables online document service for REST API offered by various services.
	qlesspyworker	Internal service.

Command Default None

Examples The following example shows how to start a Cisco CMX service:

```
[root@server]# cmxctl start consul

Done
The nodeagent service is currently running with PID: 16718
Done
The analytics service is already running with pid: 1099
Done
Exception while notifying CE
Done
The location service is already running with pid: 16005
Done
Exception while notifying CE
```

```
Done
The configuration service is already running with pid: 16165
Done
Exception while notifying CE
Done
The matlabengine service is already running with pid: 1251
Done
Exception while notifying CE
Done
The nmsplb service is already running with pid: 1377
Done
Exception while notifying CE
```

cmxctl status

To view the status of one or all Cisco Connected Mobile Experiences (Cisco CMX) services, use the **cmxctl status** command.

```
cmxctl status [analytics | agent | cache_6379 | cache_6380 | cache_6381 | cassandra |
configuration | confd | consul | database | haproxy | location | matlabengine | metrics |
nmsplb | influxdb | iodocs | qlesspyworker]
```

Syntax Description		
	analytics	Performs analytics on calculated location data.
	agent	Manages Cisco CMX system lifecycle. starts, stops, and monitors all the services running in Cisco CMX.
	cache_6379	Caches the service used by location service.
	cache_6380	Caches the service used by analytics service.
	cache_6381	Caches the service used by analytics service.
	cassandra	Enables cassandra database service used by the location service for historical data.
	configuration	Configures nodes and clusters.
	confd	Internal service.
	consul	Internal service.
	database	Enables the database service used by analytics and configuration services.
	haproxy	Enables the TCP or HTTP load balancer gateway to all service APIs.
	location	Enables location service to compute location.
	matlabengine	Provides access point heatmap for location service.
	metrics	Collects system metrics.
	nmsplb	Enables the load balancer service used for distributing Network Mobility Services Protocol (NMSP) messages to location services.
	influxdb	Enables database services used for storing statistics from various services.
	iodocs	Enables online document service for REST API offered by various services.
	qlesspyworker	Internal service.

Command Default None

Examples

The following example shows how to display the status for the consul service:

```
[root@server]# cmxctl status consul

Done
The nodeagent service is currently running with PID: 16718
+-----+-----+-----+-----+
| Host          | Service        | Status  | Uptime (HH:mm) |
+-----+-----+-----+-----+
| cmx-master-1 | Analytics      | Running | 0 days, 05:17  |
+-----+-----+-----+-----+
| cmx-master-1 | Cache_6379     | Running | 0 days, 05:17  |
+-----+-----+-----+-----+
```

```

| cmx-master-1 | Cache_6380 | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Cache_6381 | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Cassandra | Running | 0 days, 04:08 |
+-----+-----+-----+-----+
| cmx-master-1 | Confd | Running | 0 days, 03:40 |
+-----+-----+-----+-----+
| cmx-master-1 | Configuration | Running | 0 days, 03:43 |
+-----+-----+-----+-----+
| cmx-master-1 | Consul | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Database | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Haproxy | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Influxdb | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Iodocs | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Location | Running | 0 days, 03:43 |
+-----+-----+-----+-----+
| cmx-master-1 | Matlabengine | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Metrics | Running | 0 days, 05:17 |
+-----+-----+-----+-----+
| cmx-master-1 | Nmsplb | Running | 0 days, 05:16 |
+-----+-----+-----+-----+
| cmx-master-1 | Qlesspyworker | Running | 0 days, 05:23 |
+-----+-----+-----+-----+

```

cmxctl stop

To shut down a Cisco Connected Mobile Experiences (Cisco CMX) service, use the **cmxctl stop** command.

```
cmxctl stop {analytics | agent | cache_6379 | cache_6380 | cache_6381 | cassandra |
configuration | confd | consul | database | haproxy | location | matlabengine | metrics |
nmsplb | influxdb | iodocs | qlesspyworker}
```

Syntax Description		
	analytics	Performs analytics on calculated location data.
	agent	Manages Cisco CMX system lifecycle. starts, stops, and monitors all the services running in Cisco CMX.
	cache_6379	Caches the service used by location service.
	cache_6380	Caches the service used by analytics service.
	cache_6381	Caches the service used by analytics service.
	cassandra	Enables cassandra database service used by the location service for historical data.
	configuration	Configures nodes and clusters.
	confd	Internal service.
	consul	Internal service.
	database	Enables the database service used by analytics and configuration services.
	haproxy	Enables the TCP or HTTP load balancer gateway to all service APIs.
	location	Enables location service to compute location.
	matlabengine	Provides access point heatmap for location service.
	metrics	Collects system metrics.
	nmsplb	Enables the load balancer service used for distributing Network Mobility Services Protocol (NMSP) messages to location services.
	influxdb	Enables database services used for storing statistics from various services.
	iodocs	Enables online document service for REST API offered by various services.
	qlesspyworker	Internal service.

Command Default The services are running.

Examples The following example shows how to stop the analytics service:

```
[root@server]# cmxctl stop analytics

Done
The nodeagent service is currently running with PID: 16987
Stopping analytics Process...
Service analytics with pid: 19095
Retrying..
Done
Successfully shutdown analytics Process.
```

cmxctl users

To list or to configure Cisco Connected Mobile Experiences (Cisco CMX) users using the CLI, use the **cmxctl users** command.

cmxctl users {list | passwd *username*}

Syntax Description	list	Lists all the current users.
	passwd	Sets the password for a user.
	<i>username</i>	Username of a user in Cisco CMX.

Command Default None

Examples The following example shows how to list Cisco CMX users using the CLI:

```
[root@server]# cmxctl users list

+-----+-----+-----+
| Username | Full Name | Roles |
+-----+-----+-----+
| monitor | Monitor User | Read Only |
+-----+-----+-----+
| admin   | Admin User | Admin |
+-----+-----+-----+
```


cmxctl version

To know the Cisco Connected Mobile Experiences (Cisco CMX) version, use the **cmxctl version** command.

cmxctl version

Syntax Description This command has no arguments or keywords.

Command Default None

Examples The following example shows how to display version information for Cisco CMX:

```
[root@server]# cmxctl version

Build Version   : 10.1.0-27
Build Time      : 2015-05-05 03:06:45.437430
-----
Name            : cmx-ng-container
Commit Count    : 17
Short Hash      : bf20ec1
-----
Name            : cmx-ng-location
Commit Count    : 5
Short Hash      : efc84fa
-----
Name            : cmx-ng-ui
Commit Count    : 5
Short Hash      : d793df7
-----
Name            : cmx-ova
Build Time      : Fri Feb 20 06:34:38 UTC 2015
-----
```

cmxctl config Commands

This section lists the **cmxctl config** commands:

- [cmxos backup, page 1-28](#)
- [cmxctl config controllers, page 1-19](#)
- [cmxctl config import, page 1-20](#)
- [cmxctl config maps, page 1-21](#)
- [cmxctl config reload, page 1-22](#)
- [cmxos restore, page 1-35](#)
- [cmxctl config sma, page 1-23](#)
- [cmxctl config verify, page 1-24](#)

cmxctl config controllers

To manage Cisco Wireless Controllers (Cisco WLC), use the **cmxctl config controllers** command.

cmxctl config controllers {add | delete | import | show}

Syntax Description	add	Adds a Cisco WLC.
	delete	Deletes a Cisco WLC.
	import	Imports a Cisco WLC from Cisco Prime Infrastructure by providing the corresponding credentials, or by placing an exported Cisco Prime Infrastructure MAP file in the /opt directory of the Cisco CMX server and providing the path to the exported MAP file.
	show	Shows information pertaining to a Cisco WLC.

Command Default None

Usage Guidelines After a Cisco WLC is added, the following message is displayed: “controller added successfully”. Note that this refers only to the correct parsing of the command. You should issue a **cmxctl controllers show** command to ensure that the Cisco WLC is not active.

Examples The following example shows how to display the Cisco WLC information:

```
[root@server]# cmxctl config controllers show
```

```
+-----+-----+-----+-----+-----+-----+
| IP Address | Type | Version | Device Version | SHA2 | Status |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.65 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.44 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.46 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.70 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.93 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.97 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.35 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.58 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.82 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.84 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
| 30.30.30.53 | WLC | 8.0.72.141 | - | No | ACTIVE |
+-----+-----+-----+-----+-----+-----+
```

cmxctl config import

To imports a map and Cisco Wireless Controller (Cisco WLC) from Cisco Prime Infrastructure, use the **cmxctl config import** command.

cmxctl config import {prime | status}

Syntax Description

prime	Imports maps from Cisco Prime Infrastructure.
status	Shows import status.

Command Default

None

Examples

The following example shows how to import a map and Cisco WLC from Cisco Prime Infrastructure:

```
[root@server]# cmxctl config import prime
```

```
Please enter PI ip address: x.x.x.x
Please enter PI username [root]: root
Please enter PI password [Public123]:
```

```
Import successfully started from PI x.x.x.x. Check import status using cmxctl config
import status.
```

cmxctl config maps

To import and manage maps, use the **cmxctl config maps** command.

cmxctl config maps {address | delete | import}

Syntax Description	address	Imports addresses for the maps.
	delete	Deletes the campus map.
	import	Imports map from the Cisco Prime Infrastructure.

Command Default None.

Examples

The following example shows how to import and manage maps:

```
[root@server]# cmxctl config maps import
```

```
Please specify import type [PI / FILE] [FILE]: PI
Please enter PI ip address: x.x.x.x
Please enter PI username [root]: root
Please enter PI password [Public123]:
```

```
Import successfully started from PI 173.37.206.3. Check import status using cmxctl config
import status.
```

cmxctl config reload

To forcefully generate a configuration file, use the **cmxctl config reload** command.

cmxctl config reload

Syntax Description This command has no arguments or keywords.

Command Default None

Examples The following example shows how to forcefully generate a configuration file:

```
[root@server]# cmxctl config reload

2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: WARNING Skipping confd config file.
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/analytics.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/cassandra/cassandra-env.sh in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/cassandra/cassandra.yaml in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/collectd.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/configuration.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/connect.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/halo.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/haproxy.cfg in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/influxdb.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/location.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/matlabengine.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/nmsplb.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/nmspproxy.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/postgresql.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/redis_6379.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/redis_6380.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: INFO Target config
/opt/cmx/etc/redis_6381.conf in sync
2015-03-10T17:45:50Z cmx-vmdev117 -verbose[17174]: ERROR template:
redis.template.conf:15:20: executing "redis.template.conf" at <getv ($tag | printf ...>:
error calling getv: key does not exist
```

cmxctl config sma

To manage social media analytics (SMA), use the **cmxctl config sma** command.

cmxctl config sma {proxy | twitter}

Syntax Description		
	proxy	Sets HTTP proxy.
	twitter	Sets Twitter credentials.

Command Default None

Usage Guidelines After setting the SMA proxy, restart the qlesspyworker for the changes to take effect.

Examples The following example shows how to manage SMA:

```
[root@server]# cmxctl config sma proxy
```

```
Please enter value for http proxy: http://proxy.cisco.com:80
```

```
Please enter value for https proxy: http://proxy.cisco.com:80
```

```
SMA Proxies have been set. Please restart qlesspyworker for the changes to take effect.
```

cmxctl config verify

To verify the Cisco Connected Mobile Experiences (Cisco CMX) installation and configuration, use the **cmxctl config verify** command.

cmxctl config verify

Syntax Description This command has no arguments or keywords.

Command Default None

Examples The following example shows how to verify the Cisco CMX installation and configuration:

```
[root@server]# cmxctl config verify

Verifying node configuration...
NetworkManager: unrecognized service
Consul v0.4.1
Consul Protocol: 2 (Understands back to: 1)
confd 0.6.0
-----+-----+-----+-----+
--+
| module          | check              | passed  | msg
|
-----+-----+-----+-----+
==+
| netman_stopped | NetworkManager service is not | Success  |
|                | running            |          |
|
-----+-----+-----+-----+
--+
| matlabengine   | http://matlabengine.service.co | Failed   | check the log files
under |
|                | nsul:5577/api/services/matlabengine/status |         | /opt/cmx/var/log
|
-----+-----+-----+-----+
--+
| database       | connect to database port:5432 | Success  |
|
-----+-----+-----+-----+
--+
| consul_dns     | 127.0.0.1 (consul) is present | Success  |
|                | as dns server in          |         |
|                | /etc/resolv.conf          |         |
|
-----+-----+-----+-----+
--+
| etchost_hacks  | consul service hostnames not | Success  |
|                | static in /etc/hosts       |         |
|
```



```

+-----+-----+-----+-----+
--+
| analytics      | http://analytics.service.consul | Failed   | check the log files
under |
|                | 1:5556/api/services/analytics/ |          | /opt/cmx/var/log
|                | status                          |          |
|                |                                  |          |
+-----+-----+-----+-----+
--+
| hostname_ping  | ping to hostname:cmx-master-1   | Success  |
|                |                                  |          |
+-----+-----+-----+-----+
--+
| location       | http://location.service.consul | Failed   | check the log files
under |
|                | :5555/api/services/location/st |          | /opt/cmx/var/log
|                | atus                            |          |
|                |                                  |          |
+-----+-----+-----+-----+
--+
| confd_installed | Confd is installed              | Success  |
|                |                                  |          |
+-----+-----+-----+-----+
--+
| consul_installe | Consul is installed              | Success  |
| d              |                                  |          |
|                |                                  |          |
+-----+-----+-----+-----+
--+
| nmsplb        | http://nmsplb.service.consul:6 | Failed   | check the log files
under |
|                | 001/api/services/nmsplb/status |          | /opt/cmx/var/log
|                |                                  |          |
+-----+-----+-----+-----+
--+
| configuration  | http://configuration.service.c | Failed   | check the log files
under |
|                | onsul:6000/api/services/config |          | /opt/cmx/var/log
|                | uration/status                  |          |
|                |                                  |          |
+-----+-----+-----+-----+
--+
| cassandra     | connect to cassandra port:9042 | Success  |
|                |                                  |          |
+-----+-----+-----+-----+
--++

```

cmxos Commands

This section lists the **cmxos** commands:

- [cmxos addswap](#), page 1-27
- [cmxos backup](#), page 1-28
- [cmxos configure](#), page 1-30
- [cmxos firstboot](#), page 1-31
- [cmxos fixhaproxy](#), page 1-32
- [cmxos openports](#), page 1-33
- [cmxos reconfigure](#), page 1-34
- [cmxos restore](#), page 1-35
- [cmxos upgrade](#), page 1-36
- [cmxos verify](#), page 1-38

cmxos addswap

To add a 10 GB space to the operating system, use the **cmxos addswap** command.

cmxos addswap

Syntax Description This command has no arguments or keywords.

Command Default None

Usage Guidelines This command should be run at the root user level.

Examples The following example shows how to increase disk space in the operating system:

```
[root@server]# cmxos addswap

10485760+0 records in
10485760+0 records out
10737418240 bytes (11 GB) copied, 29.6845 s, 362 MB/s
Setting up swapspace version 1, size = 10485756 KiB
no label, UUID=2734f069-e687-4635-b2d6-9381241bc7ee
swap added, run system info to verify
[root@cmx-vmdev146 ~]#
```

cmxos backup

To back up a node, use the **cmxos backup** command.

cmxos backup

Syntax Description	This command has no arguments or keywords.
Command Default	None
Usage Guidelines	This command should to be run using the cmxadmin (non-root) account.
Examples	The following example shows how to back up a node:

```
[root@server]# cmxos backup

Please enter the path for backup file [/tmp]:
[17:43:50] Preparing for backup...
[17:43:50] Backup Database...
[17:43:51] Backup Cache...
[17:43:51] Backup Cassandra...
[17:43:53] Backup InfluxDb...
[17:43:53] Backup Consul...
[17:43:53] Backup Floormaps...
[17:43:53] Backup node configuration...
[17:43:59] Creating tar file..
[17:43:59] Done Backup. Created backup file
/tmp/cmx_backup_cmx-vmdev117_2015_03_10_17_43.tar.gz
?[]0;root@cmx-vmdev117:~[root@cmx-vmdev117 ~]# cmxos restore
?[]1034hPlease enter the backup file path:
/tmp/cmx_backup_cmx-vmdev117_2015_03_10_17_43.tar.gz
[17:44:12] Preparing for restore...
[17:44:12] Untarring backup file...
[17:44:13] Stopping all services...
[17:44:16] Restoring Database...
Restarting database...
[17:44:26] Restoring Cache...
Stopping cache_6379...
Restarting cache_6379...
Stopping cache_6381...
Restarting cache_6381...
Stopping cache_6380...
Restarting cache_6380...
[17:44:55] Restoring Cassandra...
Stopping Cassandra...
Restarting Cassandra...
.....
[17:45:19] Restoring Influxdb...
[17:45:19] Restoring consul...
[17:45:19] Restoring floormaps...
[17:45:19] Running Post Restore Tasks...
[17:45:19] Migrating Schemas...
[17:45:19] Migrating Cassandra schemas...
[17:45:20] Restarting all services...
```

[17:45:23] Done

cmxos configure

To configure the network and operating system parameters, use the **cmxos configure** command.

cmxos configure

Syntax Description This command has no arguments or keywords.

Command Default None

Usage Guidelines This command should to be run at the root user level. You can use the *--force* option to force a fresh configuration if the device is already configured.

Examples The following example shows how to configure the network and operating system parameters:

```
[root@server]# cmxos configure --force

*** The system is already configured

*****
Checking if the machine meets required specification...
*****

+-----+-----+-----+-----+
| Check |      expected      |   actual   | Result |
+-----+-----+-----+-----+
| memory | 8GB                | 25GB       | ?      |
+-----+-----+-----+-----+
| cpu    | 4                  | 8          | ?      |
+-----+-----+-----+-----+
| disk   | 50GB               | 51GB       | ?      |
+-----+-----+-----+-----+
| hostname | rfc compliant hostname | cmx-vmdev146 | ?      |
+-----+-----+-----+-----+
```

cmxos firstboot

To set up the Cisco Connected Mobile Experiences (Cisco CMX) again, use the **cmxos firstboot** command.

cmxos firstboot

Syntax Description This command has no arguments or keywords.

Command Default None

Usage Guidelines This command should be run at the root user level. You can use the `--force` option to force a fresh configuration if the device is already configured.

Examples The following example shows how to set up Cisco CMX again:

```
[root@server]# cmxos firstboot
Not first boot....Exiting...
```

cmxos fixhaproxy

To verify the HAproxy permissions on Cisco Connected Mobile Experiences (Cisco CMX), use the **cmxos fixhaproxy** command.

cmxos fixhaproxy

Syntax Description This command has no arguments or keywords.

Command Default None

Usage Guidelines This command should be run at the root user level.

Examples The following example shows how to verify HAproxy permissions:

```
[root@server]# cmxos fixhaproxy
Raising haproxy setcap...
```


cmxos openports

To open ports, based on a node rule, use the **cmxos openports** command.

cmxos openports {analytics | location | database}

Syntax Description		
	analytics	Adds a 10-GB swap space to a node.
	location	Configures the network and operating system parameters.
	database	Sets up the Cisco Connected Mobile Experiences (Cisco CMX) database again.

Command Default None

Usage Guidelines This command should be run at the root user level.

Examples The following example shows how to open ports based on a node:

```
[root@server]# cmxos openports analytics

Opened port 6541
Opened port 6542

Successfully opened all ports. Saving iptables info...
```

cmxos reconfigure

Network configuration information can be changed after deployment by connecting to the Cisco Connected Mobile Experiences (Cisco CMX) CLI and entering the **cmxos reconfigure** command.

cmxos reconfigure

Syntax Description

This command has no arguments or keywords.

Command Default

None

Usage Guidelines

This command, which should be run at the root user level, also allows you to change the IP address, netmask, default gateway, and DNS server information.

Examples

The following example shows how to reconfigure the network after Cisco CMX installation:

```
[root@server]# cmxos reconfigure
```

**Note**

This command opens the Device Configuration window, where you can take the appropriate action, that is reconfigure the device or the DNS.

cmxos restore

To restore a node, use the **cmxos restore** command.

cmxos restore

Syntax Description		
	--file PATH	Path where the backup file is located.
	--path DIRECTORY	Path where the backup file will be created.
	-i, --include_only TEXT	Restore selected parts only. Options are database, cache, cassandra, influxdb, consul, floormaps, licenses, setup .
	--HELP	Shows the help content.

Command Default None

Examples

The following example shows how to restore a node:

```
[root@server]# cmxos restore

Please enter the backup file path: /tmp/cmx_backup_cmx-vmdev117_2015_03_10_17_43.tar.gz
[17:44:12] Preparing for restore...
[17:44:12] Untarring backup file...
[17:44:13] Stopping all services...
[17:44:16] Restoring Database...
Restarting database...
[17:44:26] Restoring Cache...
Stopping cache_6379...
Restarting cache_6379...
Stopping cache_6381...
Restarting cache_6381...
Stopping cache_6380...
Restarting cache_6380...
[17:44:55] Restoring Cassandra...
Stopping Cassandra...
Restarting Cassandra...
.....
[17:45:19] Restoring Influxdb...
[17:45:19] Restoring consul...
[17:45:19] Restoring floormaps...
[17:45:19] Running Post Restore Tasks...
[17:45:19] Migrating Schemas...
[17:45:19] Migrating Cassandra schemas...
[17:45:20] Restarting all services...
[17:45:23] Done
```

cmxos upgrade

To upgrade Cisco Connected Mobile Experiences (Cisco CMX) with a new Red Hat Package Manager (RPM) or package, use the **cmxos upgrade** command.

cmxos upgrade

Syntax Description	This command has no arguments or keywords.
Command Default	None
Usage Guidelines	This command should be run at the root user level. The CLI accepts either a local file or an HTTP URL. This command works only when you have a later version than the existing one to upgrade.
Examples	The following example shows how to upgrade the Cisco CMX using RPM or package:

```
[root@server]# cmxos upgrade

The nodeagent service is not running.
Agent is not running, starting it now.
Starting nodeagent Process...

Stopping nodeagent Process...
Done
Successfully shutdown nodeagent Process.
Stopping consul Process...
Successfully shutdown consul Process.
Stopping qllesspyworker Process...
Successfully shutdown qllesspyworker Process.
Stopping cassandra Process...
Successfully shutdown cassandra Process.
Stopping iodocs Process...
The iodocs service is not running.
Stopping redis6383 Process...
Successfully shutdown redis6383 Process.
Stopping redis6380 Process...
Successfully shutdown redis6380 Process.
Stopping redis6381 Process...
Successfully shutdown redis6381 Process.
Stopping influxdb Process...
The influxdb service is not running.
Stopping collectd Process...
The collectd service is not running.
Stopping confd Process...
The confd service is not running.
Stopping redis6379 Process...
Successfully shutdown redis6379 Process.
Stopping redis6378 Process...
Successfully shutdown redis6378 Process.
Stopping haproxy Process...
Stopping postgres Process...
Successfully shutdown postgres Process.
Stopping analytics Process...
```

```
The analytics service is not running.  
Stopping location Process...  
The location service is not running.  
Stopping configuration Process...  
The configuration service is not running.  
Stopping halo Process...  
The halo service is not running.  
Stopping matlabengine Process...  
The matlabengine service is not running.  
Stopping nmsplb Process...  
The nmsplb service is not running.  
Shutting down
```

cmxos verify

To verify the virtual machine configuration, use the **cmxos verify** command.

cmxos verify

Syntax Description This command has no arguments or keywords.

Command Default None

Examples The following example shows how to verify the virtual machine configuration:

```
[root@server]# cmxos verify
```

Check	expected	actual	Result
memory	8GB	25GB	?
cpu	4	8	?
disk	50GB	51GB	?
hostname	rfc compliant hostname	cmx-vmdev146	?