Oracle® Enterprise Manager
Command Line Interface

13c Release 4

F23463-10

April 2022

ORACLE®

Contents

Preface

1   EM CLI Overview and Concepts

2   Downloading and Deploying EM CLI

3 Using EM CLI

4   Advanced EM CLI Script Examples

5   Verb Reference

6   Error Code Reference

A   Sample Scripts

Preface

This manual provides a verb reference, which duplicates and enhances the command-line help, for the Enterprise Manager Command Line Interface (EM CLI). This manual also covers concepts, downloading, deploying, and scripting.

Audience

This guide is written for Enterprise Manager administrators who want to perform operations remotely or script them. The reader should already be familiar with Oracle Enterprise Manager.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1
EM CLI Overview and Concepts

This section provides the following topics:

- Overview
- EM CLI Modes of Operation
- EM CLI Architecture

Overview

The Enterprise Manager Command Line Interface (EM CLI) enables users to access Enterprise Manager functionality through a command-line interface or scripts. It is accessible through classic programming language constructs, enabling tasks to be created and run either from the command-line or programatically. EM CLI enables you to access Enterprise Manager Cloud Control functionality from text-based consoles (shells and command-line windows) for a variety of operating systems.

EM CLI is fully integrated with Enterprise Manager's security and user administration functions, enabling you to carry out operations using EM CLI with the same security and confidentiality as the Enterprise Manager Cloud Control console. For example, you can only see and operate on targets for which you are authorized.

Examples of EM CLI tasks you can accomplish are as follows:

- Create a new Enterprise Manager administrator account.
- Monitor and manage targets, jobs, groups, and blackouts.
- Enable batch/complex tasks on multiple Agents or targets.
- Integrate Enterprise Manager with third-party or custom software through scripts. Actions that are part of a customer's business model can be performed through scripts.

EM CLI Modes of Operation

EM CLI offers the following modes of operation:

- **Standard mode**

  In Standard mode, each EM CLI verb entered is a single operating system command. Each command launches EM CLI, executes the command, then terminates.

- **Interactive mode**

  Interactive mode is ideal for adhoc queries or commands for real-time diagnostics or debugging. In this mode, EM CLI is started as a shell and all commands entered on the command line are executed immediately to enable several commands to be executed at will in the same shell.

- **Script mode**

  In Script mode, an administrator can create a single Python script that includes a sequence of EM CLI commands to be executed with a single invocation.

Each mode uses the same verbs. A verb is a task or action in the form of a user command which exposes Enterprise Manager functionality. Some verbs can include one or more parameters, which are arguments to the specified command. Some of the parameters are required and some are optional.In the following examples of the `create_group` verb syntax, only the `-name` parameter is required. The other parameters are optional.

- Standard mode example

```
$ emcli create_group -name="name"
      [-type=<group>]
      [-add_targets="name1:type1;name2:type2;..."]
      [-is_propagating="true/false";
```

- Interactive mode example

```
$ emcli
emcli> … other commands…
emcli> create_group(name="name"
      [;type=<group>]
      [;add_targets="name1:type1;name2:type2;..."]
      [;is_propagating="true/false");
emcli> … other commands …
emcli> exit()
$
```

- Script mode example

```
$ emcli @create_group.py;
```

Standard Command-line Mode

This is the traditional and exclusive mode prior to Enterprise Manager Cloud Control version 13.1.0.0. This mode provides a simple command-line interface to Enterprise Manager, and supports the execution of one verb at a time from the command line.

For example:

```
 emcli create_group -name=my_group -add_targets="mymachine.myco.com:host"
```

Interactive mode

This mode enables you to create a single interactive session with the server (Oracle Management Services), where you can type in commands, view the output, and potentially respond to or manipulate the output. Interactive mode opens a Jython shell, where you can provide Jython scripts using EM CLI verbs as Jython functions. Jython is a Java implementation of the Python programming language.

Note that when calling a verb in Interactive mode, the arguments are placed inside parentheses. For example:

```
 emcli> create_group(name='my_group'..)
```

Script Mode

Script mode is especially effective when performing tasks in bulk mode or many tasks at once. Scripts are useful for accomplishing several tasks, including:

- Listing or setting global target properties

- Listing or setting Agent properties
- Updating database passwords
- Listing group members

This mode enables you to create Jython scripts, store them as files, and then pass these files to EM CLI as an argument, such as ...

```
emcli @createuser.py
```

... where createuser.py is the name of a file containing the Python code to be sent to EM CLI.

You can create reusable, functional modules using existing EM CLI verbs to generate complex tasks. This intuitive, object-oriented programming model supports encapsulation, loops, functions, exception and error handling, and so forth. These abilities enable you to benefit from all of the powerful features that the Jython programming language offers.

> ✎ **See Also:**
>
> For more information about using the different modes, see Using EM CLI .

EM CLI Architecture

Figure 1–1 shows the high-level architecture of EM CLI.

Figure 1-1   EM CLI Architecture
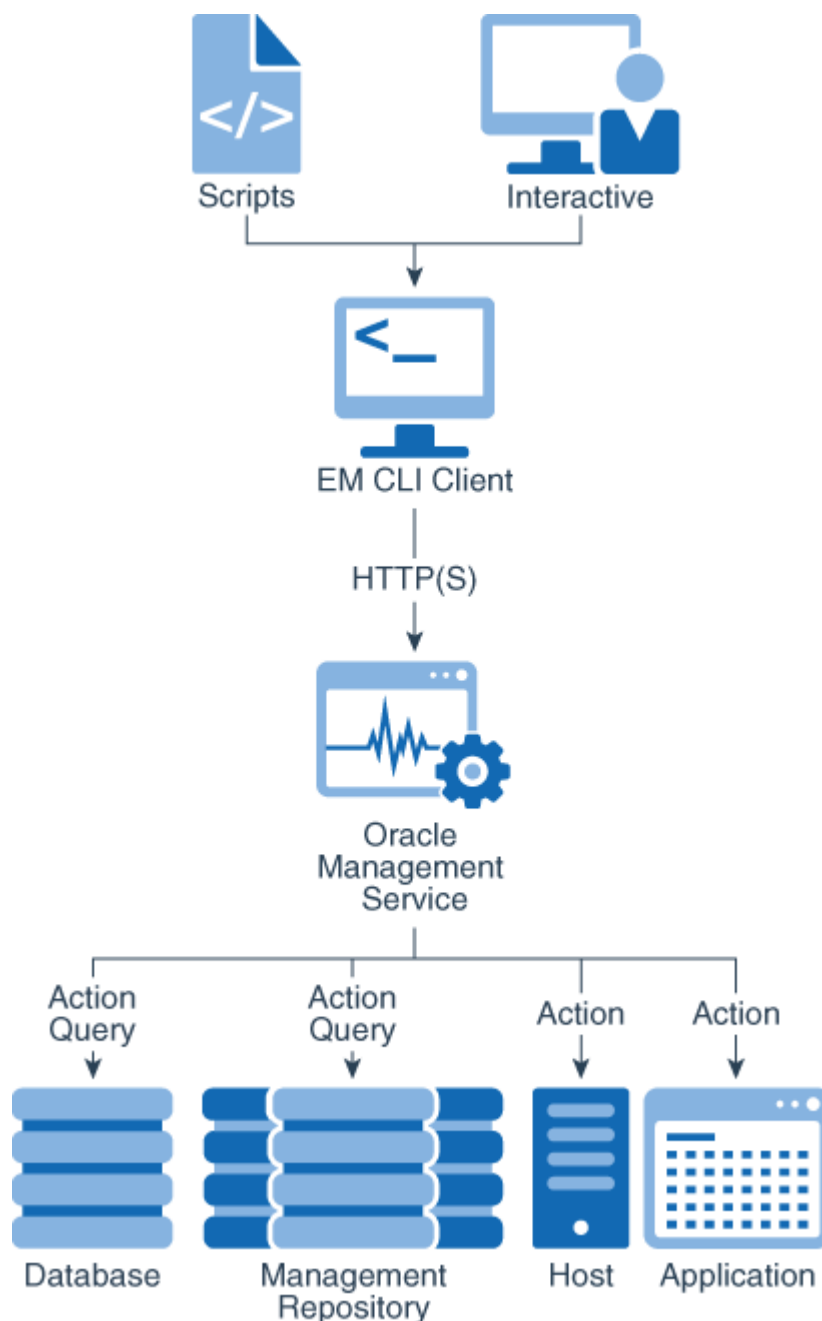
EM CLI implements client-server architecture, in which EM CLI is the client, and Oracle Management Services (OMS) is the server.

A typical verb may take zero or more arguments as input. The EM CLI client passes the input to OMS for processing. The EM CLI client connects to OMS and establishes a user session, which is used across verb executions until a logout is initiated.

2
Downloading and Deploying EM CLI

This section discusses the following Enterprise Manager Command Line Interface (EM CLI) topics:

- EM CLI Installation
- Downloading and Deploying the EM CLI Client
- Getting Started with EM CLI
- Security and Authentication

EM CLI Installation

EM CLI provides two installable kits:

- **EM CLI Standard**

  This kit supports the Standard mode only.

- **EM CLI with Scripting mode**

  This kit supports all three modes, but only Interactive and Scripting modes enable you to provide Jython-based scripts.

EM CLI consists of two components used to access the Enterprise Manager framework functionality:

- EM CLI client

  The EM CLI client is a command-line program (Sun Java JRE-based) that sends EM CLI verbs to a specific Oracle Management Service (OMS). In some respects, the EM CLI client functions as a command-line equivalent of an Enterprise Manager Cloud Control console. You can download the EM CLI client on any system within your managed network.

- EM CLI Oracle Management Services (OMS)

  The EM CLI OMS is automatically installed with the OMS and serves as the communication conduit between the EM CLI client and the OMS.

You can download the EM CLI client on any system within your managed network. The EM CLI client is a command-line program (Sun Java JRE-based) that sends EM CLI verbs to a specific Oracle Management Service (OMS). In some respects, the EM CLI client functions as a command-line equivalent of an Enterprise Manager Cloud Control console. The EM CLI OMS is automatically installed with the OMS and serves as the communication conduit between the EM CLI client and the OMS.

For instructions about setting up and running EM CLI, see Downloading and Deploying EM CLI .

> **Note:**
>
> EM CLI only supports Oracle Java. EM CLI does not support JRockit JVM, OpenJDK, or other versions of JDK.

Downloading and Deploying the EM CLI Client

The EM CLI OMS is automatically installed with the OMS, but you must download and set up the EM CLI client portion. The following instructions cover download procedures for the EM CLI client. The EM CLI client kits are available for public access, so do not require authentication.

As mentioned in EM CLI Overview and Concepts , the EM CLI client features two kits: EM CLI Standard and EM CLI with the Script option. The EM CLI Script option includes the Jython Interpreter for Jython script support (described in Using EM CLI), as well as all of the features present in the EM CLI Standard kit.

The following sections explain how to download and deploy these two kits.

Requirements

Before downloading the EM CLI client, ensure that the following system requirements have been met:

- Enterprise Manager 13c Cloud Control framework
- Sun Java JRE version 1.7.0_111 or greater
- Workstation running Solaris, Linux, HPUX, Tru64, AIX, or Windows with NTFS

> **Note:**
>
> EM CLI does not support JRockit JVM.

> **Note:**
>
> If you are using an MD5-based signature algorithm, you need to overwrite the *java.security* file to switch from the OMS JDK to your client JDK location

Downloading and Deploying the EMC CLI Client for Standard EM CLI

To download the EM CLI client for standard EM CLI:

1. Obtain the standard EM CLI client kit emclikit.jar using one of the following methods:

   - Download this kit from any 13.1.0.0 or later Cloud Control installation at the following location:

     ```
     https://<your_em_host:port>/em/public_lib_download/emcli/kit/emclikit.jar
     ```

For example:

```
wget --no-check-certificate
https://<your_em_host:port>/em/public_lib_download/emcli/kit/emclikit.jar
```

- Download this kit from the Cloud Control console:

  – From the Setup menu, select **Command Line Interface**.

  – In the EM CLI Standard section, click the **Download the EM CLI Standard Kit to your workstation** link.

2. Set your `JAVA_HOME` environment variable and ensure that it is part of your `PATH`. You must be running Sun Java JRE 1.7.0_111 or greater. For example:

Linux platform:

```
setenv JAVA_HOME /usr/local/packages/j2sdk1.7.0_111
```

Windows platform:

```
C:\Users>set JAVA_HOME=C:\Program Files\Java\jdk1.7.0_111
```

3. Install EM CLI Standard kit into any directory using emclikit.jar. The directory in which EM CLI is installed is called "EM CLI Home" (or "EM CLI Client Directory").

- For Enterprise Manager Cloud Control version 13.1.0.0 and later —

  On a Linux platform, enter:

  ```
  $JAVA_HOME/bin/java -jar emclikit.jar -install_dir=<em_cli_home_dir>
  ```

  On a Windows platform, enter:

  ```
  %JAVA_HOME%\bin\java -jar emclikit.jar -install_dir=<em_cli_home_dir>
  ```

- For Enterprise Manager Cloud Control versions prior to 13.1.0.0 —

  On a Linux platform, enter:

  ```
  $JAVA_HOME/bin/java -jar emclikit.jar client
  -install_dir=<em_cli_home_dir>
  ```

  On a Windows platform, enter:

  ```
  %JAVA_HOME%\bin\java -jar emclikit.jar client
  -install_dir=<em_cli_home_dir>
  ```

4. Change directories to the <em_cli_home_dir> directory where EM CLI is installed, then execute emcli help setup for instructions on how to use the setup verb to configure the EM CLI client for a particular OMS.

For information on configuring for shared directories environments, see Using EM CLI With Shared Directories.

Downloading and Deploying the EM CLI Client with the Script Option

> **Note:**
>
> Before proceeding, click here to see a video tutorial on how to download and get started with EM CLI in Interactive Mode, and click here to see a video tutorial on how to download and get started with EM CLI in Script Mode.
>
> Read the readme.txt file shipped with the EM CLI kit with Scripting mode.

By default, the EM CLI client is set up in secure mode. In this mode, EM CLI does not store any Enterprise Manager or SSO passwords on the EM CLI client disk. The command emcli setup -noautologin sets up the EM CLI client in secure mode. By default, -noautologin is true. Therefore, you do not need to specify it if you want to set up the EM CLI client in secure mode. In secure mode, if the EM CLI session times out due to inactivity, explicit login (using the login verb) is required before invoking any verb.

If you want to set up EM CLI in the insecure auto-login mode, you can use the emcli setup -autologin command. In this mode, if an EM CLI session times out due to inactivity, EM CLI automatically re-establishes the session when a verb needs to execute. However, if you explicitly logged out by running emcli logout, you need to explicitly log in again using emcli login.

For more information, see Secure Mode for the EM CLI Setup.

- If EM CLI is set up with the -autologin option, the script executes as the auto logged-in user. In the script, you can directly call the verb as a function without having to use login() in it.

- If EM CLI is set up without the -autologin option, the login() function has to be used. If the password is not passed as an argument in the script, you are prompted for the password during script execution.

- If EM CLI is set up with the -trustall option, EMCLI_TRUSTALL or EMCLI_CERT_LOC is not required.

To download the EM CLI client for standard EM CLI as well as Interactive and Script EM CLI:

1. Obtain the EM CLI client kit emcliadvancedkit.jar using one of the following methods:

   - Download this kit from any 13.1.0.0 or later Cloud Control installation at the following location:

     ```
     https://<your_em_host:port>/em/
     public_lib_download/emcli/kit/emcliadvancedkit.jar
     ```

     For example:

     ```
     wget --no-check-certificate
     https://<your_em_host:port>/em/
     public_lib_download/emcli/kit/emcliadvancedkit.jar
     ```

   - Download this kit from the Cloud Control console:

- From the Setup menu, select **Command Line Interface**.

- In the EM CLI with Script Option section, click the **Download the EM CLI with Script option kit to your workstation** link.

2. Set your `JAVA_HOME` environment variable and ensure that it is part of your `PATH`. You must be running Sun Java JRE 1.7.0_111 or greater. For example:

Linux platform:

```
setenv JAVA_HOME /usr/local/packages/j2sdk1.7.0_111
```

Windows platform:

```
C:\Users>set JAVA_HOME=C:\Program Files\Java\jdk1.7.0_111
```

3. Install EM CLI with Scripting mode into any directory using emcliadvancedkit.jar. The directory in which EM CLI is installed is called "EM CLI Home" (or "EM CLI Client Directory").

   • For Enterprise Manager Cloud Control version 13.1.0.0.0 and later —

   On a Linux platform, enter:

   ```
   $JAVA_HOME/bin/java -jar emcliadvancedkit.jar
   -install_dir=<em_cli_home_dir>
   ```

   On a Windows platform, enter:

   ```
   %JAVA_HOME%\bin\java -jar emcliadvancedkit.jar
   -install_dir=<em_cli_home_dir>
   ```

   • For Enterprise Manager Cloud Control versions prior to 13.1.0.0.0 —

   On a Linux platform, enter:

   ```
   $JAVA_HOME/bin/java -jar emcliadvancedkit.jar client
   -install_dir=<em_cli_home_dir>
   ```

   On a Windows platform, enter:

   ```
   %JAVA_HOME%\bin\java -jar emcliadvancedkit.jar client
   -install_dir=<em_cli_home_dir>
   ```

4. Change directories to the <em_cli_home_dir> directory where EM CLI is installed, then execute emcli help sync for instructions on how to use the sync verb to configure the EM CLI client for a particular OMS.

> **Note:**
>
> By default, EM CLI with Scripting mode does not store any user session information on disk. It is tailored to build production-grade Jython modules for Enterprise Manager.
>
> Read the readme.txt file shipped with the EM CLI kit with Scripting mode.

ORACLE

Using EM CLI With Shared Directories

To avoid contention issues when different Enterprise Manager users are accessing the same EM CLI directories, the following configuration is suggested:

1. Set the EMCLI_OPTS environment variable as shown in the following example, using the export Linux operating system command:

    ```
    export EMCLI_OPTS="-Duser.home=/home/user/cli -Demcli.state.dir=/home/user/
    cli"
    ```

    Since this is an environment variable, you can set this permanently in your session, depending on your operating system.

2. Invoke the setup command, noting the following recommendations:

    *   Use a different EM CLI state directory per user by defining the directory location with the -dir option.

    *   Use a different verb jars directory per user by defining the directory location with the -verb_jars_dir option.

    For example:

    ```
    $EMCLI_INSTALL_HOME/emcli setup
            -url=https://omsmachine.example.com:em_port/em
            -username="admin"
            -dir="/home/user/cli"
            -verb_jars_dir="/home/user/cli"
    ```

Troubleshooting After Installation

If you receive the following error when you try to execute your first EM CLI verb, EM CLI tried and failed to acquire a lock for a file:

```
Error: Some required configuration is missing, corrupt, inaccessible, or
insecure (access permissions are too liberal). Resolve the problem and run setup.
```

To see how this issue occurs, run the sample java program provided in My Oracle Support Knowledge Document 1480978.1, "EMCLI fails with Some required configuration is missing.. error or Java.io.IOException: No Locks Available ":

https://support.oracle.com/rs?type=doc&id1=480978.1

The program tries to acquire a lock for a file and fails. This error is caused by an NFS problem in your environment. To work around this issue:

1. Specify -Duser.home="<*non nfs path*>" to force EM CLI to create the files under this directory.

2. Set the EMCLI_OPTS variable as follows:

    ```
    export EMCLI_OPTS ="-Duser.home=/tmp"
    ```

    For example:

    ```
    export EMCLI_OPTS=-Duser.home=/tmp/emclitemp/prod
    ```

3. Run the emcli setup command as follows:

```
emcli setup
url="https://omsmachine.example.com:em_port/em"
     -username="admin"
     -password="password1234"
     -trustall
     -dir=/tmp/emclitemp/prod
```

Getting Started with EM CLI

After the EM CLI client is downloaded and installed, you are ready to begin using EM CLI. At this point, you can run the EM CLI client out of the installation directory location, or alternatively, you can add it to your PATH.

Using Basic Operational Verbs

Immediately after installation, only basic operational verbs are available:

- **argfile** — Execute an EM CLI verb where the verb and any options are contained in a file.

- **help** — Access command-line help for EM CLI verbs.

- **login** — Log in and establish a session with the OMS.

- **logout** — Log out of EM CLI client from Enterprise Manager.

- **setup** — Configure EM CLI to function with a specific OMS.

    (See Connecting the EM CLI Client to OMS for important information about this verb.

- **status** — Show EM CLI setup details

- **sync** — Synchronize the EM CLI client with an OMS.

- **version** — List EM CLI verb versions or the EM CLI client version.

EM CLI incorporates a comprehensive command-line help system that provides various levels of assistance. Available from any EM CLI client installation, the help system provides a listing of all available verbs, descriptive overviews for each verb, syntax, as well as usage examples. The command-line help is the definitive EM CLI information source.

Using Commands in Standard Mode

To invoke a verb in standard mode, precede the verb with the emcli command. For example, to invoke the help for an overview of all available verbs, enter one of the following commands:

Linux platform:

```
./emcli help
```

Windows platform:

```
>.\emcli help
```

Alternatively, enter the same command followed by the verb name to view a detailed verb description, the verb parameters and options, and usage examples, as in:

Linux platform:

```
./emcli help login
```

Windows platform:

```
>.\emcli help login
```

> **Note:**
>
> You can execute EM CLI without using **./** for Linux or **.** for Windows if you set the PATH environment variable to the directory where EM CLI is installed.

Using Commands in Interactive Mode

To access command-line verbs in interactive mode, you must first invoke the EM CLI command prompt:

```
$>./emcli
```

To invoke a verb, call the verb name followed by parentheses. For example, enter the following command to invoke the help:

```
emcli> help()
```

To find help for a specific verb, call the help command with the verb within the parentheses surrounded by a single quote:

```
emcli>help('login')
```

> **Note:**
>
> The setup and sync commands are not available inside Script and Interactive modes.

> **Tip:**
>
> Read the readme.txt file shipped with the advanced kit for more specific examples on how to call the verbs in the EM CLI Client in Script and Interactive modes.

Calling a Script

To call a script, you must first invoke the EM CLI command prompt:

```
$>./emcli
```

To run a script, enter `emcli` and provide the script location as shown in the following example, where `my_script.py` is the full path of a valid Python script:

```
%emcli @my_script.py
```

> **✎ Note:**
>
> The setup and sync commands are not available inside Script and Interactive modes.

> **💡 Tip:**
>
> Read the readme.txt file shipped with the advanced kit for more specific examples on how to call the verbs in the EM CLI Client in Script and Interactive modes.

Connecting the EM CLI Client to OMS

You must run the setup verb to connect the EM CLI client to the OMS running the EM CLI Management Services. Running `setup` installs all available verb-associated command-line help from the EM CLI Management Service. If you have installed EMCLI with the Script option, you can use the sync command instead of the setup command.

> **✎ Note:**
>
> If you have followed the instructions in Downloading and Deploying the EM CLI Client, the set up is already done for you.

You can use one EM CLI client installation to function with multiple OMSes. However, at any time, EM CLI can function with a particular OMS. For either scenario, you need to set up the EM CLI client once for each OMS. You also need to subsequently set the EMCLI_STATE_DIR environment variable to the directory that was specified as the EM CLI client directory for the particular OMS.

To connect the EM CLI client to OMS:

1. Understand the syntax of the setup and sync verbs and their options by entering the following commands or referring to the respective verbs in Connecting the EM CLI Client to OMS:

   • Command-line EM CLI:

   ```
   ./emcli help setup
   ```

   • Script and Interactive EM CLI:

   ```
   ./emcli help sync
   ```

2. Enter the setup verb with at least the minimally required parameters as shown in This examples:

   • Command-line EM CLI:

   ```
   ./emcli setup -url=http://omsmachine.example.com:em_port/em
     -username=em_user
   ```

   • Script and Interactive EM CLI:

```
./emcli sync -url=http://omsmachine.example.com:em_port/em
  -username=em_user -trustall
```

If you have already downloaded certificates, you can specify them using the environment variable EMCLI_CERT_LOC. In this case, the -trustall option is not needed.

> **✎ Note:**
>
> Specify the URL you are using to sign in to Enterprise Manager through the browser.

As you observed from step 1, the setup verb has several options, including the following important options:

- -autologin
- -noautologin

In autologin mode, if a session times out, EM CLI automatically logs you in. In the default noautologin mode, if no EM CLI command executes within the 45-minute default session time-out period, you need to log in using the login verb to be able to execute the verbs.

3. Enter your user password for Enterprise Manager when prompted after the EM CLI client connects with the EM CLI Management Services.

After running the `setup` verb, the message "Emcli Setup Successful" appears, and you are ready to begin using EM CLI.

> **💡 Tip:**
>
> For complete information on the setup verb and its options, including autogin and noautologin referenced in step 2, see the setup verb.
>
> To configure the EM CLI client to function with multiple Oracle Management Services by implementing multiple setups, see the Examples section for the setup verb.

Configuring an HTTP Proxy Environment

If you are planning to use EM CLI through an HTTP proxy server, you need to set an additional environment variable, `EMCLI_OPTS`, that supplies EM CLI with the requisite proxy host and port information. This examples illustrate setting the `EMCLI_OPTS` environment variable for both Windows and UNIX operating systems.

Example 2-1   Setting EMCLI_OPTS in a Microsoft Windows Environment

```
>set EMCLI_OPTS=-Dhttp.proxyHost=<proxy host> -Dhttp.proxyPort=<proxy port>
```

Example 2-2   Setting EMCLI_OPTS in a UNIX Environment (TCSH)

```
>setenv EMCLI_OPTS "-Dhttp.proxyHost=<proxy host> -Dhttp.proxyPort=<proxy port>"
```

Configuring Log File Settings for EM CLI

EM CLI creates log files to record informational and error messages generated during operation. Not all of the logs in This examples are necessarily present. Logs are created as needed and are appended — they are preserved between invocations of EM CLI. You can safely delete log files any time without affecting the EM CLI operation. The logs help you to troubleshoot any run-time errors.

> **Note:**
>
> By default, .emcli.log is only created when an exception or error occurs, or when debugging is enabled. Otherwise, the file does not exist.

This examples show possible log file locations:

```
<EM_CLI_Instance_Home>/.emcli.log
<EM_CLI_Instance_Home>/.emcli.log.1
```

`<EM_CLI_Instance_Home>` refers to the directory specified by the `-dir` option in the latest running of the `setup` verb (with an appended `.emcli` sub-directory). The current `<EM_CLI_Instance_Home>` directory can be identified by executing the `status` verb to display the setup summary.

Log files are limited to a maximum of 0.5 MB. EM CLI alternates between the two log files — as each file reaches the 0.5 MB limit, EM CLI begins writing to the other file, overwriting the oldest log file after `emcli.log.1` has been filled for the first time.

Log File Locations

This example show possible log file locations:

If you do not specify a configuration directory when you run the `setup` verb (`-dir` option is omitted), EM CLI assumes the `.emcli` configuration directory is located within your local home directory. The log files are placed at the root level of the `.emcli` directory. The `.emcli` directory must be local (not mounted remotely).

In this example, the configuration directory is specified using the `-dir` option when the `setup` verb is run. This allows you to specify a local configuration directory if the user home directory is mounted remotely (through NFS, for example).

Example 2-3   No Configuration Directory Specified with Setup Verb (default location)

```
user.home/.emcli/.emcli.log
user.home/.emcli/.emcli.log.1
```

Example 2-4   Local Configuration Directory Specified with Setup Verb (-dir=<local directory>

```
local.dir/.emcli/.emcli.log
local.dir/.emcli/.emcli.log.1
```

Log File Location and Log Level

You can specify the log file directory and the log level, if desired, using the following variables, which you can set as environment variables:

- EMCLI_LOG_LOC — Sets the log file directory to any desired location.

- EMCLI_LOG_LEVEL — Presets the log level. Allowed values in descending order are:

  - SEVERE (highest level)

  - WARNING

  - INFO

  - CONFIG

  - FINE

  - FINER

  - FINEST (lowest level)

  Additionally, you can use the level OFF to turn off logging, and the level ALL to enable logging of all messages.

Security and Authentication

To enable EM CLI to function with a particular OMS, configure EM CLI by executing the setup verb. This is a one-time operation for this particular OMS.

You can find out the OMS connection information from any EM CLI client by invoking the setup verb without any options. For example:

```
$ emcli setup
Oracle Enterprise Manager Cloud Control 13c Release 2.
Copyright (c) 1996, 2016 Oracle Corporation and/or its affiliates. All rights
reserved.

Instance Home           : /private/emcli/setup/.emcli
Verb Jars Home          : /private/emcli/setup/.emcli
EM URL                  : https://myomshost.us.example.com:5416/em
EM user                 : user1
Trust all certificates  : true
Auto login              : false
```

You can also invoke the status command, which provides more information than the setup command:

```
$ emcli status
Oracle Enterprise Manager Cloud Control 13c Release 1.
Copyright (c) 1996, 2016 Oracle Corporation and/or its affiliates. All rights
reserved.

Instance Home           : /private/emcli/setup/.emcli
Verb Jars Home          : /private/emcli/setup/.emcli
Status                  : Configured
EMCLI Home              : /private/MWHome/oms/bin
EMCLI Version           : 13.2.0.0.0
Sun Java JRE Home       : /private/MWHome/jdk17/jdk
```

```
Sun Java JRE Version   : 1.7.0_111
Log file               : /private/emcli/setup/.emcli/.emcli.log
EM URL                 : https://myomshost.us.example.com:5416/em
EM user                : sysman
Auto login             : false
Trust all certificates : true
```

Example 2-5   CLI-Enterprise Manager Authentication

```
>emcli setup -url="http[s]://host:port/em" -username="<username>"  [-trustall] [-
novalidate]

>please enter password:
```

HTTPS Trusted Certificate Management

For authenticating an OMS during the SSL server authentication phase of an HTTPS connection handshake, EM CLI searches for trusted certificates in the following key stores:

```
CONFIG_DIR/.emcli/.localkeystore
user.home/.emcli/.keystore
JRE_HOME/lib/security/cacerts
```

CONFIG_DIR is the directory specified by the -dir option in the latest running of the setup verb (with an appended .emcli sub-directory).

JRE_HOME in a Sun Java JRE installation is typically JAVA_HOME/jre.

The Sun Java JRE keytool command can manage the key stores. For more information about this tool, see the security documentation for your Sun Java JRE VM installation, or at the time of this writing:

```
http://java.sun.com/j2se/1.5.0/docs/tooldocs/solaris/keytool.html
```

Not all of the key stores in the list above will necessarily be present.

Secure EM CLI Clients

You can provide credentials to EM CLI in one of two ways:

- Provide credentials at the time of use. See the login and logout verbs for information on credentials.

- Make credentials persistent on the host system where the EM CLI client is running, as might be the case when executing EM CLI verbs from a shell script.

> **⚠ Caution:**
>
> You should only persist credentials on hosts when the host is a secure EM CLI client, since the only protection available for credentials is the file-system security of the OS.
>
> Oracle also recommends not using persistent credentials if the EM CLI user's home directory is mounted over NFS or any other insecure file system.

Secure Mode for the EM CLI Setup

The EM CLI client installs certain configuration files and a client-side implementation of verbs on the EM CLI client system. The EM CLI client configuration files contain information such as the OMS URL, Enterprise Manager user names, and Enterprise Manager passwords.

By default, the EM CLI client is set up in secure mode. In this mode, EM CLI does not store any Enterprise Manager or SSO passwords on the EM CLI client disk. The command emcli setup -noautologin sets up the EM CLI client in secure mode. By default, -noautologin is true. Therefore, you do not need to specify it if you want to set up the EM CLI client in secure mode. In secure mode, if the EM CLI session times out due to inactivity, explicit login (using the login verb) is required before invoking any verb.

If you want to set up EM CLI in the insecure auto-login mode, you can use the emcli setup -autologin command. In this mode, if an EM CLI session times out due to inactivity, EM CLI automatically re-establishes the session when a verb needs to execute. However, if you explicitly logged out by running emcli logout, you need to explicitly log in again using emcli login.

- For information on the -noautologin option, see the setup verb.

- For information on logging in, see the login verb.

- For information on logging out, see the logout verb.

- > **✎ Note:**
  >
  > The following information regarding the -script option is not to be confused with the Script mode.

For easy parsing of verb output by scripts, a `-script` option is available for all verbs that generate output data. If you use the `-script` option, all output columns become tab-separated (with non-null values), and all rows become newline-separated. You can override the default column and row separators by using the `-format` option in place of `-script`.

```
[-script|-format="name:<format
type>;column_separator:<separator_text>;row_separator:<separator_text>
"]
```

Supported `-format` options are shown in Table 2-1.

Table 2-1   Supported "-format" Options

Option	Explanation
`-format="name:pretty"`	Pretty-print the output. This is the default when both `-script` and `-format` are not specified.

Table 2-1   (Cont.) Supported "-format" Options

Option	Explanation
`-format="name:script"`	Identical to just specifying `-script`. Columns are tab-separated, and rows are newline-separated.
`-format="name:script;column_separator:<column_sep_string>"`	Causes the verb output to be column-separated by `<column_sep_string>`. Rows are separated by the newline character.
`-format="name:script;row_separator:<row_sep_string>"`	Causes the verb output to be row-separated by `<row_sep_string>`. Columns are separated by the tab character.
`-format="name:script;column_separator:<column_sep_string>;row_separator:<row_sep_string>"`	Causes the verb output to be column-separated by `<column_sep_string>` and row- separated by `<row_sep_string>`.
`-format="name:csv"`	Produces a table with the columns separated by commas and the rows by newlines.

- `-script` is equivalent to `-format="name:script;column_separator:\u0009;row_separator:\u000A"`

- The values for column and row separator are specified as one or more character strings. Any of the characters can be represented by the unicode sequence `\uXXXX` (where `X` is a hex value).

  NOTE: The ASCII character set is represented by `\u00XX`, where `XX` can range from 00 to 7F. For example, the tab character is represented by `\u0009` and the newline character is represented by `\u000A`.

- The `pretty` format type has no attributes.

- In `script` mode, any verb output cells that contain the separator strings are substituted with the unicode values for these strings so that the output does not break any scripts required to parse the output.

- `script` is the only format type for which separators can be specified.

- Separators need not be single characters, and can be specified using both regular characters interspersed with unicode sequences as shown in This example:

Example 2-6   Complex Separator

Separator Specification: `xxx\u0009xxx\u0009`

This separator appears as `xxx` followed by a tab, followed by `xxx`, followed by another tab.

3

Using EM CLI

This chapter discusses the following Enterprise Manager Command Line Interface (EM CLI) topics:

- Using Command-line EM CLI
- Using EM CLI in Interactive or Script Mode
- Advanced Script Examples
- Using the Generic 'List' Verb
- Using the Registered Clients Page

Using Command-line EM CLI

Command-line EM CLI is the traditional and most direct way of invoking an EM CLI verb. The basic syntax from the system prompt is:

```
emcli verb_name -required_parameter1 -required_parameter2 ... -optional_parameter1 -
optional_parameter2 ...
```

The syntax for a particular verb applies to its usage whether it is invoked through the command line or programatically. For example, the syntax for the create_group verb is:

```
emcli create_group
      -name="name"
      [-type=<group>]
      [-add_targets="name1:type1;name2:type2;..."]...
      [-is_propagating="true/false"]
```

This indicates that the -name parameter is required, whereas the -type parameter, enclosed within brackets, is optional, as well as the -add_targets and -is_propagating parameters. This example shows how the verb might be used at the command-line prompt:

```
emcli create_group -name=db_group
      -add_targets="emp_rec:oracle_database"
      -add_targets="payroll:oracle_database"
```

Verb Categories provides the format, descriptions of required and optional parameters, and examples for most EM CLI verbs. Those that are not documented can be found in the online help by entering the following command:

```
emcli help verb_name
```

Using EM CLI in Interactive or Script Mode

As introduced in EM CLI Overview and Concepts , EM CLI provides Interactive and Script modes to enhance and extend the basic functionality offered through the standard command-line invocation. Both Interactive and Script mode provide the same functionality. Unless otherwise stated explicitly, all of the information presented in this chapter pertains to both Interactive and Script modes.

The following sections discuss the fundamental principles associated with the EM CLI interactive or scripting modes:

- Jython Interpreter

- Script and Interactive Mode Syntax

- Interactive Mode — Connecting to an Oracle Management Server (OMS)

- Examples of Standard, Interactive, and Script Verb Invocations

- Writing and Running the First Script

- Invoking an EM CLI Verb Programatically

- Error Exception Handling

- Utility Functions

- Selected Use Cases

- Selected list Verb Use Cases

> 💡 **Tip:**
>
> For a demonstration of using EM CLI in script and interactive mode, click on the links below to view the following Enterprise Manager Screenwatches:
>
> - Getting Started with EM CLI in Script Mode
>
> - Downloading and Getting Started with EM CLI in Interactive Mode

Jython Interpreter

Beginning with Enterprise Manager Cloud Control version 12cR3, EM CLI includes an embedded Jython interpreter (Jython 2.5.3), where all of the verbs are registered as functions, known as *EM CLI verb functions* or simply *functions*. Usage of these functions is similar to the corresponding verb. In these functions, the parameters (supplied as key-value pairs) are those present in the verb arguments.

In Interactive mode, the interpreter opens a shell where you can type your commands. Using Script mode, you can run your Jython program by passing it to the interpreter in a non-interactive fashion. For both modes, apart from using the EM CLI verb functions, you can also program in Jython conventionally.

Script and Interactive Mode Syntax

The syntax for these two modes varies slightly.

To run a script:

Enter `emcli` and provide the script location as shown in the following example, where `my_script.py` is the full path of a valid Python script:

```
%emcli @my_script.py
```

To start EM CLI in Interactive mode:

Enter emcli at the command prompt to start an interactive shell, as follows:

Linux platform:

```
% emcli
emcli>
```

Windows platform:

```
C:\Directory> emcli
emcli>
```

Comparing Script and Interactive Modes

To illustrate using the interpreter in both Script and Interactive mode to achieve the same objective, This examples print the current version of the installed EM CLI client. Example 3-1 shows a Python script that uses the version() verb of EM CLI to print the current version. Example 3-2 achieves the same result using the interactive shell. Note that the version verb used in both of these examples has the same signature and functionality.

Example 3-1   Script that Prints the Current Version

For a script named emcli_helloworld.py with the following contents:

```
print 'Hello EMCLI'
print version()
```

The output is:

```
Hello EMCLI
Oracle Enterprise Manager 13c EMCLI Version 13.1
```

Example 3-2   Interactive Input that Prints the Current Version

```
$emcli>print 'Hello EMCLI'
Hello EMCLI
$emcli>version()
Oracle Enterprise Manager 13c EMCLI Version 13.1
```

Interactive Mode — Connecting to an Oracle Management Server (OMS)

Because most of the verbs require a connection to an OMS, you need to set up an OMS connection in an Interactive shell before you can invoke any verb by minimally setting the following required client properties and optionally setting others:

- EMCLI_OMS_URL

- EMCLI_TRUSTALL or EMCLI_CERT_LOC

The following procedure provides a recommended method of setting up these properties and subsequently logging in.

1. Type help('client_properties') in the Interactive shell for more information about the available client properties, as shown in the following output example:

```
emcli>help('client_properties')
    EMCLI_OMS_URL            : OMS URL To connect to.
    EMCLI_USERNAME           : OMS Username.
    EMCLI_AUTOLOGIN          : Possible values are true,false. Default is false.
    EMCLI_TRUSTALL           : Possible values are true,false. Default is false.
    EMCLI_VERBJAR_DIR        : Location of bindings directory.
    EMCLI_CERT_LOC           : Location of a valid certificate file.
    EMCLI_LOG_LOC            : Directory where log files will be stored.
    EMCLI_LOG_LEVEL          : Possible values are ALL,INFO,FINE,FINER,WARN,SEVERE
```

```
                                              Default is SEVERE.
    EMCLI_OUTPUT_TYPE      : Possible values are json, JSON, text, TEXT.
Default
                         is                  json in script mode and
text in interactive mode.

status() will list values of all the client properties.
set_client_property(propertyname,value), get_client_property(propertyname),
and
clear_client_property(name)can be used to set, get, and clear a client
property
```

2. Set the required client properties from either the Interactive shell directly using the set_client_property() function, or as environment variables before a shell is launched.

- **Function method**

  For example, to connect to an OMS at https://host1.example.com:1234/em in which you want to trust all certificates:

  ```
  emcli>set_client_property('EMCLI_OMS_URL',
        'https://host1.example.com:1234/em')
  emcli>set_client_property('EMCLI_TRUSTALL','true')
  ```

- **Environment variables method**

  For example, to set the same client properties as environment variables on a Linux platform:

  ```
  % setenv EMCLI_TRUSTALL true
  % setenv EMCLI_OMS_URL https://host1.example.com:1234/em
  ```

  Windows platform:

  ```
  C:\Directory> set EMCLI_TRUSTALL=true
  C:\Directory> set EMCLI_OMS_URL=https://host1.example.com:1234/em
  ```

3. Log into the OMS:

   ```
   emcli>login(username='<user>')
   ```

4. Provide a password at the prompt. You can also alternatively provide a password as shown:

   ```
   emcli>login(username='foo', password='bar')
   ```

Alternatively Logging in with EMCLI_USERNAME

You can use the EMCLI_USERNAME client property to log in as shown in This example for the Linux platform:

```
% setenv EMCLI_USERNAME sysman
      emcli>login()
   Enter password :   ******

   Login successful
```

Windows platform:

```
C:\Directory> set EMCLI_USERNAME sysman
      emcli>login()
   Enter password :   ******

   Login successful
```

Displaying the Status of a Session

You can use the status() command to display the status of an EM CLI session, as shown in
This example:

```
emcli>status()
        <banner>

        Verb Jars Home (EMCLI_VERBJAR_DIR)  :
                                    /<Location>/int/./bindings/13.2.0.x.0/.emcli
        EM CLI Home (EMCLI_INSTALL_HOME)       : /<Location>/int/.
        EM CLI Version                         : 13.2.0.x.0
        Java Home                              : /jdk6/jre
        Java Version                           : 1.x.0_x
        Log file (EMCLI_LOG_LOC)               : CONSOLE
        Log level (EMCLI_LOG_LEVEL)            : SEVERE
        EM URL (EMCLI_OMS_URL)                 : https://host1.example.com:1234/em
        EM user (EMCLI_USERNAME)               : <user>
        Auto login (EMCLI_AUTOLOGIN)           : false
        Trust all certificates (EMCLI_TRUSTALL) : true
```

Exiting the Interactive Shell

To exit the EM CLI interactive shell, use the exit verb as shown:

```
emcli>exit()
```

Examples of Standard, Interactive, and Script Verb Invocations

This examples contrast these three methods of verb invocations.

Example 1 — String-based Arguments

Standard invocation:

```
% emcli create_user -name='jan.doe' -type='EXTERNAL_USER'
```

Interactive mode invocation:

```
emcli>create_user(name='jan.doe',type='EXTERNAL_USER')
```

Script mode invocation:

```
create_user(name='jan.doe',type='EXTERNAL_USER')
```

Example 2 — List-based Arguments

Standard invocation:

```
% emcli grant_privs -name='jan.doe' \
        -privilege="USE_ANY_BEACON" \
        -privilege="FULL_TARGET;TARGET_NAME=host1.example.com:TARGET_TYPE=host"
```

Interactive mode invocation:

```
emcli>priv_list = ['USE_ANY_BEACON',
'FULL_TARGET;TARGET_NAME=myhost.us.example.com:TARGET_TYPE=host']
```

Script invocation:

```
priv_list=['USE_ANY_BEACON',
'FULL_TARGET;TARGET_NAME=myhost.us.example.com:TARGET_TYPE=host']
```

Example 3 — Flag-based Boolean Arguments

Standard invocation:

```
% emcli get_targets -noheader
```

Interactive mode invocation:

```
emcli>get_targets(noheader=True)
```

Script invocation:

```
get_targets(noheader=True)
```

Advisory Information About Incomplete Commands in Interactive Mode

If you do not complete a command in interactive mode, the Jython interpreter prompts with three dots on the next line. Until the line is complete, Jython continues to generate this prompt. For example:

```
emcli > status(
...
...
...
...)
```

Providing the closing parenthesis executes the status command.

In This example, "\n" completes the line, and reports a syntax error.

```
emcli> get_targets -targets="oracle_database"
...
...
...\n
```

Writing and Running the First Script

To assist you in writing your first script, this section analyzes a sample script that retrieves all targets and prints their names. Example 3-3 shows the entire script.

> **Note:**
>
> Line numbers are provided only for explanatory purposes for Table 3-1. For a copy-ready script, see Example A-1 in Sample Scripts.

> **Note:**
>
> Observe the method of accessing the JSON response from the verb response get_targets().out()['data']. The get_targets() response provides a handle to the response object, and out()['data'] provides a handle over the underlying JSON data. This methodology is consistent for most verbs.

Script Analysis

Table 3-1 provides an analysis of each line of code.

Table 3-1   Line-by-Line Script Analysis

Lines	Description

4	Jython import construct to import all EM CLI verb functions in the current program. You can also selectively import the verb functions. You can use the Jython import function to import all of the functions as wildcards or on an as needed basis explicitly. For example:
	
	`from emcli import *`
	
	... imports all of the functions, whereas ...
	
	`from emcli import get_targets`
	
	... imports only the get_targets function.

6 - 10	Custom Jython function to print the name and type of a target. It accepts a key value tuple of the form. It accepts a key value tuple of the form {Target Name, Target Type} as the parameters.
13, 15	Necessary connection to OMS in order to retrieve all targets. Before connecting to the OMS, you must first set the OMS connection details using the set_client_property() function. This sets the OMS URL to https://host1.example.com:1234/em and enables the client to trust all certificates. Note that none of these details are stored in disk. These details are stored in memory and only last for a single script execution. For more information on client properties, enter help('client_properties') from the interactive shell. You can define EMCLI_OMS_URL and EMCLI_TRUSTALL variables as environment variables if you do not want to set these in your script. If you have downloaded certificates somewhere, you can also use the environment variable EMCLI_CERT_LOC to point to the certificate directory. In this case, you do not need EMCLI_TRUSTALL.
18	Login function to connect to the OMS. The example uses the Sysman user to log in. This prompts for a password during execution.
21 - 24	Invokes the get_targets() function and captures its response in an array called targets_array. This is in JSON format. This example iterates through this array and uses the custom function print_target_details to print its name and type.

Script Execution

Example 3-4 shows that executing this script retrieves the list of all targets and their types.

The Logout Successful message indicates that the login session to the OMS is closed at the end of the execution.

Example 3-3   Script That Retrieves All Targets and Prints Their Names

```
1 #emcli_get_targets.py
2
3 #Import all emcli verbs to current program
4 from emcli import *
5
6 def print_target_details(target):
7    '''
     8    print the target name and target type given a target tuple.
     9    '''
    10    print target['Target Name'] + ' ' + target['Target Type']
11
12 #Set the OMS URL to connect to
13 set_client_property('EMCLI_OMS_URL','https://host1.example.com:1234/em')
```

```
14 #Accept all the certificates
15 set_client_property('EMCLI_TRUSTALL','true')
16
17 #Login to the OMS
18 login(username='adminuser')
19
20 #Invoke get_targets and loop over the targets array
21 targets_array = get_targets().out()['data']
22 for target in targets_array:
23      #Call print_target_details function to print the target details
24    print_target_details(target)
```

Example 3-4   Output of Script that Retrieves All Targets

```
$emcli @emcli_get_targets.py
Enter password :  ******
test.example.com host
EM Management Beacon oracle_beacon
CSAcollector oracle_csa_collector
Oemrep_Database oracle_database
EM Jobs Service oracle_em_service
test.example.com:1838 oracle_emd
Management Services and Repository oracle_emrep
Management_Servers oracle_emsvrs_sys
test.example.com:7654_Management_Service oracle_oms
test.example.com:7654_Management_Service_CONSOLE oracle_oms_console
test.example.com:7654_Management_Service_PBS oracle_oms_pbs
/EMGC_EMGC_DOMAIN/EMGC_DOMAIN weblogic_domain
Logout successful
```

Invoking an EM CLI Verb Programatically

As mentioned earlier, all of the verbs are available as global Jython functions with verb options as function parameters. Flag-based options are provided by specifying True as the value. List-based options are provided by constructing a Python list and using it as an argument. Table 3-7 provides more details on this.

Accessing Verb Invocation Responses

Every EM CLI verb invocation returns a Response object. The Response object is part of EM CLI, and has the functions listed in Table 3-2.

Table 3-2   Response Object Functions

Function	Description
out()	Provides the verb execution output. The output can be text, or the JSON.isJson() method on the Response object can be used to determine whether the output is JSON. Refer to the section "JSON Processing" for more details.
error()	Provides the error text (if any) of the verb execution if there are any errors or exceptions during verb execution. Refer to the section "Error and Exception Handling" for more details.
exit_code()	Provides the exit code of the verb execution. The exit code is zero for a successful execution and non-zero otherwise. Refer to the section "Error and Exception Handling" for more details.

Table 3-2 (Cont.) Response Object Functions

Function	Description
isJson()	Provides details about the type of output. It returns True if response.out() can be parsed into a JSON object.

Example 3-5 invokes the get_targets verb and prints the output, error, and exit code of the execution.

> **Note:**
>
> Line numbers are provided only for illustrative purposes. For a copy-ready script, see Example A-2 in Sample Scripts.

Line 16 shows that instead of printing the raw response (which will be JSON), the example uses the Jython len() function to print the length of the response array, which is basically the count of all of the targets. Note that the example uses the Jython str() function to convert an integer type to a string.

Example 3-6 shows the execution of the script in Example 3-5.

Example 3-5 Script that Incorporates Functions in the get_targets Verb

```
1   #emcli_introspect_response.py
2
3   #Import all emcli verbs to current program
4   from emcli import *
5
6   #Set the OMS URL to connect to
7   set_client_property('EMCLI_OMS_URL','https://host1.example.com:1234/em')
8   #Accept all the certificates
9   set_client_property('EMCLI_TRUSTALL','true')
10
11  #Login to the OMS
12  login(username='sysman')
13
14  res = get_targets()
15
16  print 'Number of targets:'+str(len(res.out()['data']))
17  print 'Errors           :'+res.error()
18  print 'Exit code        :'+str(res.exit_code())
19  print 'IsJson           :'+str(res.isJson())
```

Example 3-6 Output of Script that Invokes the get_targets Verb

```
$emcli @emcli_introspect_response.py
Enter password :  ******
Number of targets:12
Errors          :
Exit code       :0
IsJson          :True
Logout successful
```

JSON Processing

If a verb response is JSON, it can be interactively iterated and accessed. You can use response.isJson() to check whether the verb output is JSON. If the verb output is JSON, response.out()['data'] provides the object in the Jython object model.

JSON processing has been shown in previous examples. Example 3-7 shows another example of this processing. The example uses custom SQL with the list() function, which provides a generic method to retrieve data about managed objects in Enterprise Manager. Custom SQL only works if the OMS user has super user privileges.

> **Note:**
>
> Line numbers are provided only for explanatory purposes for Table 3-3. For a copy-ready script, see Example A-3 in Sample Scripts.

Script Analysis

Table 3-3 provides an analysis of relevant lines of code. The remainder of the program is similar to Example 3-3, which was analyzed in Table 3-1.

Table 3-3   Line-by-Line Script Analysis

Lines	Description
13 - 22	A custom Jython function get_targets_with_props() returns all of the targets with a given property name and value. It uses the list() function or verb to query the targets. This verb, introduced in 12cR3, provides a convenient way to search the Enterprise Manager repository for resources. One of its features is to list the resources matching a given SQL query, which is used in the example. The output of this verb is JSON, which can be accessed using out()['data'].
31 - 35	Iterates over the JSON response and prints the target name and target type.

Script Execution

Example 3-8 shows that executing this script retrieves the list of all targets and their types.

Example 3-7   Script that Incorporates Custom SQL with the list() Function

```
1   #emcli_json_processing.py
2   #Import all EM CLI verbs to current program
3   from emcli import *
4   def format(str):
5       '''
6       Given a string argument returns it back or returns
7       a blank string if it is of None type
8       '''
9       if str is None:
10          return ""
11      return str
12
13  def get_targets_with_props(p_prop_name, p_prop_val):
```

```
14 '''
15 Returns targets with given property name and its value. Uses list verb.
16 '''
17 l_sql = "select target_name, target_type, property_value " \
18          "from mgmt$target_properties " \
19          "where property_name = '" + p_prop_name + "' " + "    " \
20          "and property_value like '" + p_prop_val + "'"
21     obj =
list(sql=l_sql)
22     return obj
23 #Set the OMS URL to connect to
24 set_client_property('EMCLI_OMS_URL','https://host1.example.com:1234/em')
25 #Accept all the certificates
26 set_client_property('EMCLI_TRUSTALL','true')
27 #Log in to the OMS
28 login(username='sysman')
29 #Find all the targets that have Version property set to release 12
30 l_targets = get_targets_with_props('Version', '12%')
31 for target in l_targets.out()['data']:
32       tn =  target['TARGET_NAME']
33       tt =  target['TARGET_TYPE']
34       pv =  target['PROPERTY_VALUE']
35       print "Name "+tn + " Type =" + tt + " value=" + pv
```

Example 3-8   Output of Script that Incorporates Custom SQL

```
$emcli @emcli_json_processing.py
Enter password :   ******
Name test.example.com:1838 Type =oracle_emd value=13.1.0.0.0
Logout successful
```

Error Exception Handling

If an exception or error occurs during verb execution, an exception of type emcli.exception.VerbExecutionError is raised. emcli.exception.VerbExecutionError extends from RuntimeError and hence stops the execution. You can use standard Jython exception handling to catch this exception.

emcli.exception.VerbExecutionError has the functions listed in Table 3-4.

Table 3-4   Functions for emcli.exception.VerbExecutionError

Function	Description
error()	Provides the error text of the verb execution.
exit_code()	Provides the exit code of the verb execution.

Example 3-9 shows the usage of VerbExecutionError.

> **Note:**
>
> Line numbers are provided only for explanatory purposes for Table 3-5. For a copy-ready script, see Example A-4 in Sample Scripts.

Script Analysis

Table 3-5 provides an analysis of relevant lines of code.

Table 3-5   Line-by_Line Script Analysis

Lines	Description
4, 6	Imports all EM CLI verbs, and imports VerbExecutionError.
9, 11	Necessary connection to OMS in order to retrieve all targets. Before connecting to the OMS, you must first set the OMS connection details using the set_client_property() function. This sets the OMS URL to https://host1.example.com:1234/em and enables the client to trust all certificates. Note that none of these details are stored in disk. These details are stored in memory and only last for a single script execution. For more information on client properties, enter help('client_properties') from the interactive shell. You can define EMCLI_OMS_URL and EMCLI_TRUSTALL variables as environment variables if you do not want to set these in your script. If you have downloaded certificates somewhere, you can also use the environment variable EMCLI_CERT_LOC to point to the certificate directory. In this case, you do not need EMCLI_TRUSTALL.
14	Login function to connect to the OMS. The example uses the sysman user to log in. This prompts for a password during execution.
22 - 27	Exception use case to create the same group again. This produces a run-time error, which the example is handling in the try except block.

Script Execution

Example 3-10 shows the output of the script shown in Example 3-9.

Example 3-9   Script that Incorporates Exception Handling

```
1   #emcli_error_exception_handling.py
2
3   #import all emcli verbs to current program
4   from emcli import *
5   #import the verbexecutionerror
6   from emcli.exception import VerbExecutionError
7
8   #Set the OMS URL to connect to
9   set_client_property('EMCLI_OMS_URL','https://host1.example.com:1234/em')
10  #Accept all the certificates
11  set_client_property('EMCLI_TRUSTALL','true')
12
13  #Login to the OMS
14  login(username='sysman')
15
16  #Create a group
17  res = create_group(name='Jan_Doe_Group')
18
19  print res.out()
20
21  #Try to create the same group again
22  try:
23    #This will trigger an exception as the group exist already
24    create_group(name='Jan_Doe_Group')
25  except VerbExecutionError , e:
26      print e.error()
27      print 'Exit code:'+str(e.exit_code())
```

Example 3-10   Output of Error Exception Handling Script

```
$emcli @emcli_error_exception_handling.py
Enter password :  ******
Group "Jan_Doe_Group:group" created successfully

Error: Group "Jan_Doe_Group:group" already exists

Exit code:1
Logout successful
```

Utility Functions

The functions shown in Table 3-6 are also available in the EM CLI package.

Table 3-6   Additional Functions

Function	Description
last_out()	Returns the output for the last executed EM CLI command. It returns None if an EM CLI command has not been executed in the current session.
last_error()	Returns the error text (if any) for the last executed EM CLI command. It returns None if an EM CLI command has not been executed in the existing session, or all of the previous executions were successful.
clear()	Clears the current shell in Interactive mode.
exit(ret_val)	Exits from the EM CLI Interactive shell with ret_val.

Extending EM CLI with Python Libraries

You can extend EM CLI with end-user Python libraries by doing one of the following:

- Copy modules to the extension directory, as shown in This example:

  ```
  $EMCLI_INSTALL_HOME/extdir
  ```

- Specify the EMCLI_PYTHONPATH environment variable where the Python modules are loaded from.

Selected Use Cases

Table 3-7 shows various use cases and corresponding solution examples for standard EM CLI versus interactive invocations or scripts.

Table 3-7   Use Case Examples

Task/Action	Usage for Standard EM CLI	Usage for EM CLI Interpreter (interactive shell or script)
Invoke a verb with string-based arguments	`% emcli create_user -name='jane.doe' -type='EXTERNAL_USER'`	`create_user(name='jane.doe', type='EXTERNAL_USER')`

Table 3-7   (Cont.) Use Case Examples

Task/Action	Usage for Standard EM CLI	Usage for EM CLI Interpreter (interactive shell or script)

Invoke a verb with list-based arguments	`% emcli grant_privs -name='jan.doe' -privilege="USE_ANY_BEACON" \ privilege="FULL_TARGET;TARGET_NAME=host1.example.com:TARGET_TYPE=host"`	`#First construct a list`
		`priv_list = ['USE_ANY_BEACON','FULL_TARGET;TARGET_NAME=host1.example.com:TARGET_TYPE=host']`
		
		`#Now use the list`
		`grant_privs(name='jan.doe', privilege=priv_list)`

Invoke a verb with flag-based Boolean arguments	`% emcli get_targets -noheader`	`get_targets(noheader=True)`
Using help	`help $verb_name` For example, help get_targets prints the help for the get_targets verb.	`help('$verb_name')` For example, help('get_targets') prints the help for the get_targets verb.

Advanced Script Examples

This chapter provides examples of using EM CLI to write scripts and automate routine tasks. To use these scripts, Oracle recommends that you are experienced with scripting languages and familiar with Jython (Python for the Java platform).

- Changing Lifecycle Status Properties
- Changing Your Database Password
- Promoting Discovered Targets

Changing Lifecycle Status Properties

To assist you in writing scripts, this section analyzes a sample script that changes the lifecycle status properties.

Example 3-11 enables an Enterprise Manager administrator to change the lifecycle status of all the Oracle databases (release 11.2) in their test environment from Test to Production. Without this script, you would have to sign in to the Enterprise Manager Cloud Control console, and identify all the release 11.2 databases, then manually change the property to Production for each database target from the target's home page.

You can reuse this script whenever there is a request to change a set of targets to a different Lifecycle status.

> **✎ Note:**
>
> Line numbers are provided only for explanatory purposes for Table 3-3. For a copy-ready script, see Example A-5 in Sample Scripts.

Example 3-11   LifeCyclePropertyChange.py

```
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A) THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE OF,
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.
 1 from emcli import *
 2
 3 search_list = ['PROPERTY_NAME=\'DBVersion\'','TARGET_TYPE=
\'oracle_database\'','PROPERTY_VALUE LIKE \'11.2%\'']
 4
 5 if len(sys.argv) == 2:

 6        print login(username=sys.argv[0])
 7        l_prop_val_to_set = sys.argv[1]
 8        l_targets = list(resource="TargetProperties", search=search_list,
columns="TARGET_NAME,TARGET_TYPE,PROPERTY_NAME")
 9        for target in l_targets.out()['data']:
10            t_pn =  'LifeCycle Status'
11            print "INFO: Setting Property name " + t_pn + " to value " +
l_prop_val_to_set
12            print set_target_property_value(property_records=target['TARGET_NAME']
+":"+target['TARGET_TYPE']+":"+t_pn+":"+l_prop_val_to_set)
13  else: ]
14        print "\n ERROR: Property value argument is missing"
15        print "\n INFO: Format to run this file is filename.py <username> <Database
Target LifeCycle Status Property Value>"
```

Script Analysis

Table 3-1 provides an analysis of each line of the code.

Table 3-8   Line-by_Line Script Analysis

Lines	Description
1	Jython import construct to import all EM CLI verb functions in the current program.

Table 3-8   (Cont.) Line-by_Line Script Analysis

Lines	Description
3	search_list is a variable to pass to the search option in the list verb. This example uses escape characters to represent single quotes. To pass more than one value for the same option in the list verb, define as comma separated values, surrounded by square brackets.
5	Defines an if condition to ensure the user provides two arguments with the script, otherwise the script prints an error message (defined in lines #15, 16)
6	Provides a login to Enterprise Manager. You can remove this if you have set up EM CLI with autologin. For more information about setup and autologin, see Downloading and Deploying the EM CLI Client with the Script Option and the setup and the login verbs.
7	l_prop_val_to_set is a variable. This is the property value to be set. Remember that this script is changing this value from Test to Production. You can change this value to any acceptable Lifecycle property value. For a list of valid values, see the modify_lifecycle_stage_name verb or see *Using Administration Groups* in the *Oracle Enterprise Manager Cloud Control Administrator's Guide.*
8	Stores the output of the list verb in l_targets. In the list verb, this script passes the resource as TargetProperties, and search as the search_list variable. This script specifies three columns: • target_name • target_type • property_name
9	Defines a for loop. The data in l_targets is available in JSON format. This loop iterates through the information target property information returned from the list verb. For information about JSON processing, see JSON Processing
10	Sets t_pn to the LifeCycle Status value.
11	Provides a progress message to the user stating that the script is setting the LifeCycle Status to the value passed to the script from the command line.
12	Defines the set_target_property_value verb, which sets the value using the property_records option. When this verb is set for a target pair, it moves to the next one. This example shows three databases, but in reality, use this script for a larger number of databases.
13 -15	Else statement combined with the if condition in line #5. If the arguments specified in line #5 are not provided correctly, then the script displays one of the following error messages: • Property value argument is missing • Format to run this file is filename.py <username> <Database Target LifeCycle Status Property Value>

Script Output

Running Example 3-11 provides the following output:

Example 3-12   Output from LifeCyclePropertyChange.py

```
$ emcli @myScript.py user Production

Login successful

INFO: Setting Property name LifeCycle Status to value Production for db1
Properties updated successfully
INFO: Setting Property name LifeCycle Status to value Production for db2
Properties updated successfully
INFO: Setting Property name LifeCycle Status to value Production for db3
```

```
Properties updated successfully

Logout successful
```

Changing Your Database Password

To assist you in writing scripts, this section analyzes a sample script that changes the password of your database.

Example 3-13 is useful if you want to reset the database password on a regular basis for security compliance. If you change the database password in a target database, then Enterprise Manager monitoring for that target database is unavailable. To ensure consistent monitoring, you would have to sign in to the Enterprise Manager Cloud Control UI, select the required database target and change the password for each and every database, which is very time-consuming.

By using Example 3-13, you can add a number of databases to a group and then change the password for all these databases within the group.

> **Note:**
>
> Line numbers are provided only for explanatory purposes for Table 3-9. For a copy-ready script, see Example A-6 in Sample Scripts.

Example 3-13   dbPasswordChange.py

```
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A) THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE OF,
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.

 1 from emcli  import *
 2 from emcli.exception import VerbExecutionError
 3 import sys
 4 import time
 5
 6 def check_job_status(job):
 7   count=0
 8   while (count < 10):
 9     count = count + 1
10     obj = emcli.get_jobs(job_id=job)
11     #print obj.out()
```

```
12    for entry in obj.out()['data']:
13        l_status = entry['Status ID']
14        l_exec_id = entry['Execution ID']
15        #print entry['Status ID']
16        if (l_status == '5'):
17            print "Job completed successfully"
18            count=100
19        elif (l_status == '4'):
20            l_resp = get_job_execution_detail(execution=l_exec_id,
showOutput=True, xml=True)
21            print "Job failed, error details "
22            print "Output " + str(l_resp.out())
23            count=100
24        else:
25            time.sleep(2)
26
27 def update_db_pwd_for_target(p_target_name, p_target_type, p_old_password,
p_new_password):
28    l_target_name = p_target_name
29    l_target_type = p_target_type
30    print "Changing the password for member : name = " + l_target_name + "
type = " + l_target_type
31    try :
32        l_resp = update_db_password (target_name=l_target_name,
33                                     target_type = l_target_type,
34                                     change_at_target="yes",
35                                     user_name="dbsnmp",
36                                     old_password=p_old_password,
37                                     new_password=p_new_password,
38                                     retype_new_password=p_new_password)
39        l_job_submitted = l_resp.out()['JobId']
40        check_job_status(l_job_submitted)
41    except emcli.exception.VerbExecutionError, e:
42        print "ERROR : Change Password failed for name = " + l_target_name + "
type = " + l_target_type
43        print "ERROR : " + e.error()
44
45 def update_db_pwd_for_group(p_group, p_old_password, p_new_password):
46    print "Changing the password for group - " + p_group + " from " +
p_old_password + " to " + p_new_password
47    members = get_group_members(name=p_group).out()['data']
48    for member in members:
49        l_target_name = member['Target Name']
50        l_target_type = member['Target Type']
51        update_db_pwd_for_target(l_target_name, l_target_type, p_old_password,
p_new_password)
52
53
54 #Set the OMS URL to connect to
56 set_client_property('EMCLI_OMS_URL','https://myoms.com/em')
57 #Accept all the certificates
58 set_client_property('EMCLI_TRUSTALL','true')
59
60 login(username=sys.argv[0])
61
62
63 l_grp_name = 'maurGroup'
64
65 l_group_members =
['db1:oracle_database','db2:oracle_database','db3:rac_database']
66
```

```
67
68
69 res = create_group(name = l_grp_name, add_targets = l_group_members)
70
71 print "Listing members for group " + l_grp_name
72
73 for member in get_group_members(name=l_grp_name).out()['data']:
74    print member
75
76
77 y_n_input = raw_input('Now lets change the password for all the members in this
group(y/n)')
78 if y_n_input != 'y':
79   exit(0)
80
81 l_tgt_username = "dbsnmp"
82 l_old_password = "secret1"
83 l_new_password = "secret2"
84
85 update_db_pwd_for_group(l_grp_name, l_old_password, l_new_password)
```

Script Analysis

Table 3-9 provides an analysis of each line of the code.

Table 3-9   Line-by_Line Script Analysis

Lines	Description
1-2	Imports all EM CLI verbs, and imports VerbExecutionError.
3-4	Imports Jython libraries.
6-25	Defines the job where the script updates the database password for each member of the l_grp_name group. After each successful job completion, the script displays a message to the user, and waits 2 seconds before processing the next job, unless there are any errors or all database passwords are updated.
27-43	Defines variables for updating the database password on each target member of the l_grp_name group. While the script successfully updates the database password, it provides the following message to the user before proceeding to update the password of the next database target: Changing the password for member : name = *database_name* type = *database_type*
45-51	Defines a loop to get all members from the l_grp_name group and update the password for each member as defined in line #85. When the script starts processing this loop, it provides this message to the user: Changing the password for group - *l_grp_name* from *l_old_password* to *l_new_password*
54-58	Necessary connection to OMS to retrieve all targets. Before connecting to the OMS, you must set the OMS connection details using the set_client_property() function. This sets the OMS URL to https://myoms.com/em and enables the client to trust all certificates. Note that none of these details are stored in disk. These details are stored in memory and only last for a single script execution. For more information on client properties, enter help('client_properties') from the interactive shell. You can define EMCLI_OMS_URL and EMCLI_TRUSTALL variables as environment variables if you do not want to set these in your script. If you have downloaded certificates somewhere, you can also use the environment variable EMCLI_CERT_LOC to point to the certificate directory. In this case, you do not need EMCLI_TRUSTALL. For more information, see Interactive Mode — Connecting to an Oracle Management Server (OMS).

Table 3-9   (Cont.) Line-by_Line Script Analysis

Lines	Description
60	Provides a login to the OMS. You can remove this if you have set up EM CLI with autologin. For more information about setup and autologin, see Downloading and Deploying the EM CLI Client with the Script Option and the setup and the login verbs.
63	l_grp_name is a variable for the group name
65	l_group_members is a variable for the array of the database name and type.
69	Adds members from l_group_members to the l_grp_name group
71-74	Provides a message to users while the script is adding members to the l_grp_name group
77-79	Provides a prompt to users to decide if they want to continue with the script. If the user enters n, then the script exits. If the user enters y, then the script proceeds.
81	l_tgt_username is a variable for the user name of the database owner.
82	l_old_password is a variable for the existing password associated with the database owner.
83	l_new_password is a variable for the new password associated with the database owner.
85	Replaces the existing password with the new password for all members of the l_grp_name group.

Script Output

Running Example 3-13 provides the following output:

Example 3-14   Output From dbPasswordChange.py

```
$ emcli @myScript.py user
Enter password : **********

Listing members for group maurGroup
('Target Name': 'aixsdbsi', 'Target Type': 'oracle_database')
('Target Name': 'winsidb1', 'Target Type': 'oracle_database')
('Target Name': 'db10g', 'Target Type': 'rac_database')
('Target Name': 'solcdbone', 'Target Type': 'rac_database')
Now let's change the password for all the members in this group (y/n)y
Changing the password for group - maurGroup from secret1 to secret2
Changing the password for member : name = aixsdbsi type = oracle_database
Job completed successfully
Changing the password for member : name = winsidb1 type = oracle_database
Job completed successfully
Changing the password for member : name = db10g type = rac_database
Job completed successfully
Changing the password for member : name = solcdbone type = rac_database
Job completed successfully

Logout successful
```

Promoting Discovered Targets

To assist you in writing scripts, this section analyzes a sample script that promotes discovered Oracle Database targets.

Consider a corporate environment where databases are added for each user requesting an instance using their UI. Most companies automate the entire process

from database creation, adding data files, and so on. As part of your automation process, at the end, you can add this script.

Example 3-15 promotes the databases automatically, which are then ready to be monitored by Enterprise Manager. This removes the necessity for you to have to log in to the Enterprise Manager Cloud Control UI and promote the databases manually.

> **✎ Note:**
>
> Line numbers are provided only for explanatory purposes for Table 3-10. For a copy-ready script, see Example A-7 in Sample Scripts.

Example 3-15   promote_discovered_dbs.py

```
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A) THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE OF,
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.

 1 from emcli.exception import VerbExecutionError
 2 import sys
 3
 4 alltargets=False
 5 targetparms=0
 6 uname=''
 7 pword=''
 8 url=''
 9 monitor_pw=''
10
11 def helpUsage():
12    print 'Usage: promote_discovered_dbs.py [-help]'
13    print '[-all] Add all discovered Single Instance DBs'
14    print '[-targets <target1:target2:...] Add only targets listed'
15    sys.exit()
16
17 for i in range(len(sys.argv)):
18    if sys.argv[i] in ("-help"):
19            helpUsage()
20    elif sys.argv[i] in ("-targets"):
21            if i+1 < len(sys.argv):
22                    targetparms = sys.argv[i+1]
23    else:
24                    print 'Usage: promote_discovered_dbs.py [-help]'
```

```
25                    print '[-all] Add all discovered Single Instance DBs'
26                    print '[-targets <target1:target2:...] Add only targets
listed'
27                    sys.exit()
28         elif sys.argv[i] in ("-url"):
29             if i+1 < len(sys.argv):
30                    url = sys.argv[i+1]
31         elif sys.argv[i] in ("-username"):
32             if i+1 < len(sys.argv):
33                    uname = sys.argv[i+1]
34     elif sys.argv[i] in ("-password"):
35             if i+1 < len(sys.argv):
36                    pword = sys.argv[i+1]
37     elif sys.argv[i] in ("-monitor_pw"):
38             if i+1 < len(sys.argv):
39                    monitor_pw = sys.argv[i+1]
40     elif sys.argv[i] in ("-all"):
41                    alltargets = True
42
43 # Make sure user did not specify target list and all targets.
44 if alltargets<>0 and targetparms <>0:
45           print 'Cannot specify target list and all switch'
46     print 'Usage: promote_discovered_dbs.py -url <EM URL> -username
<username> -password <password> -monitor_pw <password>'
47     print '[-all] Add all discovered SI Databses'
48     print '[-targets <target1:target2:...] Add only list targets'
49     print '[-help]'
50     sys.exit()
51
52 if len(uname)==0 or len(pword)==0 or len(url)==0:
53         print 'Missing required arguments (-url, -username, -password)'
54         print 'Usage: promote_discovered_dbs.py -url <EM URL> -username
<username> -password <password> -monitor_pw <password>'
55         print '[-all] Add all discovered SI Databses'
56         print '[-targets <target1:target2:...] Add only list targets'
57          print '[-help]'
58         sys.exit()
59
60 # Set Connection properties and logon
61 set_client_property('EMCLI_OMS_URL',url)
62 set_client_property('EMCLI_TRUSTALL','true')
63 login(username=uname,password=pword)
64
65 cred_str = "UserName:dbsnmp;password:" + monitor_pw + ";Role:Normal"
66
67 if targetparms <> 0:
68     targetparms = targetparms.replace(":",":oracle_database;")
+":oracle_database"
69     target_array =
get_targets(unmanaged=True,properties=True,targets=targetparms).out()['data']
70 elif alltargets:
71     target_array =
get_targets(targets="oracle_database",unmanaged=True,properties=True ).out()
['data']
72 else:
73         print 'Missing required arguments (-targets or -all)'
74     helpUsage()
75
76 if len(target_array) > 0:
77     for target in target_array:
78                    print 'Adding target ' + target['Target Name'] + '...',
```

```
79
80          for host in str.split(target['Host Info'],";"):
81              if host.split(":")[0] == "host":]
82                  print host.split(":")[1]
83          try:
84              res1 = add_target(type='oracle_database',name=target['Target
Name'],host=host.split(":")[1], credentials=cred_str,properties=target['Properties'])
85              print 'Succeeded'
86                  except VerbExecutionError, e:
87              print 'Failed'
88              print e.error()
89              print 'Exit code:'+str(e.exit_code())
90 else:
91    print 'INFO: There are no targets to be promoted. Please verify the targets in
Enterprise Manager webpages.'
92
```

Script Analysis

Table 3-10 provides an analysis of each line of the code.

Table 3-10   Line-by_Line Script Analysis

Lines	Description
1	Imports all EM CLI verbs, and imports VerbExecutionError.
2	Imports Jython libraries.
4-9	Sets variables: • uname: User name that allows access to Enterprise Manager • pword: Password associated with the user name • url: Enterprise Manager URL • monitor_pw: Password that allows monitoring of targets
10	Defines input arguments.
11-15	Defines the message displayed to the user if they run the script with invalid or missing arguments: `Usage: promote_discovered_dbs.py [-help][-all] Add all discovered Single Instance DBs[-targets <target1:target2:...] Add only targets listed`
17-41	Defines a For loop that checks that the input variables (defined in lines 4 to 9) are valid and present, otherwise the script terminates and displays the message defined in lines 11-15.
43-50	Defines an If statement to check that the user doesn't provide the `-targets` and the `-all` arguments when running the script. If the user enters both arguments, then the scripts terminates and displays the message defined in lines 11-15.
52-58	Defines an If statement to check that user provides the user name, password, and URL of Enterprise Manager when running the script. If any of these arguments are missing, then the scripts terminates and displays the message defined in lines 11-15.

Table 3-10   (Cont.) Line-by_Line Script Analysis

Lines	Description
60-62	Necessary connection to OMS to retrieve all targets. Before connecting to the OMS, you must set the OMS connection details using the set_client_property() function. This sets the OMS URL to https://myoms.com/em and enables the client to trust all certificates.
	Note that none of these details are stored in disk. These details are stored in memory and only last for a single script execution. For more information on client properties, enter help('client_properties') from the interactive shell.
	You can define EMCLI_OMS_URL and EMCLI_TRUSTALL variables as environment variables if you do not want to set these in your script. If you have downloaded certificates somewhere, you can also use the environment variable EMCLI_CERT_LOC to point to the certificate directory. In this case, you do not need EMCLI_TRUSTALL. For more information, see Interactive Mode — Connecting to an Oracle Management Server (OMS).
63	Provides a login to the OMS. You can remove this if you have set up EM CLI with autologin. For more information about setup and autologin, see Downloading and Deploying the EM CLI Client with the Script Option and the setup and the login verbs.
65	Defines a variable for the credential string required for monitoring targets.
67	Defines an if statement to determine if the -targets argument is provided and if targets exist.
68	Sets the value for target_params
69	Sets the values for target_array, using the targets (where the list of targets is defined by targetparams), unmanaged, and properties parameters of the get_targets verb. When it is set for the first target, the script then moves on to the next target.
70-71	Defines an else if statement to set the values for target_array if the -all option is provided when running the script, using the targets, unmanaged, and properties parameters of the get_targets verb. When it is set for the first target, the script then moves on to the next target.
72-74	Defines an else statement in case the -targets or -all options are not provided when running the script. If this happens, the script terminates and displays the message defined in lines 11-15.
76-78	Determines if there is data in the array, and if there is data, the script displays a message similar to the following: `Adding target abchost.us.example.com... host.us.example.com`
80-82	Extracts the host name from the host information
83-88	Adds the targets to the Management Repository using the add_target verb and displays the following message: Succeeded If the script fails to add targets to the Management Repository, then it displays the following message: Failed
90	From line 76, if there are no targets in the array, the script terminates, and displays the following message to the user: `INFO: There are no targets to be promoted. Please verify the targets in Enterprise Manager webpages.Logout successful`

Script Output

Running Example 3-15 with various options provides the following outputs:

Example 3-16   Output from promote_discovered_dbs.py with -all option

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -username
sysman -password welcome1 -monitor_pw welcome1 -all
Adding target sid7458.us.example.com... host.us.example.com
Succeeded
Logout successful
```

Example 3-17   Output from promote_discovered_dbs.py with -targets option

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -username
sysman -password welcome1 -monitor_pw welcome1 -targets sid7458.us.example.com
Adding target sid7458.us.example.com... host.us.example.com
Succeeded
Adding target db1... host.us.example.com
Succeeded
Adding target db2... host.us.example.com
Succeeded
Adding target db3... host.us.example.com
Succeeded
Logout successful
```

Example 3-18   Output from promote_discovered_dbs.py with -targets option

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -username
sysman -password welcome1 -monitor_pw welcome1 -targets db1:db2:db3
Adding target db1... host.us.example.com
Succeeded
Adding target db2... host.us.example.com
Succeeded
Adding target db3... host.us.example.com
Succeeded
Logout successful
```

**Example 3-19   Output from promote_discovered_dbs.py where the specified targets
do not exist**

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -username
sysman -password welcome1 -monitor_pw welcome1 -targets abc
INFO: There are no targets to be promoted. Please verify the targets in Enterprise
Manager webpages.
Logout successful
```

**Example 3-20   Output from promote_discovered_dbs.py where no targets are
available for promotion**

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -username
sysman -password welcome1 -monitor_pw welcome1 -all
INFO: There are no targets to be promoted. Please verify the targets in Enterprise
Manager webpages.
Logout successful
```

**Example 3-21   Output from promote_discovered_dbs.py where the -all or -targets
option is missing**

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -username
sysman -password welcome1 -monitor_pw welcome1
Missing required arguments (-targets or -all)
Usage: promote_discovered_dbs.py [-help]
[-all] Add all discovered Single Instance DBs
[-targets <target1:target2:...>] Add only targets listed
Logout successful
```

Using the Generic 'List' Verb

EM CLI provides dozens of listing verbs, such as list, get, show, and describe. Rather than selecting from all of these choices, EM CLI provides a generic list verb that you can execute with various types of queries.

The generic list verb provides the following benefits:

- Backed by a RESTful web service

- Generates JavaScript Object Notation (JSON) for script use and standard output for command-line use

- Can specify your own custom SQL to retrieve data from the repository using repository views

Selected list Verb Use Cases

The following sections provide examples of using the list verb for various purposes.

Listing Registered Resources

The list verb supports describing the registered listable resources.

To list all registered resource groups and resources:

```
emcli list -help
```

To describe a specific resource:

```
emcli list -resource="<resource_name>" -help
```

This provides a list of all of the columns along with descriptions.

To list data:

```
emcli list -resource="<resource_name>"
```

To list a specific number of columns:

```
emcli list -resource="<resource_name>" -columns="col1,col2"
```

This command lists only col1 and col2 columns from the specified resource.

Searching for Data

The list verb supports search capabilities.

To search using the list verb:

```
emcli list -resource="<resource_name>" -search="<column>='<value>'"
```

To specify multiple search conditions:

```
emcli list -resource="<resource_name>" -search="<column1> = '<value1>'" -
search="<colum2> = '<value2>'"
```

Registering Resources with the Bind Parameter

To list resources with the bind option:

```
emcli list -resource="<resource_name>" -bind="col1 = "'val1'"
```

This is required for a few resources that require bind parameters as specific input.

Listing with End-user Defined SQL

To execute user-defined SQL using the -sql option:

```
emcli list -sql='select * from mgmt$target'
```

The SQL provided in the -sql option is executed as the Enterprise Manager user MGMT_VIEW, which has read-only access to the Enterprise Manager published MGMT$ database views in the SYSMAN schema. The -sql option requires Super Administrator privileges.

Using the Registered Clients Page

The registered clients page shows all of the EM CLI client installations, and enables you to clean up or delete unused installations.

Note the following characteristics of this page:

- Registered clients are only visible to super users.
- You can search for existing registered clients by host name and version.

Accessing the Page

To access the Registered Clients page and display its contents:

1. From the Enterprise Manager Cloud Control console Setup menu, select Command Line Interface.

   The Command Line Interface tab appears. The figure below shows the page with a single registered client.

Figure 3-1   Registered Client Entries

2. To select the columns to be displayed, select Columns from the View drop-down, then select Manage Columns. Choose any columns to be hidden from the pop-up, then click **OK**.

Column definitions are as follows:

- Client Hostname — Host name of the client where EM CLI is installed.

- Client Version — Version of the EM CLI client.

- Install Home — Complete path of the directory where the EM CLI client is installed.

- Instance Home — Complete path of the directory of the EM CLI instance home.

- Install Type — Can be Standalone, Shell, or Script.

- Setup By — EM CLI was set up by this OS user.

- Auto Login — Enabled or disabled.

- Trust All Certificates — Enabled or disabled.

- Setup at Time — Time at which EM CLI was set up.

- Last Synced At — Time when the sync verb was last executed.

- Last Login At — Time when the last login occurred.

- Last Logged In User — Enterprise Manger user last logged in.

- EM URL — URL of Enterprise Manager.

Deleting an Entry from the Table

Deleting an entry from the table in this page only deletes it from the repository, but does not delete the files from the client host where EM CLI is installed. After deletion, when EM CLI is launched from the same deleted entries directory, it re-adds the entry into the table.

For example, suppose a row in the table indicates that EM CLI is installed in host.example.com at /u01/dir. If you delete this row from the table, this action does not delete the files in /u01/dir at host.example.com. Now if you sign in to host.example.com and execute emcli setup, Enterprise Manger adds the same row into the table which was deleted earlier.

4

Advanced EM CLI Script Examples

This chapter provides examples of using EM CLI to write scripts and automate routine tasks. To use these scripts, Oracle recommends that you are experienced with scripting languages and familiar with Jython (Python for the Java platform).

- Changing Lifecycle Status Properties
- Changing Your Database Password
- Promoting Discovered Targets

Changing Lifecycle Status Properties

To assist you in writing scripts, this section analyzes a sample script that changes the lifecycle status properties.

Example 4-1 enables an Enterprise Manager administrator to change the lifecycle status of all the Oracle databases (release 11.2) in their test environment from Test to Production. Without this script, you would have to sign in to the Enterprise Manager Cloud Control console, and identify all the release 11.2 databases, then manually change the property to Production for each database target from the target's home page.

You can reuse this script whenever there is a request to change a set of targets to a different Lifecycle status.

> **Note:**
>
> Line numbers are provided only for explanatory purposes for Table 4-1. For a copy-ready script, see Example A-5 in Sample Scripts.

Example 4-1   LifeCyclePropertyChange.py

```
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A) THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE OF,
```

```
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.
 1 from emcli import *
 2
 3 search_list = ['PROPERTY_NAME=\'DBVersion\'','TARGET_TYPE=
\'oracle_database\'','PROPERTY_VALUE LIKE \'11.2%\'']
 4
 5 if len(sys.argv) == 2:

 6        print login(username=sys.argv[0])
 7        l_prop_val_to_set = sys.argv[1]
 8        l_targets = list(resource="TargetProperties", search=search_list,
columns="TARGET_NAME,TARGET_TYPE,PROPERTY_NAME")
 9        for target in l_targets.out()['data']:
10            t_pn =  'LifeCycle Status'
11            print "INFO: Setting Property name " + t_pn + " to value " +
l_prop_val_to_set
12            print
set_target_property_value(property_records=target['TARGET_NAME']
+":"+target['TARGET_TYPE']+":"+t_pn+":"+l_prop_val_to_set)
13  else: ]
14        print "\n ERROR: Property value argument is missing"
15        print "\n INFO: Format to run this file is filename.py <username>
<Database Target LifeCycle Status Property Value>"
```

Script Analysis

Table 4-1 provides an analysis of each line of the code.

Table 4-1   Line-by_Line Script Analysis

Lines	Description
1	Jython import construct to import all EM CLI verb functions in the current program.
3	search_list is a variable to pass to the search option in the list verb. This example uses escape characters to represent single quotes. To pass more than one value for the same option in the list verb, define as comma separated values, surrounded by square brackets.
5	Defines an if condition to ensure the user provides two arguments with the script, otherwise the script prints an error message (defined in lines #15, 16)
6	Provides a login to Enterprise Manager. You can remove this if you have set up EM CLI with autologin. For more information about setup and autologin, see Downloading and Deploying the EM CLI Client with the Script Option and the setup and the login verbs.
7	l_prop_val_to_set is a variable. This is the property value to be set. Remember that this script is changing this value from Test to Production. You can change this value to any acceptable Lifecycle property value. For a list of valid values, see the modify_lifecycle_stage_name verb or see olink:EMADM-GUID-E6C3A9BC-AD00-46BA-9CE3-B51338E0AA97 in *Oracle Enterprise Manager Cloud Control Administrator's Guide.*
8	Stores the output of the list verb in l_targets. In the list verb, this script passes the resource as TargetProperties, and search as the search_list variable. This script specifies three columns: • target_name • target_type • property_name

Table 4-1   (Cont.) Line-by_Line Script Analysis

Lines	Description
9	Defines a for loop. The data in l_targets is available in JSON format. This loop iterates through the information target property information returned from the list verb. For information about JSON processing, see JSON Processing
10	Sets t_pn to the LifeCycle Status value.
11	Provides a progress message to the user stating that the script is setting the LifeCycle Status to the value passed to the script from the command line.
12	Defines the set_target_property_value verb, which sets the value using the property_records option. When this verb is set for a target pair, it moves to the next one. This example shows three databases, but in reality, use this script for a larger number of databases.
13 -15	Else statement combined with the if condition in line #5. If the arguments specified in line #5 are not provided correctly, then the script displays one of the following error messages: • Property value argument is missing • Format to run this file is filename.py <username> <Database Target LifeCycle Status Property Value>

Script Output

Running Example 4-1 provides the following output:

Example 4-2   Output from LifeCyclePropertyChange.py

```
$ emcli @myScript.py user Production

Login successful

INFO: Setting Property name LifeCycle Status to value Production for db1
Properties updated successfully
INFO: Setting Property name LifeCycle Status to value Production for db2
Properties updated successfully
INFO: Setting Property name LifeCycle Status to value Production for db3
Properties updated successfully

Logout successful
```

Changing Your Database Password

To assist you in writing scripts, this section analyzes a sample script that changes the password of your database.

Example 4-3 is useful if you want to reset the database password on a regular basis for security compliance. If you change the database password in a target database, then Enterprise Manager monitoring for that target database is unavailable. To ensure consistent monitoring, you would have to sign in to the Enterprise Manager Cloud Control UI, select the required database target and change the password for each and every database, which is very time-consuming.

By using Example 4-3, you can add a number of databases to a group and then change the password for all these databases within the group.

> **Note:**
>
> Line numbers are provided only for explanatory purposes for Table 4-2. For a copy-ready script, see Example A-6 in Sample Scripts.

Example 4-3   dbPasswordChange.py

```
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED
ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS.
ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY,
FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A)
THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON
AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE
SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE
OF,
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.

 1 from emcli  import *
 2 from emcli.exception import VerbExecutionError
 3 import sys
 4 import time
 5
 6 def check_job_status(job):
 7   count=0
 8   while (count < 10):
 9    count = count + 1
10    obj = emcli.get_jobs(job_id=job)
11    #print obj.out()
12    for entry in obj.out()['data']:
13        l_status = entry['Status ID']
14        l_exec_id = entry['Execution ID']
15        #print entry['Status ID']
16        if (l_status == '5'):
17            print "Job completed successfully"
18            count=100
19        elif (l_status == '4'):
20            l_resp = get_job_execution_detail(execution=l_exec_id,
showOutput=True, xml=True)
```

```
21              print "Job failed, error details "
22              print "Output " + str(l_resp.out())
23              count=100
24          else:
25              time.sleep(2)
26
27 def update_db_pwd_for_target(p_target_name, p_target_type, p_old_password,
p_new_password):
28     l_target_name = p_target_name
29     l_target_type = p_target_type
30     print "Changing the password for member : name = " + l_target_name + " type = "
+ l_target_type
31     try :
32         l_resp = update_db_password (target_name=l_target_name,
33                                      target_type = l_target_type,
34                                      change_at_target="yes",
35                                      user_name="dbsnmp",
36                                      old_password=p_old_password,
37                                      new_password=p_new_password,
38                                      retype_new_password=p_new_password)
39         l_job_submitted = l_resp.out()['JobId']
40         check_job_status(l_job_submitted)
41     except emcli.exception.VerbExecutionError, e:
42         print "ERROR : Change Password failed for name = " + l_target_name + " type
= " + l_target_type
43         print "ERROR : " + e.error()
44
45 def update_db_pwd_for_group(p_group, p_old_password, p_new_password):
46     print "Changing the password for group - " + p_group + " from " + p_old_password
+ " to " + p_new_password
47     members = get_group_members(name=p_group).out()['data']
48     for member in members:
49         l_target_name = member['Target Name']
50         l_target_type = member['Target Type']
51         update_db_pwd_for_target(l_target_name, l_target_type, p_old_password,
p_new_password)
52
53
54 #Set the OMS URL to connect to
56 set_client_property('EMCLI_OMS_URL','https://myoms.com/em')
57 #Accept all the certificates
58 set_client_property('EMCLI_TRUSTALL','true')
59
60 login(username=sys.argv[0])
61
62
63 l_grp_name = 'maurGroup'
64
65 l_group_members = ['db1:oracle_database','db2:oracle_database','db3:rac_database']
66
67
68
69 res = create_group(name = l_grp_name, add_targets = l_group_members)
70
71 print "Listing members for group " + l_grp_name
72
73 for member in get_group_members(name=l_grp_name).out()['data']:
74     print member
75
76
77 y_n_input = raw_input('Now lets change the password for all the members in this
```

```
   group(y/n)')
78 if y_n_input != 'y':
79   exit(0)
80
81 l_tgt_username = "dbsnmp"
82 l_old_password = "secret1"
83 l_new_password = "secret2"
84
85 update_db_pwd_for_group(l_grp_name, l_old_password, l_new_password)
```

Script Analysis

Table 4-2 provides an analysis of each line of the code.

Table 4-2   Line-by_Line Script Analysis

Lines	Description
1-2	Imports all EM CLI verbs, and imports VerbExecutionError.
3-4	Imports Jython libraries.
6-25	Defines the job where the script updates the database password for each member of the l_grp_name group. After each successful job completion, the script displays a message to the user, and waits 2 seconds before processing the next job, unless there are any errors or all database passwords are updated.
27-43	Defines variables for updating the database password on each target member of the l_grp_name group. While the script successfully updates the database password, it provides the following message to the user before proceeding to update the password of the next database target: Changing the password for member : name = *database_name* type = *database_type*
45-51	Defines a loop to get all members from the l_grp_name group and update the password for each member as defined in line #85. When the script starts processing this loop, it provides this message to the user: Changing the password for group - *l_grp_name* from *l_old_password* to *l_new_password*
54-58	Necessary connection to OMS to retrieve all targets. Before connecting to the OMS, you must set the OMS connection details using the set_client_property() function. This sets the OMS URL to https://myoms.com/em and enables the client to trust all certificates. Note that none of these details are stored in disk. These details are stored in memory and only last for a single script execution. For more information on client properties, enter help('client_properties') from the interactive shell. You can define EMCLI_OMS_URL and EMCLI_TRUSTALL variables as environment variables if you do not want to set these in your script. If you have downloaded certificates somewhere, you can also use the environment variable EMCLI_CERT_LOC to point to the certificate directory. In this case, you do not need EMCLI_TRUSTALL. For more information, see Interactive Mode — Connecting to an Oracle Management Server (OMS).
60	Provides a login to the OMS. You can remove this if you have set up EM CLI with autologin. For more information about setup and autologin, see Downloading and Deploying the EM CLI Client with the Script Option and the setup and the login verbs.
63	l_grp_name is a variable for the group name
65	l_group_members is a variable for the array of the database name and type.
69	Adds members from l_group_members to the l_grp_name group

Table 4-2   (Cont.) Line-by_Line Script Analysis

Lines	Description
71-74	Provides a message to users while the script is adding members to the l_grp_name group
77-79	Provides a prompt to users to decide if they want to continue with the script. If the user enters n, then the script exits. If the user enters y, then the script proceeds.
81	l_tgt_username is a variable for the user name of the database owner.
82	l_old_password is a variable for the existing password associated with the database owner.
83	l_new_password is a variable for the new password associated with the database owner.
85	Replaces the existing password with the new password for all members of the l_grp_name group.

Script Output

Running Example 4-3 provides the following output:

Example 4-4   Output From dbPasswordChange.py

```
$ emcli @myScript.py user
Enter password : **********

Listing members for group maurGroup
('Target Name': 'aixsdbsi', 'Target Type': 'oracle_database')
('Target Name': 'winsidb1', 'Target Type': 'oracle_database')
('Target Name': 'db10g', 'Target Type': 'rac_database')
('Target Name': 'solcdbone', 'Target Type': 'rac_database')
Now let's change the password for all the members in this group (y/n)y
Changing the password for group - maurGroup from secret1 to secret2
Changing the password for member : name = aixsdbsi type = oracle_database
Job completed successfully
Changing the password for member : name = winsidb1 type = oracle_database
Job completed successfully
Changing the password for member : name = db10g type = rac_database
Job completed successfully
Changing the password for member : name = solcdbone type = rac_database
Job completed successfully

Logout successful
```

Promoting Discovered Targets

To assist you in writing scripts, this section analyzes a sample script that promotes discovered Oracle Database targets.

Consider a corporate environment where databases are added for each user requesting an instance using their UI. Most companies automate the entire process from database creation, adding data files, and so on. As part of your automation process, at the end, you can add this script.

Example 4-5 promotes the databases automatically, which are then ready to be monitored by Enterprise Manager. This removes the necessity for you to have to log in to the Enterprise Manager Cloud Control UI and promote the databases manually.

> **✏ Note:**
>
> Line numbers are provided only for explanatory purposes for Table 4-3. For a copy-ready script, see Example A-7 in Sample Scripts.

Example 4-5   promote_discovered_dbs.py

```
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED
ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS.
ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY,
FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A)
THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON
AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE
SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE
OF,
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.

 1 from emcli.exception import VerbExecutionError
 2 import sys
 3
 4 alltargets=False
 5 targetparms=0
 6 uname=''
 7 pword=''
 8 url=''
 9 monitor_pw=''
10
11 def helpUsage():
12     print 'Usage: promote_discovered_dbs.py [-help]'
13     print '[-all] Add all discovered Single Instance DBs'
14     print '[-targets <target1:target2:...] Add only targets listed'
15     sys.exit()
16
17 for i in range(len(sys.argv)):
18     if sys.argv[i] in ("-help"):
19             helpUsage()
20     elif sys.argv[i] in ("-targets"):
```

```
21              if i+1 < len(sys.argv):
22                  targetparms = sys.argv[i+1]
23      else:
24                  print 'Usage: promote_discovered_dbs.py [-help]'
25                  print '[-all] Add all discovered Single Instance DBs'
26                  print '[-targets <target1:target2:...] Add only targets listed'
27                  sys.exit()
28          elif sys.argv[i] in ("-url"):
29              if i+1 < len(sys.argv):
30                  url = sys.argv[i+1]
31          elif sys.argv[i] in ("-username"):
32              if i+1 < len(sys.argv):
33                  uname = sys.argv[i+1]
34      elif sys.argv[i] in ("-password"):
35              if i+1 < len(sys.argv):
36                  pword = sys.argv[i+1]
37      elif sys.argv[i] in ("-monitor_pw"):
38              if i+1 < len(sys.argv):
39                  monitor_pw = sys.argv[i+1]
40      elif sys.argv[i] in ("-all"):
41                  alltargets = True
42
43  # Make sure user did not specify target list and all targets.
44  if alltargets<>0 and targetparms <>0:
45          print 'Cannot specify target list and all switch'
46      print 'Usage: promote_discovered_dbs.py -url <EM URL> -username <username> -
password <password> -monitor_pw <password>'
47      print '[-all] Add all discovered SI Databses'
48      print '[-targets <target1:target2:...] Add only list targets'
49      print '[-help]'
50      sys.exit()
51
52  if len(uname)==0 or len(pword)==0 or len(url)==0:
53          print 'Missing required arguments (-url, -username, -password)'
54          print 'Usage: promote_discovered_dbs.py -url <EM URL> -username <username>
-password <password> -monitor_pw <password>'
55          print '[-all] Add all discovered SI Databses'
56          print '[-targets <target1:target2:...] Add only list targets'
57          print '[-help]'
58          sys.exit()
59
60  # Set Connection properties and logon
61  set_client_property('EMCLI_OMS_URL',url)
62  set_client_property('EMCLI_TRUSTALL','true')
63  login(username=uname,password=pword)
64
65  cred_str = "UserName:dbsnmp;password:" + monitor_pw + ";Role:Normal"
66
67  if targetparms <> 0:
68      targetparms = targetparms.replace(":",":oracle_database;")+":oracle_database"
69      target_array =
get_targets(unmanaged=True,properties=True,targets=targetparms).out()['data']
70  elif alltargets:
71      target_array =
get_targets(targets="oracle_database",unmanaged=True,properties=True ).out()['data']
72  else:
73          print 'Missing required arguments (-targets or -all)'
74      helpUsage()
75
76  if len(target_array) > 0:
77      for target in target_array:
```

```
78                  print 'Adding target ' + target['Target Name'] + '...',
79
80          for host in str.split(target['Host Info'],";"):
81              if host.split(":")[0] == "host":]
82                  print host.split(":")[1]
83          try:
84              res1 = add_target(type='oracle_database',name=target['Target
Name'],host=host.split(":")[1],
credentials=cred_str,properties=target['Properties'])
85              print 'Succeeded'
86                  except VerbExecutionError, e:
87              print 'Failed'
88              print e.error()
89              print 'Exit code:'+str(e.exit_code())
90 else:
91    print 'INFO: There are no targets to be promoted. Please verify the
targets in Enterprise Manager webpages.'
92
```

Script Analysis

Table 4-3 provides an analysis of each line of the code.

Table 4-3   Line-by_Line Script Analysis

Lines	Description
1	Imports all EM CLI verbs, and imports VerbExecutionError.
2	Imports Jython libraries.
4-9	Sets variables: • uname: User name that allows access to Enterprise Manager • pword: Password associated with the user name • url: Enterprise Manager URL • monitor_pw: Password that allows monitoring of targets
10	Defines input arguments.
11-15	Defines the message displayed to the user if they run the script with invalid or missing arguments: `Usage: promote_discovered_dbs.py [-help][-all] Add all discovered Single Instance DBs[-targets <target1:target2:...] Add only targets listed`
17-41	Defines a For loop that checks that the input variables (defined in lines 4 to 9) are valid and present, otherwise the script terminates and displays the message defined in lines 11-15.
43-50	Defines an If statement to check that the user doesn't provide the -targets and the -all arguments when running the script. If the user enters both arguments, then the scripts terminates and displays the message defined in lines 11-15.
52-58	Defines an If statement to check that user provides the user name, password, and URL of Enterprise Manager when running the script. If any of these arguments are missing, then the scripts terminates and displays the message defined in lines 11-15.

Table 4-3   (Cont.) Line-by_Line Script Analysis

Lines	Description
60-62	Necessary connection to OMS to retrieve all targets. Before connecting to the OMS, you must set the OMS connection details using the set_client_property() function. This sets the OMS URL to https://myoms.com/em and enables the client to trust all certificates.
	Note that none of these details are stored in disk. These details are stored in memory and only last for a single script execution. For more information on client properties, enter help('client_properties') from the interactive shell.
	You can define EMCLI_OMS_URL and EMCLI_TRUSTALL variables as environment variables if you do not want to set these in your script. If you have downloaded certificates somewhere, you can also use the environment variable EMCLI_CERT_LOC to point to the certificate directory. In this case, you do not need EMCLI_TRUSTALL. For more information, see Interactive Mode — Connecting to an Oracle Management Server (OMS).
63	Provides a login to the OMS. You can remove this if you have set up EM CLI with autologin. For more information about setup and autologin, see Downloading and Deploying the EM CLI Client with the Script Option and the setup and the login verbs.
65	Defines a variable for the credential string required for monitoring targets.
67	Defines an if statement to determine if the `-targets` argument is provided and if targets exist.
68	Sets the value for target_params
69	Sets the values for target_array, using the targets (where the list of targets is defined by targetparams), unmanaged, and properties parameters of the get_targets verb. When it is set for the first target, the script then moves on to the next target.
70-71	Defines an else if statement to set the values for target_array if the -all option is provided when running the script, using the targets, unmanaged, and properties parameters of the get_targets verb. When it is set for the first target, the script then moves on to the next target.
72-74	Defines an else statement in case the -targets or -all options are not provided when running the script. If this happens, the script terminates and displays the message defined in lines 11-15.
76-78	Determines if there is data in the array, and if there is data, the script displays a message similar to the following:
	`Adding target abchost.us.example.com... host.us.example.com`
80-82	Extracts the host name from the host information
83-88	Adds the targets to the Management Repository using the add_target verb and displays the following message:
	Succeeded
	If the script fails to add targets to the Management Repository, then it displays the following message:
	Failed
90	From line 76, if there are no targets in the array, the script terminates, and displays the following message to the user:
	`INFO: There are no targets to be promoted. Please verify the targets in Enterprise Manager webpages.Logout successful`

Script Output

Running Example 4-5 with various options provides the following outputs:

Example 4-6   Output from promote_discovered_dbs.py with -all option

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -
username sysman -password welcome1 -monitor_pw welcome1 -all
Adding target sid7458.us.example.com... host.us.example.com
Succeeded
Logout successful
```

Example 4-7   Output from promote_discovered_dbs.py with -targets option

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -
username sysman -password welcome1 -monitor_pw welcome1 -targets
sid7458.us.example.com
Adding target sid7458.us.example.com... host.us.example.com
Succeeded
Adding target db1... host.us.example.com
Succeeded
Adding target db2... host.us.example.com
Succeeded
Adding target db3... host.us.example.com
Succeeded
Logout successful
```

Example 4-8   Output from promote_discovered_dbs.py with -targets option

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -
username sysman -password welcome1 -monitor_pw welcome1 -targets db1:db2:db3
Adding target db1... host.us.example.com
Succeeded
Adding target db2... host.us.example.com
Succeeded
Adding target db3... host.us.example.com
Succeeded
Logout successful
```

**Example 4-9   Output from promote_discovered_dbs.py where the specified
targets do not exist**

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -
username sysman -password welcome1 -monitor_pw welcome1 -targets abc
INFO: There are no targets to be promoted. Please verify the targets in
Enterprise Manager webpages.
Logout successful
```

**Example 4-10   Output from promote_discovered_dbs.py where no targets are
available for promotion**

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -
username sysman -password welcome1 -monitor_pw welcome1 -all
INFO: There are no targets to be promoted. Please verify the targets in
Enterprise Manager webpages.
Logout successful
```

Example 4-11   Output from promote_discovered_dbs.py where the -all or -targets option is missing

```
$ emcli @promote_discovered_dbs.py -url https://host.us.example.com:7799/em -username
sysman -password welcome1 -monitor_pw welcome1
Missing required arguments (-targets or -all)
Usage: promote_discovered_dbs.py [-help]
[-all] Add all discovered Single Instance DBs
[-targets <target1:target2:...] Add only targets listed
Logout successful
```

5

Verb Reference

This chapter provides a complete listing of all EM CLI verbs in categorical as well as alphabetical order. Each verb provides complete syntax and usage information.

Verb Categories

This section lists all of the verbs for this release in the following categories:

- Basic Operational Verbs
- Account Management Verbs - Oracle Database
- Add Host Verbs
- Application Data Model Verbs
- Agent Administration Verbs
- Agent Recovery Verbs
- Agent Upgrade Verbs
- Application Data Models Verbs
- Audit Settings Verbs
- AWR Warehouse Verbs
- Bare Metal Provisioning Verbs
- BI Publisher Reports Verbs
- Big Data Appliance
- Blackout Verbs
- CFW Verbs
- Chargeback Verbs
- Clean Coherence Cluster Verbs
- Cloning Verbs
- Compliance Verbs
- Configuration Association History
- Configuration Data
- Configuration Compare
- Connector Verbs
- Cost Center Management Verbs
- Credential Verbs
- Credential Verbs - Oracle Database
- Custom Plug-in Update Verbs

- Data Guard Verbs
- Database High Availibility Verbs
- Database Lifecycle Management Verbs
- Database Machine Targets Customer Support Identifier (CSI) Assignment Verbs
- Database Profile Job Verbs
- Database Replay Verbs
- Data Subset Verbs
- DBaaS Verbs
- Deployment Procedure Verbs
- Diagchecks Verbs
- Diagnostic Snapshots Verbs
- Discover and Push to Agents Verbs
- Discovery Prechecks Verbs
- Auto Service Request (ASR) Verbs
- Event and Incident Verbs
- Execute Command Verbs
- Fusion Middleware Provisions Verbs
- Fusion Middleware Diagnostic Advisor Verbs
- Generic Middleware Provisioning Verbs
- Gold Agent Image Verbs
- Group Verbs
- Incident Rules Verbs
- Installation Verbs
- Internal Metrics Verbs
- Java EE Application Component Verbs
- JBoss Target Management Group
- Job Verbs
- Latest Configurations Verbs
- Licensing Verbs
- Log Management Verbs
- Masking Verbs
- Metric Collection and Alerts Verbs
- Metric Data Loading Verbs
- Metric Verbs
- Monitoring Template Verbs
- Notification Verbs
- OMS Configuration Properties

5-3

- OMS CPU Activity Report Verbs
- OMS Plug-in Deployment Verbs
- Oracle Database as Service (DBaaS) Verbs
- Package Fusion Application Problem Verbs
- Patch Verbs
- Ping Subsytem Verbs
- Platform as a Service (PaaS) Verbs
- Pluggable Database Job Verbs
- Prerequisite Check Verbs
- Privilege Delegation Settings Verbs
- Provisioning Verbs
- Reconfig Job Verbs
- Redundancy Group Verbs
- Refresh Coherence Verbs
- Refresh WLS Domain Verbs
- Report Import/Export Verbs
- Resource Verbs
- Saved Configurations Verbs
- Secure Communication Verbs
- Self Update Verbs
- Services Verbs
- Server-generated Alert Metric Verbs
- Siebel Verbs
- SiteGuard Verbs
- Software Library Verbs
- Software Maintenance Verbs
- SSA Verbs
- Storage Management Framework Verbs
- Switch EM Monitoring Agent for Cluster Target Verbs
- System Verbs
- Target Data Verbs
- Tenant Administrative Verbs
- Toaster Verbs
- Trace Verbs
- Transparent Data Encryption
- User-defined Metrics (UDM) Migration Verbs
- Upgrade Database Job Verbs

Basic Operational Verbs

Note: Only these verbs are available immediately after installation.

Account Management Verbs - Oracle Database

Add Host Verbs

Application Data Model Verbs

Agent Administration Verbs

Agent Recovery Verbs

Agent Upgrade Verbs

Application Data Models Verbs

Audit Settings Verbs

AWR Warehouse Verbs

Bare Metal Provisioning Verbs

BI Publisher Reports Verbs

Big Data Appliance

delete_bda_cluster

discover_bda_cluster

discover_cloudera_cluster

relocate_bda_target

show_bda_clusters

Blackout Verbs

add_blackout_reason

create_blackout

delete_blackout

get_blackout_details

get_blackout_reasons

get_blackout_targets

get_blackouts

stop_blackout

CFW Verbs

cancel_cloud_service_requests

delete_cloud_service_instances

delete_cloud_user_objects

get_cloud_service_instances

get_cloud_service_requests

get_cloud_user_objects

Chargeback Verbs

add_chargeback_entity

assign_charge_plan

assign_cost_center

create_charge_entity_type

create_charge_item

delete_charge_item

export_charge_plans

export_custom_charge_items

get_metering_data

import_charge_plans

import_custom_charge_items

list_chargeback_entities

list_chargeback_entity_types

list_charge_item_candidates

list_charge_plans

list_cost_centers

remove_chargeback_entity

unassign_charge_plan

unassign_cost_center

Clean Coherence Cluster Verbs

clean_down_members

Cloning Verbs

Compliance Verbs

Configuration Association History

Configuration Data

Configuration Compare

Connector Verbs

Cost Center Management Verbs

Credential Verbs

Credential Verbs - Oracle Database

Custom Plug-in Update Verbs

Data Guard Verbs

Database High Availibility Verbs

Database Lifecycle Management Verbs

Database Machine Targets Customer Support Identifier (CSI) Assignment Verbs

Database Profile Job Verbs

Database Replay Verbs

Data Subset Verbs

generate_subset
import_subset_definition
import_subset_dump
import_subset_dump
list_subset_definitions

DBaaS Verbs

See *DBaaS REST APIs and EM CLI Verbs Based Use Cases* in the *Enterprise Manager Cloud Administration Guide.*

Deployment Procedure Verbs

confirm_instance
delete_instance
delete_procedure
describe_procedure_input
get_executions
get_instance_data
get_instance_status
get_instances
get_procedure_types
get_procedure_xml
get_procedures
get_retry_arguments
get_runtime_data
ignore_instance
reschedule_instance
resume_instance
retry_instance
save_procedure_input
stop_instance
submit_procedure
suspend_instance
update_and_retry_step
update_procedure_input

Diagchecks Verbs

apply_diagcheck_exclude
define_diagcheck_exclude
diagchecks_deploy_status
diagchecks_deploy_tglist
list_diagcheck_exclude_applies
list_diagcheck_exclusions
list_diagchecks
undeploy_diagchecks
update_diagchecks

Diagnostic Snapshots Verbs

create_diag_snapshot
delete_diag_snapshot

ORACLE®

Discover and Push to Agents Verbs

delete_siebel

discover_coherence

discover_db

discover_fa

discover_fa

discover_gf

discover_siebel

discover_wls

generate_discovery_input

refresh_fa

run_fa_diagnostics

Discovery Prechecks Verbs

fmw_discovery_prechecks

Auto Service Request (ASR) Verbs

em_asr_asset_actcred

em_asr_asset_activation_details

em_asr_asset_activation_job

em_asr_asset_exclude_list

em_asr_asset_include_list

em_asr_deregister

em_asr_register

em_asr_xsl_upload

Event and Incident Verbs

clear_problem

create_resolution_state

delete_resolution_state

get_resolution_states

modify_incident_rule

modify_resolution_state

publish_event

Execute Command Verbs

execute_hostcmd

execute_sql

Fusion Middleware Diagnostic Advisor Verbs

activate_mda_finding_types

deactivate_mda_finding_types

disable_mda_finding_types_for_targets

enable_mda_finding_types_for_targets

get_mda_engine_status

list_mda_finding_types

list_mda_properties

run_mda_health_check

start_mda_engine

Installation Verbs

Internal Metrics Verbs

Java EE Application Component Verbs

JBoss Target Management Group

Job Verbs

Latest Configurations Verbs

Licensing Verbs

Log Management Verbs

Masking Verbs

Metric Collection and Alerts Verbs

Metric Data Loading Verbs

Metric Verbs

Monitoring Template Verbs

Notification Verbs

OMS Configuration Properties

OMS CPU Activity Report Verbs

OMS Plug-in Deployment Verbs

Oracle Database as Service (DBaaS) Verbs

Package Fusion Application Problem Verbs

Patch Verbs

Ping Subsytem Verbs

Platform as a Service (PaaS) Verbs

Pluggable Database Job Verbs

Prerequisite Check Verbs

Privilege Delegation Settings Verbs

Services Verbs

Server-generated Alert Metric Verbs

Siebel Verbs

SiteGuard Verbs

Software Library Verbs

ORACLE®

Software Maintenance Verbs

SSA Verbs

Storage Management Framework Verbs

disable_snapclone
edit_storage_ceiling
enable_snapclone
manage_storage_access
register_storage
remove_storage
synchronize_storage

Switch EM Monitoring Agent for Cluster Target Verbs

switch_cluster_agent

System Verbs

create_system
delete_system
get_system_members
modify_system

Target Data Verbs

add_proxy
add_target
add_target_property
add_to_target_properties_master_list
change_target_owner
create_assoc
delete_assoc
delete_from_target_properties_master_list
delete_proxy
delete_target
disable_target
enable_target
get_accesspoints
get_best_implementer
get_target_properties
get_target_types
get_targets
list_allowed_pairs
list_assoc
list_target_properties_master_list
list_proxies
list_target_property_names
list_targets_having_property_value
migrate_to_lifecycle_status
modify_lifecycle_stage_name
modify_proxy
modify_target
relocate_targets
remove_target_property
rename_target

Tenant Administrative Verbs

Toaster Verbs

Trace Verbs

Transparent Data Encryption

User-defined Metrics (UDM) Migration Verbs

Upgrade Database Job Verbs

User Administration Verbs

-input_file Syntax Guidelines

-input_file Syntax

This option enables you to provide an argument to be specified in a file. For example:

```
emcli xyzverb -input_file="arg1:file1.txt" -input_file="arg2:file2.txt"
```

This string literally translates to:

```
emcli xyzverb -arg1=<contents of file1.txt> -arg2=<contents of file2.txt>

emcli xyzverb -input_file="name:/tmp/b1.txt"  -input_file="type:/tmp/b2.txt"
-input_file="bcnName:/tmp/b3.txt"
```

This example makes User1 an Enterprise Manager user, which is already created on an external user store like the SSO server. The contents of priv_file are view_target;host1.example.com:host. User1 will have view privileges on the host1.example.com:host target.

```
emcli create_user
      -name="User1"
      -type="EXTERNAL_USER"
      -input_file="privilege:/home/user1/priv_file"
```

-input_file for Jobs

For most job verbs, you can specify all of the needed properties in a property file. You can also provide a few properties on the command line. Properties set on the command line override values set in the file.

The property file consists of name=value pairs. For example, put the following into myFile.txt:

```
name=MY JOB 1
    type=OSCommand
    description=this is a test job
    target_list=target1:host
    variable.default_shell_command=ls -l
    schedule.frequency=IMMEDIATE
```

... then run:

```
emcli create_job -input_file=property_file:myFile.txt
```

This creates an OS Command job called "MY JOB 1" using preferred credentials.

Usage of Properties

For the create verbs, all properties set in the file are used. For verbs that act on multiple jobs, like suspend and resume, only "search" properties are used (name, type, targets. and scheduled starting and ending times).

Creating a Property File

The best way to create a property file is to start by describing a job similar to the one you want to create, and/or by describing a job type. This provides a list of which properties are needed by a given job type.

Determining Variables for a Job

Most properties are the same from one job to another. For example:

name, type, description, kind, targetType, cred, schedule notification

The variables needed for a job type change from job to job. Describe a job type to find out which variables it requires.

For example, the following command creates a property file template based on job MYJOB1. This lists the properties set by this job.

```
emcli describe_job [-verbose] -name=MyJob1 > myPropFile.txt
```

This example creates a property file template for an OS Command job. This lists the properties allowed by this job type, including all required and optional variables. Variables marked as deprecated should be avoided.

```
emcli describe_job_type [-verbose] -type=OSCommand > myPropFile.txt
```

Overriding the Separator and Subseparator

Not all verbs allow separator and subseparator to be overridden. The semi-colon (;) and colon (:) are respectively the default separator and subseparator. The separator

is used for arguments that take multiple values, and subseparator is used when the value itself has multiple values. You can override either one of them or both.

The syntax is:

```
separator=<option_for_which_separator_has_to_be_applied>="separator_value"
```

As an example of using the separator and subseperator to create a group containing database2 and database3, the command could be:

```
emcli create_group -name="tstgrp" -add_targets="database2:oracle_database;
database3:oracle_database"
```

Using this command as the basis for modification, these examples show overrides of separator and/or subseperator:

```
emcli create_group -name="tstgrp1" -add_targets="database2:oracle_database,
database3:oracle_database" -separator=add_targets=","
```

```
emcli create_group -name="tstgrp2" -add_targets="database2&oracle_database,
database3&oracle_database" -separator=add_targets="," -subseparator=add_targets="&"
```

```
emcli create_group -name="tstgrp3" -add_targets="database2&oracle_database;
database3&oracle_database" -subseparator=add_targets="&"
```

EM CLI Verbs

The following sections provide descriptions, formats, and options for all EM CLI verbs. Some of the verbs also contain one or more examples.

abort_udmmig_session

Aborts the migration of user-defined metrics (UDMs) to metric extensions in a session.

Format

```
emcli abort_udmmig_session
        -session_id=<sessionId>
        [-input_file=specific_tasks:<complete_path_to_file>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **session_id**

  Specify the ID that was returned when the session was created, or from the output of udmmig_summary.

- **input_file**

  Points at a file name that contains a target UDM, one per line in the following format:

  ```
  <targetType>,<targetName>,<collection name>
  ```

  Use targetType=Template to indicate a template. Use * for the collection name to abort all UDMs for a target. The input file should be in UTF-8 format.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

Example

This example aborts the specified migration session. The UDM is returned to the unconverted list.

```
emcli abort_udmmig_session -session_id=<sessionId>
```

activate_mda_finding_types

Activates the specified MDA finding types. A finding type can have status 'N' (new), 'A' (active), or 'I' (inactive). Only those finding types that are currently inactive, for example, status 'I', will be activated. For others the previous status is retained.

Format

```
emcli activate_mda_finding_types       [-finding_types="<list of finding
types>"]       [-separator=finding_types="separator_for_finding_types_values"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   finding_types

    List of finding types. The default separator to be used is ';'.

*   separator=finding_types

    Indicates the custom separator used for the list of finding types. This option is mandatory if any other character apart from ';' is used as a separator in the finding types list.

Example

The following example activates multiple finding types with a custom separator:

```
emcli activate_mda_finding_types
      -
finding_types="oracle.sysman.emas.wls_gc_overhead$oracle.sysman.emas.wls_heap_con
fig"
      -separator=finding_types="$"
```

add_beacon

Adds a beacon to the monitoring set of beacons. All enabled tests are pushed to the beacon.

Format

```
emcli add_beacon
      -name=target_name
      -type=target_type
      -bcnName=beacon_name
      [-dontSetKey]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Service target name.

- **type**

  Service target type.

- **bcnName**

  Beacon name to add.

- **dontSetKey**

  Indicates the added beacon is not automatically a key beacon. Only use this option if you do not want the beacon to participate in the availability calculation of the service and tests.

Example

This example adds MyBeacon as a key beacon to the MyTarget service target of type `generic_service`.

```
emcli add_beacon -name='MyTarget' -type='generic_service'
     -bcnName='MyBeacon'
```

add_blackout_reason

Adds a new blackout reason. Only Super Administrators can perform this action.

Format

```
emcli add_blackout_reason -name="<blackout reason>"
```

Example

This example adds the blackout reason "Testing Purposes."

```
emcli add_blackout_reason -name="Testing Purposes"
```

add_chargeback_entity

Adds the given entity to Chargeback.

Format

```
add_chargeback_entity
     -entity_name="eName"
     -entity_type="eType"
     -usage_mode="uMode"
```

Options

- **entity_name**

  Name of the entity to be added to Chargeback.

- **entity_type**

  Type of entity to be added to Chargeback.

- **usage_mode**

  Usage mode by which it should be added to Chargeback. You can see the usage modes for a particular entity type by entering list_chargeback_entity_types -entity_type.

See Also

assign_charge_plan
assign_cost_center
list_chargeback_entities
list_chargeback_entity_types
list_charge_plans
list_cost_centers
remove_chargeback_entity
unassign_charge_plan
unassign_cost_center

add_chef_cookbook

Adds a software library component and directives for a chef cookbook. If more than one software library storage location is configured, the default storage location is used. Use this verb once for each cookbook.

Format

```
emcli add_chef_cookbook
      -name="component_name"
      -folder_name="swlib_folder_name"
      -filename="filename"
      -recipe_names="recipe_name11;recipe_name2;..."
```

Options

- name

  The software library component name

- recipe_names

  Cookbook recipe names. The recipe names are semi-colon separated. A directive will be created for each recipe.

- folder_name

  The software library folder where the component and directives will be saved.

- filename

  Name of the cookbook file. It must be in tar, gzip format. The cookbook name must be the "base" of the filename. For example, if the cookbook is named 'cookie', then the cookbook filename must be cookie.tar.gz.

Example

The following example creates a custom software library component and directives for a new chef cookbook. Entities are created in a sub folder of the MyComponents folder.

```
emcli add_chef_cookbook
      -name=chef_component
```

```
                    -recipe_names="start;stop;install"
                    -folder_name="MyComponents folder"
                    -filename="mysql.tar.gz"
```

add_cloud_db_target

You can use `add_cloud_db_target` to discover an Autonomous Database in Oracle Enterprise Manager.

Format

```
emcli add_cloud_db_target
              -name="target_name"
              -type="target_type"
              -host="agent_host"
              -zip_file_location="cred_file_zip_location"
              -
credentials="UserName:<db_username>;password:<db_password>;Role:<db_user_role>"
              -wallet_password="wallet_password"
              -service_name="tns_service_name"
              [-is_dedicated="is_dedicated"]
              [-standby_agent_host="standby_agent_host"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **-name**

  Name of the Autonomous Database target.

- **-type**

  Type of Autonomous Database. The target type value for **Autonomous Data Warehouse** databases `oracle_cloud_adw` and the value for **Autonomous Transaction Processing** databases is `oracle_cloud_atp`.

- **-host**

  Host on which the agent that you are using to discover the Autonomous Database is installed and running.

- **-zip_file_location**

  Location of the Oracle Cloud Infrastructure Client Credentials (Wallet) `.zip` file. The `.zip` file location should be accessible from the OMS host.

- **-credentials**

  Monitoring user credentials. It is recommended that you use the adbsnmp user account. See About User Accounts.

- **-wallet_password**

  The wallet password set in the Oracle Cloud Infrastructure Console when downloading the OCI Client Credential Wallet.

- **-service_name**

  Predefined database service name of the Autonomous Database. The `low` database service is the default value and to perform monitoring and management tasks, it is recommended that you use the `low` database service.

For information on Database Service Names, see the Predefined Database Service Names section in the documentation for the Autonomous Database you are discovering.

> **Note:**
>
> Oracle Enterprise Manager supports both TCP and TCPS using the TLS protocol for Autonomous Databases – Dedicated, and only TCPS using the TLS protocol for Autonomous Databases – Shared. The default protocol for Autonomous Databases – Dedicated is TCP.

- **-is_dedicated**

  `True` for an Autonomous Database – Dedicated and `False` for an Autonomous Database – Shared. If a value is not specified for this parameter, then it defaults to an Autonomous Database – Shared.

- **-standby_agent_host**

  Host on which a backup agent is installed. If the primary agent goes down or crashes, then the backup agent monitors the target. This is an optional parameter.

Example

This example adds a new cloud db target of type `oracle_cloud_adw`.

```
emcli add_cloud_db_target
            -name="ATPD1"
            -type="oracle_cloud_atp"
            -host="myhostname.example.com"
            -zip_file_location="/u01/oracle/atpd/wallet_ATPD1.zip"
            -credentials="UserName:adbsnmp;password:password;Role:Normal"
            -wallet_password="password"
            -service_name="ATPD1_low"
            -is_dedicated="True"
            -standby_agent_host="standbyhostname.example.com"
```

add_comment_to_event

Adds a comment to a specified event.

Format

```
emcli add_comment_to_event
    -event_id="event ID"
     -comment="text"
```

Options

- event_id

  ID of the event to which the comment is to be added.

- comment

  Comment text.

Example

The following command adds the comment "Working on this" to an event with the ID 2.

```
emcli -add_comment_to_event -event_id="2" -comment="Working on this"
```

add_comment_to_problem

Adds a comment to a specified problem.

Format

```
emcli add_comment_to_problem
      -problem_id="Problem ID"
      -comment="text"
```

Options

*   problem_id

    ID of the problem to which a comment is to be added.

*   comment

    Comment verbiage.

Example

The following command adds the comment "Working on this" to a problem with the ID 2.

```
emcli -add_comment_to_problem -problem_id="2" -comment="Working on this"
```

add_comment_to_incident

Adds a comment to a specified incident.

Format

```
emcli add_comment_to_incident
    -incident_id="Incident ID"
    -comment="text"
```

Options

*   incident_id

    ID of the incident to which a comment is to be added.

*   comment

    Comment text.

Example

The following example adds the comment "Working on this" to an incident with the ID 2.

```
emcli -add_comment_to_incident -incident_id="2" -comment="Working on this"
```

add_engr_sys_patches

Adds a system patch for performing a patching operation at the component level, on a specific target, or on a list of targets specified in the target file.

Format

```
emcli add_engr_sys_patches
      -system_target_name="target_name"
      -system_target_type="target_type"
      -target_name="target_name" -target_type="target_type" -
patch_composite_id="patch_composite_id" | -input_file=data:"target list
specified in file"
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- target_name

  Specifies the target name.

- target_type

  Specifies the target type.

- input_file

  Specifies the path of the file that contains the inputs for setting up the patching options.

  The following is an example of an input file to add patches for targets:

  ```
  target.0.target_name=slcm12adm01.example.com
  target.0.target_type=oracle_exadata
  target.0.sys_patch.patch_id=19893788
  target.0.sys_patch.release_id=9800371121010
  target.0.sys_patch.platform_id=226
  target.0.sys_patch.language_id=0
  target.1.target_name=slcm12adm02.example.com
  target.1.target_type=oracle_exadata
  target.1.sys_patch.patch_id=20093789
  target.1.sys_patch.release_id=9800371121010
  target.1.sys_patch.platform_id=226
  target.1.sys_patch.language_id=0
  ```

  The following is an example of an input file to add patches for a component:

  ```
  component.type=Oracle Infiniband Switch
  component.sys_patch.patch_id=19893788
  component.sys_patch.release_id=9800371121010
  component.sys_patch.platform_id=226
  component.sys_patch.language_id=0
  ```

- patch_composite_id

Specifies the patch composite ID. The format of the patch composite ID must follow `p<patchID>_<ReleaseID>_<PlatformID>_<LanguageID>`.

Example

The following example adds patches for the member target `clusteradm0102.example.com:cluster` of the engineered system `DB Machine slcm12.example.com:oracle_dbmachine`, using the inputs specified in `/tmp/property.prop`:

```
emcli add_engr_sys_patches
    -system_target_name="DB Machine slcm12.example.com"
    -system_target_type="oracle_dbmachine"
    -input_file=data:"/tmp/property.prop"
```

add_forwarders_for_paas_agent

Adds forwarders for a given PaaS agent.

Format

```
emcli add_forwarders_for_paas_agent
    -paas_agent_name="paas_agent_name"
    -agent_list="agent_list"
```

Options

- paas_agent_name

  Agent name of the hybrid agent.

- agent_list

  Forwarder agent list separated by a space.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example forwards paas_agent_1 and paas_agent_2 to paas_agent:

```
emcli add_forwarders_for_paas_agent      -paas_agent_name=<paas_agent>      -
agent_list="paas_agent_1,paas_agent_2"
```

add_proxy

Adds a proxy which mediates the https traffic from Oracle Management Server to the Management Agent. Note that this proxy is modeled as an oracle_em_proxy target type. Management Agents can be associated with the proxy in two ways:

1. Using Management Agent names.

2. Using patterns for Management Agent names.

Currently, the oracle_em_proxy target type proxies are supported only for the traffic from Oracle Management Server to the Management Agent. For traffic in the reverse direction (i.e. from the Management Agent to Oracle Management Server), proxy settings can be specified in the corresponding Management Agent's emd.properties file.

ORACLE

Format

```
emcli add_proxy
      -name="<name>"
      -host="<host>"
      -port=<port>
      -protocol=<http | https>
      [-named_credential="<credential name>"]
      [-agents="<name1>,<name2>,..."]
      -agent_patterns="<name pattern1>,<name pattern2>,..."]
      [-excluded_agent_patterns="<name pattern1>,<name pattern2>,..."]
      -oms_agent="<name>"
```

```
[ ] indicates that the parameter is optional.
```

Options

*   name

    Unique name identifying a proxy.

*   host

    Hostname of a machine where the proxy is set up.

*   port

    Port on the proxy host offering the proxy service.

*   protocol

    Protocol used by the traffic which the proxy mediates. The valid values of this option are http or https.

*   named_credential

    Name of the Named Credential to be used for authentication with the proxy.

*   agents

    Comma separated list of the names of Management Agents which the proxy mediates for. The backslash character (\) can be used as an escape character. If the Management Agent with the given name does not exist, it is ignored.

*   agent_patterns

    Comma separated list of patterns for the names of Management Agents which the proxy mediates for. Pattern can use two wildcard characters:Asterisks (*) for one or more characters.Question mark (?) for a single character.Backslash (/) for an escape character.

*   excluded_agent_patterns

    Comma separated list of patterns for the names of Management Agents which need to be excluded from the names of Management Agents identified by Management Agent name patterns associated with the proxy (using the -agent_patterns option). These patterns can also use the same wildcard characters that apply to the -agent_patterns option.

*   oms_agent

    Name of Management Agent set up on Oracle Management Server.

*   parameter_1

Sentence fragment describing the parameter.

Example

The following example adds a new proxy using the options name, host, port, and protocol. In this example, the Management Agents are in the form "host<id>.<domain>.oracle.com:3535" , where 'id' is a 3 digit number ranging from 000 to 999, and domain with values "in", "us", "uk", or "fr". In such a deployment, this proxy is associated with host000 and host001 from the "us" domain and host100 to host179 from all the domains.

```
emcli add_proxy
     -name="us-proxy-4"
     -host="www-proxy-4.us.mycompany.com"
     -port=80
     -protocol=http
     -agents="host000.mycompany.com:3535,host001.mycompany.com:3535"
     -agent_patterns="host1*"
     -excluded_agent_patterns="host18*,host19*"
     -oms_agent="slc02pgw.mycompany.com:1838"
```

add_siteguard_aux_hosts

Associates new auxiliary hosts with the system. An auxiliary host can be any host that is not part of the system but is managed by Enterprise Manager Cloud Control. These hosts can be used to execute any script. Any other targets running on this host will not be part of Site Guard operation plan(s).

Format

```
emcli add_siteguard_aux_hosts
      -system_name="name_of_the_system"
      -host_name="name_of_the_auxiliary_host"
```

[ ] indicates that the parameter is optional

Options

*   **system_name**

    Name of the system.

*   **host_name**

    Name of the auxiliary host that the current user needs to add. This host must be managed by Enterprise Manager Cloud Control.

Example

This example associates auxiliary hosts host1.domain.com and host2.domain.com that are part of austin-system to the system:

```
emcli add_siteguard_aux_hosts
      -system_name="austin-system"
      -host_name="host1.domain.com;host2.domain.com"
```

add_siteguard_script_credential_params

Adds a named credential as a parameter for a Site Guard script. The values of user name and password of this credential can be accessed within the script.

Format

```
emcli add_siteguard_script_credential_params       -script_id="Id associated
with the script"         -credential_name="name of the credential"       [-
credential_owner="credential owner"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   script_id

    The script ID.

*   credential_name

    The name of the credential.

*   credential_owner

    The owner of the credential. This option does not need to be specified if the owner of the credential is same as the logged in user.

Example

The following command adds a script ID and credential name to the siteguard script where the credential owner is SG_ADMIN.

```
emcli add_siteguard_script_credential_params
     -script_id="2"
     -credential_name="NAMED_CREDENTIAL_Y"
     -credential_owner="SG_ADMIN"
```

See Also

get_siteguard_script_credential_params

add_siteguard_script_hosts

Adds a host to the Site Guard configuration scripts.

Format

```
emcli add_siteguard_script_hosts
     -script_id=<script_id>
     -host_name=<name1;name2;...>
```

Options

*   **script_id**

    ID associated with the script.

*   **host_name**

Name of the host where this script will be run. You can specify more than one host name.

Example

The following example adds the `host1.domain.com` host to the Site Guard configuration script with the ID 10.

```
emcli add_siteguard_script_hosts
      -script_id="10"
      -host_name ="host1.domain.com"
```

See Also

get_siteguard_script_credential_params
create_siteguard_script
get_siteguard_script_hosts

add_swlib_storage_location

Adds a storage location in the software library.

Format

```
emcli add_swlib_storage_location
      -name="location_name"
      -path="location_path"
      [-type="OmsShared|OmsAgent|Http|Nfs|ExtAgent"]
      [-host="hostname"]
      [-credential_set_name="setname"] | [-credential_name="name" - credential_
        owner="owner"]

[ ]  indicates that the parameter is optional
```

Options

• **name**

  Name of the storage location.

• **path**

  Path of the storage location, which can be a file system path or a URL, depending on the storage type chosen.

• **type**

  Type of storage location. The default is OmsShared.

• **host**

  Target name of the host where the path for the storage location exists. This option is required for storage types OmsAgent, Nfs, and ExtAgent. For the Nfs storage type, the host is not required to be a target in Enterprise Manager.

• **credential_set_name**

  Set name of the preferred credential stored in the repository for the host target. This is a requiredoption for storage types OmsAgent and ExtAgent. The set names can be one of the following:

  – HostCredsNormal: Default unprivileged credential set

  – HostCredsPriv: Privileged credential set

- **credential_name**

  Name of a named credential stored in the repository. this option is required for storage types OmsAgent and ExtAgent. This option must be specified together with the credential_owner option.

- **credential_owner**

  Owner of a named credential stored in the repository. This option is required for storage types OmsAgent and ExtAgent. This option must be specified together with the credential_name option.

Example

This example adds an OMS Agent File system storage location named myOMSAgtLocation for the path /u01/swlib' on host 'fs1.us.acme.com. The named credential MyAcmeCreds owned by ACME_USER is used for reading/writing files from this location.

```
emcli add_swlib_storage_location
      -name="myOMSAgtLocation"
      -path="/u01/swlib"
      -type="OmsAgent"
      -host="fs1.us.acme.com"
      -credential_name="MyAcmeCreds"
      -credential_owner="ACME_USER"
```

add_target

Adds a target to be monitored by Enterprise Manager. The target type specified is checked on the Management Agent for existence and for required properties, such as user name and password for host target types, or log-in credentials for database target types. You must specify any required properties of a target type when adding a new target of this type.

For `oracle_database` target types, you must specify Role with the monitoring credentials. If the Role is Normal, the UserName must be `dbsnmp`. Otherwise, the Role must be SYSDBA, and UserName can be any user with SYSDBA privileges.

> **Note:**
>
> You cannot use this verb for composite targets. The verb does not support adding an association between a parent target such as IAS and a child target such as OC4J. t db instance targets,You must use the instances option to specify DB instances before adding the cluster database.

Standard Mode

```
emcli add_target
      -name="name"
      -type="type"
      [-host="hostname"]
      [-properties="pname1:pval1;pname2:pval2;..."]
      [-separator=properties="sep_string"]
      [-subseparator=properties="subsep_string"]
      [-credentials="userpropname:username;pwdpropname:password;..."]
```

```
[-input_file="parameter_tag:file_path"]
[-display_name="display_name"]
[-groups="groupname1:grouptype1;groupname2:grouptype2;..."]
[-timezone_region="gmt_offset"]
[-monitor_mode="monitor_mode"]
[-instances="rac_database_instance_target_name1:target_type1;..."]
[-monitoring_cred="HostCredsPriv;host;HostCreds;HostUserName;foo;..."]     [-
access_point_name="ap_name"]      [-access_point_type="ap_type"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Target name. Names cannot contain colons (:), semi-colons (;), or any leading or trailing blanks.

- **type**

  Target type. Standard target types include: `host`, `oracle_database`, `oracle_apache`, `oracle_listener`, and `oracle_emd`. To see all available target types available for your environment, check the `$AGENT_HOME/sysman/admin/metadata` directory. A metadata file (XML) exists for each target type.

- **host**

  Network name of the system running the Management Agent that is collecting data for this target instance.

- **properties**

  Name-value pair (that is, `prop_name:prop_value`) list of properties for the target instance. The "name"(s) are identified in the target-type metadata definition. They must appear exactly as they are defined in this file. Metadata files are located in `$AGENT_HOME/sysman/admin/metadata`.

  > **✎ Note:**
  >
  > This verb does not support setting global target properties. It is recommended that you use set_target_property_values to set target properties.

- **separator=properties**

  Specify a string delimiter to use between name-value pairs for the value of the -properties. The default separator delimiter is ";".

  For more information about the separator option, see -input_file Syntax Guidelines.

- **subseparator=properties**

  Specifies a string delimiter to use between the name and value in each name-value pair for the value of the -properties option. The default subseparator delimiter is ":".

  For more information about the subsepator option, see -input_file Syntax Guidelines.

- **credentials**

  Monitoring credentials (name-value pairs) for the target instance. The "name"(s) are identified in the target-type metadata definition as credential properties. The credentials

must be specified exactly as they are defined in the target's metadata file. Metadata files are located in `$AGENT_HOME/sysman/admin/metadata`.

- **input_file**

  Used in conjunction with the `-credentials` option, this enables you to store specific target monitoring credential values, such as passwords, in a separate file. The `-input_file` specifies a mapping between a tag and a local file path. The tag is specified in lieu of specific monitoring credentials of the `-credentials` option. The tag must not contain colons (:) or semi-colons (;).

  For more information about the input_file option, see -input_file Syntax Guidelines.

- **display_name**

  Target name displayed in the Enterprise Manager Cloud Control console.

- **groups**

  Name-value pair list of the groups to which this target instance belongs. Follows the format of `groupname:grouptype;groupname2:grouptype2`.

- **timezone_region**

  GMT offset for this target instance. (-7 or -04:00 are acceptable formats.)

- **monitor_mode**

  Either 0, 1, or 2. The default is 0. 1 specifies OMS mediated monitoring, and 2 specifies Agent mediated monitoring.

- **instances**

  Name-value pair list of RAC database instances that the RAC database target has. Database instance targets must be added before trying to add the cluster database.

- **force**

  Forces the target to be added even if the target with the same name exists. Updates the properties of the target with your latest input.

- **timeout**

  Time in seconds for the command to wait to add the target to the Agent. The default is 10 minutes.

- **monitoring_cred**

  Sets a monitoring credential set for this target.

- **Separator=monitoring_cred**

  Specify a string delimiter to use between name-value pairs for the value of the -monitoring_cred option. The default separator delimiter is ";".

- **Server Discovery**

  Use the following syntax for server discovery:

  ```
  emcli add_target
      -name=${TARGET_NAME}
      -type=oracle_si_server_map
      -host=${AGENT_HOST}
      -access_point_name=${TARGET_NAME_AP}
      -access_point_type='oracle_si_server_ilom'
      -properties='dispatch.url=ilom-ssh://${TARGET_IP}:22'
  ```

```
                    -subseparator=properties='='
                    -monitoring_cred='ilom_creds_set;oracle_si_server_ilom;ilom_
                         creds;username:${USERNAME};password:${PASSWORD}'
                    -monitoring_cred='ssh_creds_set;oracle_si_server_ilom;ssh_
                         creds;username:${USERNAME};userpass:${PASSWORD}'
```

- **subseparator=monitoring_cred**

  Specify a string delimiter to use between name and value in each name-value pair for the value of the -monitoring_cred option. The default subseparator delimiter is ":".

- **access_point_name**

  Name of the access point target to be added which is tagged to given server target.

- **access_point_type**

  Type of the access point target to be added which is tagged to given server target.

Example

Example 1

The following example adds an `oracle_database` target with the name "database." Note how the credentials are specified. The names in the name-value pairs come from the `oracle_database` metadata file. They must appear exactly as they are named in that file. This also applies for the property names. The example uses the base minimum of required credentials and properties for the database target.

```
emcli add_target
     -name="database"
     -type="oracle_database"
     -host="myhost.us.example.com"
     -credentials="UserName:dbsnmp;password:dbsnmp;Role:Normal"
     -properties="SID:semcli;Port:15091;OracleHome:/oracle;
      MachineName:smpamp-example.com"
     -groups="Group1:group;Group2:group"
```

Example 2

The following example adds a standalone Oracle HTTP Server.

```
emcli add_target
     -host=test.mycompany.com
     -name=test13c
     -type=oracle_apache
     -properties=
"OracleHome->/scratch/smariswa/test13c|ConfigPath->/scratch/user1/test13c/
user_projects/domains/test1213_domain/config/fmwconfig/components/OHS/ohs1|
EMTargetType->oracle_apache|OracleInstance->na|compVersion->na|LifeCycleMBeanName->na|
OPMNMBeanName->na| monMode->na|ProxyMBeanObjectName->na|ServerNames->na|CanonicalPath-
> /instance1/ohs1/|HTTPMachine->blr2201947.mycompany..com|HTTPPort->7778|version->13.1|
NMUser->weblogic|NMPwd->welcome1" -separator=properties="|" -subseparator=properties="-
>"
```

Example 3

The following example adds an `oracle_database` target with the name "database." Note how the credentials are specified. The names in the name-value pairs come from the `oracle_database` metadata file. They must appear exactly as they are named in that file. This also applies for the property names. The example uses the base minimum of required credentials and properties for the database target.

```
emcli add_target
      -name="database"
      -type="oracle_database"
      -host="myhost.us.example.com"
      -credentials="UserName:dbsnmp;password:dbsnmp;Role:Normal"
      -properties="SID:semcli;Port:15091;OracleHome:/oracle;
       MachineName:smpamp-example.com"
      -groups="Group1:group;Group2:group"
```

Example 4

The following example discovers the server ILOM_SPARC at the target IP
10.133.245.231 using the user `root` and the password `admin123`.

```
emcli add_target
      -name=ILOM_SPARC
      -type=oracle_si_server_map
      -host=`hostname`.mycompany.com
      -access_point_name='ILOM_SPARC_AP'
      -access_point_type='oracle_si_server_ilom'
      -properties='dispatch.url=ilom-ssh://10.133.245.231:22'
      -subseparator=properties='='
      -monitoring_cred='ilom_creds_set;oracle_si_server_ilom;ilom_
          creds;username:root;password:admin123'
    -monitoring_cred='ssh_creds_set;oracle_si_server_ilom;ssh_
          creds;username:root;userpass:admin123'
```

Example 5

The following example adds an Apache Tomcat version 8 target with the name
"TARGET_NAME."

```
emcli add_target
        -name="TARGET_NAME"
        -type="tomcat"
        -host="XYZ.oracle.com"
        -
properties="Host:XYZ.oracle.com;JMXPort:65432;JMXProtocol:rmi;JMXService:jmxrmi;
          VersionCategory:8;SSLTrustStore:AGENT_HOME/agent_inst/sysman/config/
montrus t/AgentTrust.jks"
        -
credentials="JMXUserName:adminUser1;JMXPassword:password1;SSLTrustStorePassword:w
elcome"
```

add_target_property

Adds a new target property for a given target type. All targets of this target type will
have this new target property.

Format

```
emcli add_target_property
        -target_type="target_type"
        -property="prop_name"
```

Options

- **target_type**

  Target type for which this property needs to be added. To add this property to all
  existing target types, you can specify a "*" wildcard character.

- **property**

  Name of the property to be created for this target type. Property names are case-sensitive. The property name cannot be the same as the following Oracle-provided target property names (in English):

  Comment, Deployment Type, Line of Business, Location, Contact

Example

This example adds the Owner Name property for all targets of type oracle_database.

```
emcli add_target_property -target_type="oracle_database" -property="Owner Name"
```

add_target_to_rule_set

Adds a target to an enterprise rule set.

Privilege Requirements: A Super Administrator can add a target to any enterprise rule set except for predefined (out-of-box) rule sets supplied by Oracle.Only the owner or co-author of a rule set can add a target to it.

Format

```
emcli add_target_to_rule_set
      -rule_set_name="rule set name"
      -target_name="target name"
      -target_type="internal name for the target type. For example, host"
      [-rule_set_owner=<ruleset owner>]

[ ]  indicates that the parameter is optional
```

Options

- **rule_set_name**

  Name of an enterprise rule set. This option only applies to rule sets that are associated with a list of targets.

- **target_name**

  Name of the target to be added.

- **target_type**

  Target type of the target to be added. For example, host.

- **rule_set_owner**

  Optionally, you can specify the owner of the rule set.

Example

This example adds the host target *myhost.com* to a rule set named *rules*. This rule set is owned by the administrator *sysman*.

```
emcli add_target_to_rule_set -rule_set_name='rules' -target_name='myhost.com' -
target_type='host' -rule_set_owner='sysman'
```

add_to_target_properties_master_list

Adds the property to the property master list.

Format

Standard Mode

```
emcli add_to_target_properties_master_list
      -property_name="null"
      -property_value="null"
```

Interactive or Script Mode

```
emcli add_to_target_properties_master_list(
     property_name="null"
     ,property_value="null"
)
```

Options

* property_name

  The name of the property to be added.

* property_value

  The value of the property to be added.

Exit Codes

Enter exit codes, if available, or delete if no exit codes. For example:

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

add_virtual_platform

Adds Oracle Virtual Platform(s) to remotely monitor Xen-based Hypervisor(s). The associated Oracle Server and Oracle Virtual Server running on the Hypervisor will is also added.

You can add multiple Hypervisors at the same time. The command returns the name and the execution identifier of the job submitted to add the target(s).

To delete an Oracle Virtual Platform and its related targets, use the delete_target verb.

Format

```
emcli add_virtual_platform
      -name="host_name/IP_address_or_list_from_an_input_file"
      -agent="agent_target_name"
      [-failover_agent="failover_agent_target_name"]
      -credentials="property_name1:property_value1;property_name2:
          property_value2;..."
          [-wait_for_completion=true|false]
          [-wait_for_completion_timeout=<time_in_minutes>]
      [-separator=credentials="separator_for_key_value_pairs"]
          [-subseparator=credentials="separator_for_key_value_pair"]
      [-input_file="FILE:file_path_or_name:FILE"]

[ ]  indicates that the parameter is optional
```

Options

* **name**

IP address or host name of the Xen-based Hypervisor being added as an Oracle Virtual Platform in Enterprise Manager. There are two ways to provide this value. For only one target, you can directly pass this value at the command line with the name of the Host or the IP address. For multiple values, you can use the "-input_file" option and list the host names, IP addresses, or an IP address range from a file by passing the name of the input file. A new line is used to delimit the host names or the IP addresses. You can specify the host name of a machine, an IP address, or an IP address range on each line.

See the examples for details.

- **agent**

  Target name of the primary agent used to monitor the Oracle Virtual Platform(s) and related targets.

- **failover_agent**

  Target name of the failover agent used to monitor the Oracle Virtual Platform(s) and related targets.

- **credentials**

  Monitoring credentials (name-value pairs) for the target instance. The "names" are defined in the target type metadata definition as credential properties. Metadata files are located at $AGENT_HOME/sysman/admin/metadata.

  See the examples for details on various options.

- **wait_for_completion**

  Flag to indicate if the CLI is going to wait for the submitted job to finish. The default value is false. If the value is true, the progress of the job is printed on the command line as and when the addition of Oracle Virtual Platform(s) Succeeds/Fails.

- **wait_for_completion_timeout**

  Time in minutes after which CLI stops waiting for the job to finish. This option is honored only if the value for the wait_for_completion option is true. A negative or zero value does not wait for the job to finish.

  See the examples for details.

- separator=credentials

  Custom separator for the credential key value pairs. Specify a string delimiter to use between name-value pairs for the values of the -credentials option. The default separator delimiter is ";".

  For more information about the separator option, see -input_file Syntax Guidelines.

- subseparator=credentials

  Custom separator for a key value pair. Specify a string delimiter to use between name and value in each name-value pair for the values of the -credentials option. The default separator delimiter is ":".

  For more information about the subseparator option, see -input_file Syntax Guidelines.

- **input_file**

  Optionally use in conjunction with the -credentials option. You can use this option to set specific target monitoring credential values, such as passwords or SSH keys, in a separate file.

  This option specifies a mapping between a tag and a local file path. The tag is specified in lieu of specific -credentials property values.

Examples

Example 1

This example adds an Oracle Virtual Platform with a Unix Sudo user who requires SSH key Passphraseless-based authentication. The SSH private key, SSH public key, and password are read from input files.

```
emcli add_virtual_platform
      -name=example.com
      -agent=example.com:1838
      -credentials='type:DMOvsSshKeyCreds;PrivilegeType:sudo;privilegedUser:true;
           RunAs:root;PrivilegeCommand:/usr/bin/sudo -S -u %RUN_AS% %COMMAND%;
           EnablePseudoTerminal:false;SshPrivateKey:PRIVATE_KEY;
           SshPublicKey:PUBLIC_KEY;OVSUsername:sudoer1;OVSPassword:PWD_FILE'
      -input_file='PRIVATE_KEY:id_dsa'
      -input_file='PUBLIC_KEY:id_dsa.pub'
      -input_file='PWD_FILE:password'
```

Example 2

This example adds an Oracle Virtual Platform with a Unix PowerBroker user who requires SSH key Passphraseless based authentication. The SSH private key, SSH public key, and password are read from input files.

```
emcli add_virtual_platform
      -name=example.com
      -agent=example.com:1838
      -credentials='type:DMOvsSshKeyCreds;PrivilegeType:powerbroker;
           privilegedUser:true;RunAs:root;PrivilegeCommand:
           /usr/bin/pbrun -l -u %RUN_AS% %COMMAND%;
           PowerBrokerProfile:profile;PowerBrokerPasswordPrompt:myprompt;
           SshPrivateKey:PRIVATE_KEY;SshPublicKey:PUBLIC_KEY;
           OVSUsername:myuser;OVSPassword:PWD_FILE'
      -input_file='PRIVATE_KEY:id_dsa'
      -input_file='PUBLIC_KEY:id_dsa.pub'
      -input_file='PWD_FILE:password'
```

allocate_quota

Allocates quota to an assignee.

Format

Interactive or Script Mode

```
emcli allocate_quota
      assignee_name="assignee name"
      assignee_type="assignee type"
      [quota="quota"]
```

Options

- assignee_name

  The name of the assignee to whom the quota will be allocated.

- assignee_type

  The type of assignee. Valid values are: Tenant, User_Profile, or User.

- quota

  The quota that will be allocated for the assigned user. Quota allocations can be defined:

  – at the entity level.

  – on a quota assignable object if the selected entity supports quota allocation on objects. When quota is allocated on an object, all quota related computations will be done in the context of that object.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example allocates quota on a quota assignable object.

```
emcli allocate_quota
      -assignee_name=T1
      -assignee_type=tenant
      -quota=Component1:Entity1;0.1:Gb;object_name=OBJ1:object_type=type1
```

analyze_unconverted_udms

Analyzes UDMs and lists unique UDMs, any possible matches, and templates that can apply these matching metric extensions.

Format

```
emcli analyze_unconverted_udms
      [-session_id=<sessionId>]

[ ]  indicates that the parameter is optional
```

Options

- **session_id**

  ID of a session to be analyzed. Not specifying a session ID creates an analysis session that contains all unconverted UDMs. You can specify this session ID in future invocations to generate a fresh analysis.

Example

This example lists matches for all unconverted UDMs in the specified migration session.

```
emcli list_unconverted_udms -session_id=<sessionId>
```

apply_diagcheck_exclude

Applies a diagnostic check exclusion to a set of target instances. You can exclude certain diagnostic checks by defining an exclusion name. This rule is applied when all diagnostic checks are evaluated for the particular target type so that the checks specified in the rule are excluded.

Format

```
emcli apply_diagcheck_exclude
      -target_type="type"
```

```
           -exclude_name="name"
           [-target_name="target_name" ]*
```

[ ] indicates that the parameter is optional

Options

- **target_type**

  Type of target.

- **exclude_name**

  Name to use for the exclusion. To create the exclude_name, use the define_diagcheck_exclude verb.

- **target_name**

  Target names to apply the exclusion to.

apply_privilege_delegation_setting

Activates Sudo or PowerBroker settings for specified targets.

Standard Mode

```
emcli apply_privilege_delegation_setting
        -setting_name="setting"
        -target_type="host/composite"
        [-target_names="name1;name2;..."]
        [-input_file="FILE:file_path"]
        [-force="yes/no"]
```

[ ] indicates that the parameter is optional

Interactive or Script Mode

```
apply_privilege_delegation_setting
        (setting_name="setting"
        ,target_type="host/composite"
        [,target_names="name1;name2;..."]
        [,input_file="FILE:file_path"]
        [,force="yes/no"])
```

[ ] indicates that the parameter is optional

Options

- **setting_name**

  Name of the setting you want to apply.

- **target_names**

  List of target names. The newly submitted setting applies to this list of Enterprise Manager targets.

  – All targets must be of the same type.

  – The target list must not contain more than one element if the element's target type is "group."

  – The group referenced above should have at least one host target.

- **target_type**

  Type of targets to which the setting is applied. Valid target types are "host" or "composite" (group).

- **input_file**

  Path of the file that has target names. This enables you to pass targets in a separate file. The file cannot contain any colons (:) or semi-colons (;).

  For more information about the input_file option, see -input_file Syntax Guidelines.

- **force**

  If `yes`, the operation continues and ignores any invalid targets. The default is `no`.

Examples

Example 1

This example applies a privilege setting named `sudo_setting`. This setting applies to targets of type `host`, and it is being applied to `host1`, `host2`, and so forth.

```
emcli apply_privilege_delegation_setting
      -setting_name=sudo_setting
      -target_type=host
      -target_names="host1;host2;"
```

Example 2

This example applies a privilege setting named `sudo_setting`. This setting applies to targets of type `host`, and it is being applied to `host1`, `host2`, and so forth. The `force` flag indicates that the setting is applied to all valid targets, and invalid targets are ignored.

```
emcli apply_privilege_delegation_setting
      -setting_name=sudo_setting
      -target_type=host
      -target_names="host1;host2;"
      -force=yes
```

apply_template

Applies a monitoring template to a list of specified targets. The options to the verb can be supplied in any order.

Format

```
emcli apply_template
      -name="template_name"
      -targets="tname1: ttype1;tname2: ttype2;..."
      [-copy_flags="0" or "1" or "2"]
      [-replace_metrics="0" or "1"]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Template name as it exists in the database. Names cannot contain colons (:), semi-colons (;), or any leading or trailing blanks.

- **targets**

  The targets should be specified in the following sequence:

  TargetName1:TargetType1;TargetName2:TargetType2

  For example:

  ```
  db1:oracle_database;my db group:composite
  ```

  A semi-colon is the target separator. Ideally, non-composite targets should be of the target type applicable to the template. If not, the template is not applied to the indicated target. For composite targets, the template is applied only to the member targets that belong to the target type for which the template is applicable.

- **copy_flags**

  This applies only for metrics with multiple thresholds.

  '0' indicates: Apply threshold settings for key values common to the template and target.

  '1' indicates: Remove key value threshold settings in the target and replace them with key value threshold settings from the template.

  '2' indicates: Apply threshold settings for all key values defined in the template. The default is '0'.

- **replace_metrics**

  0 indicates that the thresholds of the metrics not included in the template but available in the target will not be changed. This is the default value. 1 indicates that the thresholds of the metrics present in the target, but not in the template, will be set to NULL. That is, such metrics in the target will not be monitored and therefore, no alert will be raised for them.

Examples

Example 1

This example applies a monitoring template named `my_db_template`. This template applies to targets of type `oracle_database`, and it is being applied to `db1` , which is of type `oracle_database`, and `my_db_group`, which is of type `composite`.

For composite targets, the template is only applied to member targets that belong to the target type for which the template is applicable. Since the `copy_flags` is not specified, the default ("Apply threshold settings for monitored objects common to both template and target") is meant.

```
emcli apply_template
      -name="my_db_template"
      -targets="db1:oracle_database;my_db_group:composite"
```

Example 2

This example applies the monitoring template named my_db_template. This template is applicable to targets of type oracle_database. This command applies this template to two targets: target db1 of type oracle_database and target my_db_group of type composite.

For composite targets, the template is applied only to the member targets that belong to the target type for which the template is applicable. In this case, since the copy_flags option is specified as "1", the template is superimposed on the target. All

keys in the template are copied to the target, and any extra keys present in the target are deleted.

The replace_metrics flag set to 1 denotes that the thresholds of the metrics present in the target, but not in the template, are set to NULL. That is, these metrics in the target are not monitored, and therefore, no alert is raised for them.

```
emcli apply_template -name="my_db_template"
        -targets="db1:oracle_database;my_db_group:composite"
        -copy_flags="1" -replace_metrics="1"
```

apply_template_tests

Applies the variables and test definitions from the file(s) into a repository target.

Format

```
emcli apply_template_tests
     -targetName=target_name
     -targetType=target_type
     -input_file=template:template_filename
     [-input_file=variables:<variable_filename>]
     [-input_file=atsBundleZip:<ats_bundle_zip_filename>]
     [-useBundleDatabankFile]
     [-useFirstRowValues]
     [-overwriteExisting=all | none | <test1>:<type1>;<test2>:<type2>;...]
     [-encryption_key=key]
     [-swlibURN=<URN_for_swlib_entity>]
     [-swlibPath=<Path_for_swlib_entity>]

[ ]  indicates that the parameter is optional
```

Options

- **targetName**

  Target name.

- **targetType**

  Target type.

- **input_file=template**

  Name of the input file containing the test definitions.

  For more information about the input_file option, see -input_file Syntax Guidelines.

- **input_file=variables**

  Name of the input file containing the variable definitions. If this attribute is not specified, the variables are extracted from the same file containing the test definitions.

  The variables file format is as follows:

  ```
  <variables xmlns="template">
  <variable name="<name1>" value="<value1>"/>
  <variable name="<name2>" value="<value2>"/>
  ...
  </variables>
  ```

  For more information about the input_file option, see -input_file Syntax Guidelines.

- **input_file=atsBundleZip**

Name of the ATS bundle zip defined in the template.

For more information about the input_file option, see -input_file Syntax Guidelines.

- **useBundleDatabankFile**

  If you specify this option, the bundle databank files are used.

- **useFirstRowValues**

  If you specify this option, the first row values are used.

- **overwriteExisting**

  Specifies which tests should be overwritten in case they already exist on the target. The possible values are:

  1. 'none' (default): None of the existing tests on the target will be overwritten.

  2. 'all': If a test with the same name exists on the target, it will be overwritten with the test definition specified in the template file.

  3. <test1>:<type1>;<test2>:<type2>;...: If any of the tests with names <test1>, <test2>, and so forth exist on the target, they are overwritten with the definition in the template file.

- **encryption_key**

  Optional key to decrypt the file contents. This key should be the same as the one used to encrypt the file.

- **swlibURN**

  Loads the software library entity through an URN. The respective entity data such as OATZ zip file and Zip File Name will be associated to the new service test. Either this option or the -swlibPath option are required to associate the OATS zip file to the service test.

- **swlibPath**

  Loads the software library entity through an entity path. The respective entity data such as OATZ zip file and Zip File Name will be associated to the new service test. Either this option or the -swlibURN option are required to associate the OATS zip file to the service test.

Examples

You must have the following privileges to perform these examples:

- Operator privilege on the target.

- Operator privilege on all beacons currently monitoring the target. Alternatively, you must have the "use any beacon" privilege.

Example 1

This example applies the test definitions contained in the file `my_template.xml` into the Generic Service target `my_target`, using the key `my_password` to decrypt the file contents. If tests with names `my_website` or `my_script` exist on the target, they are overwritten by the test definitions in the file.

```
emcli apply_template_tests
    -targetName='my_target' -targetType='generic_service'
    -input_file=template:'my_template.xml' -encryption_key='my_password'
    -overwriteExisting='my_website:HTTP;my_script:OS'
```

Example 2

This example applies the test definitions contained in file my_template.xml into the Web Application target my_target using the variable values specified in file my_variables.xml. If any tests in the target have the same name as tests specified in the template file, they are overwritten.

```
emcli apply_template_tests
        -targetName='my_target' -targetType='website'
        -input_file=template:'my_template.xml' -input_file=variables:
          'my_variables.xml'
        -overwriteExisting='all'
```

apply_update

Applies an update.

Format

```
emcli apply_update
        -id="internal id"
```

Options

• **id**

Internal identification for the update to be applied.

Example

This example submits a job to apply an update, and prints the job execution ID upon submission.

```
emcli apply_update
        -id="914E3E0F9DB98DECE040E80A2C5233EB"
```

argfile

Executes one or more EM CLI verbs, where both verbs and the associated arguments are contained in an ASCII file. argfile enables you to use verbs with greater flexibility. For example, when specifying a large list of targets to be blacked out (`create_blackout` verb), you can use the `argfile` verb to input the target list from a file.

Multiple EM CLI verb invocations are permitted in this file. You should separate each verb invocation with a new line.

Format

```
emcli argfile <file_name>
      [-delim=<delimiter_string>]

[ ]  indicates that the parameter is optional
```

Options

• **delim**

String used as a delimiter between two verbs in the argument file. The default delimiter is a newline character.

assign_charge_plan

Assigns a charge plan to the given entity.

Format

```
assign_charge_plan
     -entity_name="eName"
     -entity_type="eType"
     -plan_name="pName"
     -[entity_guid="entity_guid"]

[ ]  indicates that the parameter is optional
```

Options

*   **entity_name**

    Name of the entity for which the charge plan is to be assigned.

*   **entity_type**

    Type of entity for which the charge plan is to be assigned.

*   **plan_name**

    Name of the charge plan to be assigned.

*   **entity_guid**

    guid of the entity to be added to Chargeback.

    When more than one entity is active in Chargeback with the given entity name and entity type, the command lists all such entities with additional details such as creation date, parent entity name, entity guid, and so forth to choose the correct entity. Select the correct entity from the given list and execute the command again with entity guid as the parameter instead of entity name and entity type.

Example

This example assigns charge plan "plan1" to "db1", an oracle_database entity.

```
emcli assign_charge_plan -entity_name="db1" -entity_type="oracle_database" -
plan_name="plan1"
```

assign_compliance_ca

Triggers a corrective action for a specified rule.

Format

```
emcli assign_compliance_ca
     -rule_iname="<rule_internal_name>"
     -target_type="<target_type>"
     -ca_name="<correction_action_name>"
```

Options

*   rule_iname

Internal name of the compliance standard rule. Rule internal names are available in the MGMT$CS_RULE_ATTRS view.

- target_type

  Target type associated with the compliance standard rule.

- ca_name

  Name of the corrective action.

Example

The following example assigns the my_action corrective action to the my_rule compliance standard rule which is located in the host target.

```
emcli assign_compliance_ca
      -rule_iname="my_rule"
      -target_type="host"
      -ca_name="my_action"
```

assign_cost_center

Assigns the cost center to the given entity.

Format

```
assign_cost_center
      -entity_name="eName"
      -entity_type="eType"
      -cost_center_name="cName"
      -[entity_guid="entity guid" ]

[ ]  indicates that the parameter is optional
```

Options

- **entity_name**

  Name of the entity for which the cost center is to be assigned.

- **entity_type**

  Type of entity for which the cost center is to be assigned.

- **cost_center**

  Name of the cost center to be assigned.

- **entity_guid**

  guid of the entity in Chargeback.

  When more than one entity is active in Chargeback with the given entity name and entity type, the command lists all such entities with additional details such as creation date, parent entity name, entity guid, and so forth to choose the correct entity. Select the correct entity from the given list and execute the command again with entity guid as the parameter instead of entity name and entity type.

Example

This example assigns the cost center "cc1" to "db1", an oracle_database entity.

```
emcli assign_cost_center -entity_name="db1" -entity_type="oracle_database" -
cost_center_name="cc1"
```

assign_csi_at_target_level

Assigns or updates the Customer Support IDentifier (CSI) to the given target name
and type.

Format

```
emcli assign_csi_at_target_level
      -target_name="Target_name"
      -target_type="Target_type"
      -csi="Customer_Support_IDentifier_value"
      -mos_id="My_Oracle_Support_ID"
```

Options

*   **target_name**

    Name of the Cloud Control target.

*   **target_type**

    Type of Cloud Control target

*   csi

    Customer Support Identifier value to be assigned.

*   mos_id

    My Oracle Support (MOS) user ID.

Example

```
emcli assign_csi_at_target_level
      -target_name="myhost.us.example.com"
      -target_type="oracle_example_type"
      -csi=12345678
      -mos_id="abc@xyz.com"
```

assign_csi_for_dbmachine_targets

Assigns or updates the Customer Support Identifier (CSI) for all of the associated
Exadata, RAC, and database targets for a database machine name.

Format

```
emcli assign_csi_for_dbmachine_targets
      -target_name="database_system_name"
      -csi="customer_support_identifier_value"
      -mos_id="my_oracle_support_ID"
```

Options

*   **target_name**

    Name of the database system target.

*   **csi**

    Customer Support Identifier (CSI) to be assigned.

- **mos_id**

  My Oracle Support (MOS) user ID.

Example

This example assigns the CSI 1234567 to database system abcdef.company.com.

```
emcli assign_csi_for_dbmachine_targets
    -target_name=abcdef.company.com
    -csi=1234567
    -mos_id=abc@xyz.com
```

assign_test_to_target

Assigns a test-type to a target-type. If a test-type t is assigned to target-type T, all targets of type T can be queried with tests of type t.

Format

```
emcli assign_test_to_target
    -testtype=test-type_to_be_assigned
    -type=target_type
    [-tgtVersion]=version_of_target_type
```

```
[ ]  indicates that the parameter is optional
```

Options

- **testtype**

  Test-type to be assigned. Should be the internal name; that is, 'HTTP' instead of 'Web Transaction'.

- **type**

  Service target type.

- **tgtVersion**

  Version of the target type. If not specified, the latest version is used.

Example

This example assigns test type HTTP to targets of type generic service v2.

```
emcli assign_test_to_target
    -testtype='HTTP'
    -type='generic_service'
    -tgtVersion='2.0'
```

associate_cs_group_targets

Associates the specified compliance standard with the specified groups of targets for test mode or normal mode. Test mode is applicable only for drift and consistency compliance standards.

Format

```
emcli associate_cs_group_targets
    -cs_iname="<internal_name_of_standard>"
```

```
    -author="<author>"
    -version="<version>"
    (-group_names="<group_name_list>" | -group_names_file="<file_name>)"
```

Options

- cs_iname

  Internal name of the compliance standard.

- author

  Author of the compliance standard.

- version

  Version of the compliance standard.

- group_names

  Comma separated list of group names.

- group_names_file

  Name of the file that contains the group names. The group names can be either comma-separated values or in a file where the group names are listed on separate lines. Examples are:

  -group_names_file=group1,group2,group3

  -group_names_file="group.txt" where group.txt contains the following lines:

      group1
      group2
      group3

  Note: Use either group_names or group_names_file.

Example

The following example creates associations for the second version of the security standard, authored by Jones, for the groups named tgt_grp1 and tgt_grp2.

```
emcli associate_cs_group_targets
    -cs_iname="security_standard"
    -author="Jones"
    -version="2"
    -group_names="tgt_grp1,tgt_grp2"
```

associate_cs_targets

Associates the specified standard with the listed targets.

Note: When the standard is provided by Oracle, the <std_name> is the standard internal name.

Format

```
associate_cs_targets
 -name="<std_name>"
 -version="<std_version>"
 -author="<author_name>"
 -target_list="<target_name>[,<target_name>,<group_name>:Group]*"
```

```
-target_list_file="<file_name>"
```

```
[ ] indicates that this option is optional
```

Options

- name

  Name of the standard.

- version

  Version of the standard.

- author

  Author of the standard. When the standard is provided by Oracle, the <std_name> is the standard internal name, for example, sysman.

- target_list

  Name of the target. This option is useful when a compliance standard is to be associated with one or a small number of targets. Targets are separated by commas. When providing a group target, it should be appended with ":Group". Examples are:

  -target_list="slc0host"

  -target_list="slc0host,slc-host01"

  -target_list="slc0host,host_grps:Group"

- target_list_file

  Name of the file that contains the list of targets. The targets can be either comma-separated values or in a file where the targets are listed on separate lines. Examples are:

  -target_list_file=slc0host,slc0host1,slc0host02

  -target_list_file="slc0host.txt" Where slc0host.txt contains the following lines:

  slc0host

  slc0host01

  slc0host02

  Note: Use either the target_list option or the target_list_file option.

Examples

Example 1

The following example specifies the target_list.

```
emcli associate_cs_targets
-name="secure configuration for host"
-version="1"
-author="sysman"
-target_list="host1,host2,group1:Group"
```

Example 2

The following example specifies the target_list_file.

```
emcli associate_cs_targets
-name="secure configuration for host"
-version="1"
```

```
-author="sysman"
-target_list_file="file with target name list"
```

associate_target_to_adm

Associates a target to an existing Application Data Model.

Format

```
emcli associate_target_to_adm
    -adm_name=<application_data_model_name>
    -target_name=<target_name>
    -target_type=<target_type>
```

[ ] indicates that the parameter is optional

Options

*   **adm_name**

    Application Data Model name to which the target will be associated.

*   **target_name**

    Application Data Model name to which the target will be associated.

*   **target_type**

    Type of target that will be associated with the Application Data Model.

Output

Success/error messages

Example

This example associates target test_database to the Application Data Model named Sample_ADM.

```
emcli associate_target_to_adm      -adm_name=Sample_ADM       -
target_name=test_database
      -target_type=oracle_pdb
```

associate_user_profile

Associates a user profile with a set of users.

Format

Standard Mode

```
emcli associate_user_profile
    -name="profile name"
    -users="users to be associated"
```

Interactive or Script Mode

```
emcli associate_user_profile(
    name="profile name",
    users="users to be associated"      )
```

Options

- name

  Identifies the name of the user profile.

- users

  Identifies the name of the users with whom the user profile is to be associated.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

This example associates the `profile1` user profile with users `user1` and `user2`.

```
emcli associate_user_profile
      -name="profile1"
      -users=user1;user2
```

awrwh_add_src_db

Adds a database as a source to the AWR Warehouse.

Prerequisites:

- AWR Warehouse should be configured.

- The database being added cannot be:

  - An instance of a RAC database

  - The AWR Warehouse

- The database user should be a DBA.

- The database user should have execute privilege on SYS.DBMS_SWRF_INTERNAL.

Format

```
emcli awrwh_add_src_db
       -target_name=<target_database_name>
       -target_type=<target_database_type>
       -db_cred=<target_database_named_credential>
       -db_host_cred=<target_database_host_named_credential>
       [-retention_days=<number_of_retention_days>]
       [-upload_interval=<snapshot_upload_interval>]
       -src_dir=<directory_to_store_extracted_dump_files>
[ ] indicates that the parameter is optional
```

Options

- target_name

  Name of the target database added as a source to the AWR Warehouse.

- target_type

  Type of target. The possible values for target type are `oracle_database`, `oracle_pdb`, and `rac_database`.

- db_cred

  Named credential for the target database.

- db_host_cred

  Named credential for the target database host.

- retention_days

  Number of days the AWR data will be retained in warehouse. The default value is one year.

- upload_interval

  Snapshot upload interval for the source database in AWR Warehouse. The default value is 24 hours.

- src_dir

  Directory to store extracted dump files. This is an optional parameter except for PDBs on cluster and by default the default agent directory is used.

Output

Success/Error message

Example

The following example adds the target database `sample_database` as a source to the AWR Warehouse:

```
emcli awrwh_add_src_db
        -target_name=sample_database
        -target_type=oracle_database
        -db_cred=DB1_NAMED_CRED
        -db_host_cred=TEST_HOST_CRED
```

awrwh_list_src_dbs

Lists all the source databases uploading to the AWR Warehouse. Enterprise Manager Super Administrators can view all databases uploading to the AWR Warehouse. Enterprise Manager Administrators can only view databases added by themselves or databases on which they have view privileges.

Prerequisite: AWR Warehouse should be configured.

Format

```
emcli awrwh_list_src_dbs
        [-all | -dbs_with_no_recent_uploads]
        [-script | -format=[name:<pretty|script|csv>];

[column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];]
        [-noheader]
[ ] indicates that the parameter is optional
```

Options

- all

  Sentence fragment describing the option.

Chapter 5
EM CLI Verbs

- dbs_with_no_recent_uploads

  Lists all source databases with no recent uploads.

- script

  Option equivalent to -format='name: script'.

- format

  Format specification. The default value is -format="name: pretty". Use the following commands for alternate formats:

  – -format="name:pretty" prints the output table in a readable format not intended to be parsed by scripts.

  – -format="name:script" sets the default column separator to a tab and the default row separator to a newline.

  – -format="name:csv" sets the column separator to a comma and the row separator to a newline.

- noheader

  If this option is specified, then column headers are not printed.

Output

Lists the databases uploading to AWR Warehouse.

Examples

Example 1

The following example lists all the source databases uploading to the AWR Warehouse:

```
emcli awrwh_list_src_dbs
```

Example 2

The following example lists all the source databases uploading to the AWR Warehouse without the column headers:

```
emcli awrwh_list_src_dbs
            -noheader
```

awrwh_reconfigure

Reconfigures AWR Warehouse database parameters.

Prerequisite: AWR Warehouse should be configured.

Format

```
emcli awrwh_reconfigure
            [-upload_interval=<upload_interval>]
            [-retention=<retention_period>]
            [-dest_dir=<dump_location>]
            [-db_cred=<target_database_named_credential>]
            [-db_host_cred=<target_database_host_named_credential>]
            [-schema_password=<password_for_the_staging_schema>]
            [-list]
[ ] indicates that the parameter is optional
```

Options

- upload_interval

  New snapshot upload interval in hours.

- retention

  New retention period in years. If set to 0, AWR snapshots are retained permanently.

- dest_dir

  New dump location for storing AWR snapshots. In the case of RAC databases, provide a common location accessible to all instances.

- db_cred

  New named credential for the target database.

- db_host_cred

  New named credential for the target database host.

- schema_password

  If the selected warehouse database has a password policy, then specify a policy compliant password for the staging schema.

- list

  Lists current values of the parameters given above. This is the default when the other options are not specified.

Output

Success/Error

Examples

Example 1

The following example reconfigures the upload interval and dump location parameters:

```
emcli awrwh_reconfigure
           -upload_interval=12
           -dest_dir="foo/bar"
```

Example 2

The following example reconfigures the database credential and database host credential parameters:

```
emcli awrwh_reconfigure
           -db_cred=DB_CRED_NAME
           -db_host_cred=DB_HOST_CRED_NAME
```

awrwh_reconfigure_src

Reconfigures AWR Warehouse source database parameters.

Prerequisites:

- AWR Warehouse should be configured.

- Database specified should be uploading to the AWR Warehouse.

Format

```
emcli awrwh_reconfigure_src
           -target_name=<target_database_name>
           -target_type=<target_database_type>
           [-inst_num=<RAC_instance_number>]
           [-src_dir=<dump_location>]
           [-upload_interval_hrs=<upload_interval>]
           [-retention_days=<retention_period>]
           [-db_cred=<database_named_credential>]
           [-host_cred=<database_host_named_credential>]
[ ] indicates that the parameter is optional
```

Options

- target_name

  Name of existing target database.

- target_type

  Type of target. The possible values for target type are `oracle_database`, `oracle_pdb`, and `rac_database`.

- inst_num

  Instance number in case of a RAC database.

- src_dir

  New dump location for storing AWR snapshots. In the case of RAC databases, either specify an instance using the inst_num parameter or provide a common location accessible to all instances.

- upload_interval_hrs

  New snapshot upload interval in hours.

- retention_days

  New retention period in days. If set to 0, AWR snapshots are retained permanently.

- db_cred

  New named credential for the target database.

- host_cred

  New named credential for the target database host.

Output

Success/Error

Examples

Example 1

The following example reconfigures the AWR Warehouse source database to switch to a new dump location:

```
emcli awrwh_reconfigure_src
           -target_name=database
```

```
            -target_type=oracle_database
            -src_dir="foo/bar"
```

Example 2

The following example changes the retention period of the source database to 30 days:

```
emcli awrwh_reconfigure_src
            -target_name=database
            -target_type=oracle_database
            -retention_days=30
```

Example 3

The following example changes the snapshot upload interval of the source database to 15 hours and the host credential to MY_HOST_CRED:

```
emcli awrwh_reconfigure_src
            -target_name=database
            -target_type=oracle_database
            -upload_interval_hrs=15
            -host_cred=MY_HOST_CRED
```

awrwh_remove_src_db

Removes an existing database from the AWR Warehouse.

Prerequisites:

- AWR Warehouse should be configured.

- Database specified should be uploading to the AWR Warehouse.

Format

```
emcli awrwh_remove_src_db
            -target_name=<target_database_name>
            -target_type=<target_database_type>
            [-retain_data_in_warehouse]
[ ] indicates that the parameter is optional
```

Options

- target_name

  Name of existing target database.

- target_type

  Type of target. The possible values for target type are oracle_database, oracle_pdb, and rac_database.

- retain_data_in_warehouse

  If this flag is added, then the already collected AWR data is retained in the warehouse but all upload activity is stopped.

Output

Success/Error

Example

The following example removes the target database `sample_database` from the AWR Warehouse:

```
emcli awrwh_remove_src_db
          -target_name=sample_database
          -target_type=oracle_database
          -retain_data_in_warehouse
```

backup_database

Schedules a backup for database targets. This verb has multiple subcommands that perform different types of database backups. Some options are common across multiple subcommands, while other options are specific to a subcommand. The available subcommands are as follows:

- customBackup: Schedules a custom backup, with user-specified scope, level, and frequency.

- suggestedBackup: Schedules a backup using the Oracle-suggested strategy appropriate to the local backup configuration.

backup_database -customBackup

Schedules a custom Database Backup deployment procedure for one or more database targets. Each backup uses the database target specific RMAN configuration.

Format

```
emcli backup_database -customBackup
    ((-target_name="<database target name>" -target_type="oracle_database|rac_database|
oracle_cloud_dbcs")
    | -input_file="target_list:<full path name of input file>")
    [-db_cred="<database named credential>"]
    [-db_host_cred="<database host named credential>"]
    [-scope="wholeDB|tablespace|datafile|archivedLog|recoveryFilesOnDisk|cdbRoot|
pluggableDbs"]
    [-scope_value="comma-separated list of values"]
    [-backup_type="full|incr0|incr1" [-incr_type="diff|cumulative"]]
    [-destination_media="disk|tape|ra|cloud"]
    [-skip_backup_archived_logs]
    [-delete_archived_logs_after_backup]
    [-delete_obsolete]
    [-max_files_per_set="n"]
    [-max_corruptions="n"]
    [-section_size="n"]
    [-procedure_name="<deployment procedure name>"]
    [-storage_container="<backup storage container name>"]
    [[-rman_encryption="wallet|password|both"]
     [-rman_encryption_cred="<RMAN encryption named credential>"]
     [-encryption_algorithm="AES128|AES192|AES256"]]
    [-schedule=
        {
        start_time:yyyy/MM/dd HH:mm;
        tz:{java timezone ID};
        frequency:interval/weekly/monthly/yearly;
        repeat:#m|#h|#d|#w;
        months:#,#,...;
        days:#,#,...;
```

```
 end_time:yyyy/MM/dd HH:mm;
 grace_period:;
 }]
```

Options

- target_name

  A single-instance, cluster database, or database cloud service target name for the database that will be backed up. (A multi-database backup can be performed by using the -input_file option instead of -target_name/-target_type.)

- target_type

  The type of the target specified by -target_name, either a single-instance database (oracle_database), a cluster database (rac_database), or a database cloud service (oracle_cloud_dbcs).

- input_file

  The name of a file containing the information for each database that will be backed up. This option is an alternative to -target_name/-target_type. The format of this file is as follows:

```
target.0.target_name=<database #1 target name>
target.0.target_type=oracle_database|rac_database|oracle_cloud_dbcs
target.0.db_cred=<database named credential for database #1>
target.0.db_host_cred=<database host named credential for database #1>
target.1.target_name=<database #1 target name>
target.1.target_type=oracle_database|rac_database|oracle_cloud_dbcs
target.1.db_cred=<database named credential for database #2>
target.1.db_host_cred=<database host named credential for database #2>
```

  The target_name and target_type lines are required for each database. The db_cred and db_host_cred lines are optional; if present for a database, they override the command-line named credential settings (if any) for that database.

- db_cred

  The name of an existing Enterprise Manager database named credential that can be used to connect to all the specified target databases. If the operating system user specified by the -db_host_cred argument is a member of the operating system DBA group, the database credentials can specify a Normal role user. Otherwise, the database credentials must specify a SYSDBA role user. If this argument is not specified, then preferred credentials will be used. If multiple databases are specified, this should be a global named credential. If the -input_file option is used, this setting can be optionally overridden for an individual database as noted above.

- db_host_cred

  The name of an existing Enterprise Manager database host named credential that can be used to run the RMAN command on the specified target database hosts. If this argument is not specified, then preferred credentials are used. If multiple databases are specified, this should be a global named credential. If the -input_file option is used, this setting can be optionally overridden for an individual database as noted above.

- scope

  The scope of the backup. If this argument is not specified, the default value is wholeDB. The possible values are:

- wholeDB: All data files in the database.

- tablespace: One or more data files associated with the tablespace name.

- datafile: One or more data files.

- archivedLog: Exact copy of each distinct log sequence number.

- recoveryFilesOnDisk: All recovery files on disk, whether they stored in the fast recovery area or other locations on disk.

- cdbRoot: The data files in the whole container root.

- pluggableDbs: The data files in one or more pluggable databases.

- scope_value

  A comma-separated list of values to back up. It is required when the -scope value is `tablespace|datafile|pluggableDbs`.

- backup_type

  The type of backup. If this argument is not specified, the default value is full. The possible values are:

  - full: Back up all data blocks in the data files being backed up.

  - incr0: Incremental level 0. This is similar to a full backup, but can be used as the base of an incremental backup strategy.

  - incr1: Incremental level 1. This back up is only for the changed blocks.

- incr_type

  The type of incremental backup. This is used in combination with `-backup_type="incr"`. If this argument is not specified, the default is diff. The possible values are:

  - diff: Back up blocks updated since the most recent level 0 or level 1 incremental backup.

  - cumulative: Back up all blocks changed since the most recent level 0 backup.

- destination_media

  The destination media for this backup. If this argument is not specified, the default value is disk. The possible values are:

  - disk: Backup to disk (not allowed when `-scope="recoveryFilesOnDisk"`).

  - tape: Backup to a SBT (system backup to tape) device.

  - ra: Backup to a Recovery Appliance.

  - cloud: Backup to Database Backup Cloud Service.

- rman_encryption

  Specify to encrypt the backup using the Oracle Encryption Wallet, a user-supplied password, or both. The possible values are:

  - wallet: Oracle Encryption Wallet

  - password: user-supplied password

  - both: Oracle Encryption Wallet and user-supplied password

  This option can be specified when `-scope="wholeDB|tablespace|datafile|archivedLog|recoveryFilesOnDisk|cdbRoot|pluggableDbs"`.

- rman_encryption_cred

The name of an existing Enterprise Manager generic named credential that contains the encryption password. This is required when:

–   -rman_encryption="password" is specified

–   -rman_encryption_algorithm

The name of the encryption algorithm to use while encrypting backups. This is used in combination with -rman_encryption. The possible values are AES128, AES192, and AES256. If this argument is not specified, the default value is AES256.

*   storage_container

    The name of the Oracle Storage Service container where this backup will be stored. If the container does not already exist it will be created. This argument overrides the container obtained from the Hybrid Cloud Setup account information or the Database Backup Cloud Service Settings (whichever is applicable) when the initial backup configuration was performed.

    This option is applicable only when -destination_media="cloud".

*   skip_backup_archived_logs

    Do not back up all archived logs on disk that have not been backed up.If this argument is not specified, the default behavior is to back up all archived logs that have not been backed up.

    This option can be specified when -scope="wholeDb|tablespace|datafile| cdbRoot|pluggableDbs".

*   delete_obsolete

    Delete backups that are no longer required to satisfy the retention policy. This option can be specified when -scope="wholeDB|tablespace|datafile|cdbRoot| pluggableDbs".

*   max_files_per_set

    The maximum number of files to include in each backup set. This option can be specified when -scope="wholeDB|tablespace|datafile|archivedLog|cdbRoot| pluggableDbs".

*   max_corruptions

    The maximum number of physical corruptions allowed in data files. This option can be specified when -scope="datafile".

*   section_size

    The size in MB of each backup section produced during a data file backup, resulting in a multi-section backup where each backup piece contains one file section. This option cannot be used if the Maximum Backup Piece Size is set in the RMAN configuration. This option can be specified when -scope="wholeDB| tablespace|datafile|archivedLog|cdbRoot|pluggableDbs".

*   procedure_name

    The name of the Database Backup deployment procedure. At procedure execution time, the name will be modified to include a timestamp.

*   delete_archived_logs_after_backup

Delete all archived logs from disk after they are successfully backed up. Ignored if -*skip_backup_archived_logs* is specified. This option can be specified when -scope="wholeDB|tablespace|datafile|cdbRoot|pluggableDbs".

- schedule

  Schedules the customized backup deployment procedure. If schedule option is not provided, the procedure runs immediately.

  – start_time: Time when the procedure has to start execution. The format should be "yyyy/MM/dd HH:mm".

  – tz: The timezone ID (optional).

  – frequency: Valid values are once/interval/weekly/monthly/yearly (optional).

    If frequency is set to interval then repeat has to be specified.

    If frequency is set to weekly or monthly, days has to be specified.

    If frequency is set to yearly, both days and months have to be specified.

  – repeat: Frequency with which the procedure has to be repeated. This is required only if the frequency is set to interval.

  – days: Comma separated list of days. This is required only if frequency is weekly, monthly, or yearly.

  – months: Comma separated list of months. This is required only if the frequency is yearly. The valid range is 1 through 12.

  – end_time: End time for procedure executions. If it is not specified, procedure will run indefinitely (optional). The format should be "yyyy/MM/dd HH:mm".

  – grace_period: Grace period in minutes (optional).

Example 1

The following example performs an incremental whole-database backup to Oracle Cloud with password encryption. It also schedules the backup for a later time.

```
emcli backup_database -customBackup -scope="wholeDB"
-target_name="db1" -target_type="oracle_database"
-backup_type="incr0" -destination_media="cloud"
-storage_container="dbContainer1" -rman_encryption="password"
encryption_algorithm="AES128"
-rman_encryption_cred="NC1" -schedule="start_time:2016/11/08 10:08;tz:PST;
```

Example 2

The following example backups a Database Cloud Service target to cloud. It does not include archived logs in the backup.

```
emcli backup_database -customBackup -target_name="SALES-DBCS"
-target_type="oracle_cloud_dbcs" -scope="wholeDB"
-backup_type="full" -destination_media="cloud"
-skip_backup_archived_logs -db_cred="SYS_DB_CRED" -db_host_cred="NZHAO_HOST_CRED"
```

Example 3

The following example performs a differential incremental backup for pluggable databases CDB1_PDB1 and PDB1 to disk. It also deletes the archived logs and obsolete backups after the backup.

```
emcli backup_database -customBackup -target_name="db1"
-target_type="oracle_database" -scope="pluggableDbs"
-scope_value="CDB1_PDB1,PDB1" -backup_type="incr1"
-incr_type="diff" -delete_archived_logs_after_backup -delete_obsolete
-destination_media="disk" -db_cred="DB_SYS_CRED"
-db_host_cred="OMS_HOST_CRED"
```

Example 4

The following example performs a full backup of the databases specified in the target list of the input_file to disk.

```
 emcli backup_database -customBackup -input_file="target_list:rcap.prop" -
backup_type="full" -destination_media="disk"
```

where the content in `rcap.prop` is

```
rcap.prop
target.0.target_name=rcap
target.0.target_type=oracle_database
target.0.db_cred=DB_CREDS
target.0.db_host_cred=HOST_CREDS
target.1.target_name=rcap2
target.1.target_type=oracle_database
target.1.db_cred=DB_CREDS
target.1.db_host_cred=HOST_CREDS
```

backup_database -suggestedBackup

Schedules a backup using the Oracle suggested strategy appropriate to the specified backup destination. For disk, the standard incremental-update and roll forward strategy is used. For Recovery Appliance, an incremental-forever strategy is used.

Format

```
emcli backup_database -suggestedBackup
    ((-target_name="<database target name>"
    -target_type="oracle_database|rac_database|oracle_cloud_dbcs")
    | -input_file="target_list:<full path name of input file>")
    [-db_cred="<database named credential>"]
    [-db_host_cred="<database host named credential>"]
    [-destination_media="disk|ra"]
    [-skip_backup_archived_logs]
    [-delete_archived_logs_after_backup]
    [[-rman_encryption="wallet|password|both"]
     [-rman_encryption_cred="<RMAN encryption named credential>"]
     [-encryption_algorithm="AES128|AES192|AES256"]]
    [-schedule=
        {
         start_time:yyyy/MM/dd HH:mm;
         tz:{java timezone ID};
         frequency:interval/weekly/monthly/yearly;
         repeat:#m|#h|#d|#w;
         months:#,#,...;
         days:#,#,...;
         end_time:yyyy/MM/dd HH:mm;
         }]
```

Options

• target_name

A single-instance, cluster database, or database cloud service target name for the database that will be backed up. (A multi-database backup can be performed by using the -input_file option instead of -target_name/-target_type.)

- target_type

  The type of the target specified by -target_name, either a single-instance database (oracle_database), a cluster database (rac_database), or a database cloud service (oracle_cloud_dbcs).

- input_file

  The name of a file containing the information for each database that will be backed up. This option is an alternative to -target_name/-target_type. The format of this file is as follows:

  ```
  target.0.target_name=<database #1 target name>
  target.0.target_type=oracle_database|rac_database|oracle_cloud_dbcs
  target.0.db_cred=<database named credential for database #1>
  target.0.db_host_cred=<database host named credential for database #1>
  target.1.target_name=<database #1 target name>
  target.1.target_type=oracle_database|rac_database|oracle_cloud_dbcs
  target.1.db_cred=<database named credential for database #2>
  target.1.db_host_cred=<database host named credential for database #2>
  ```

- db_cred

  The name of an existing Enterprise Manager database named credential that can be used to connect to all the specified target databases. If the operating system user specified by the -db_host_cred argument is a member of the operating system DBA group, the database credentials can specify a Normal role user. Otherwise, the database credentials must specify a SYSDBA role user. If this argument is not specified, then preferred credentials will be used. If multiple databases are specified, this should be a global named credential. If the -input_file option is used, this setting can be optionally overridden for an individual database as noted above.

- db_host_cred

  The name of an existing Enterprise Manager database host named credential that can be used to run the RMAN command on the specified target database hosts. If this argument is not specified, then preferred credentials will be used. If multiple databases are specified, this should be a global named credential. If the -input_file option is used, this setting can be optionally overridden for an individual database as noted above.

- destination_media

  The destination media for this backup. If this argument is not specified, the default value is disk. The possible values are:

  – disk: Backup to disk.

  – ra: Backup to a Recovery Appliance.

- rman_encryption

  Specifies to encrypt the backup using the Oracle Encryption Wallet, a user-supplied password, or both. The possible values are:

  – wallet: Oracle Encryption Wallet

  – password: user-supplied password

  – both: Oracle Encryption Wallet and user-supplied password

- rman_encryption_cred

The name of an existing Enterprise Manager generic named credential that contains the encryption password. This is required when -`rman_encryption=\"password\"` is specified.

*   rman_encryption_algorithm

    The name of the encryption algorithm to use when encrypting backups. Used in combination with -rman_encryption. The possible values are AES128, AES192, and AES256. If this argument is not specified, the default value is AES256.

*   skip_backup_archived_logs

    This option does not back up all the archived logs on disk that have not been backed up. If this argument is not specified, the default behavior is to back up all the archived logs that have not been backed up. This option should be provided if the database is configured to ship redo to the Recovery Appliance. This option can only be specified when -`destination_media=\"ra\"`.

*   delete_archived_logs_after_backup

    Deletes all the archived logs from disk after they are successfully backed up. This task is ignored if -skip_backup_archived_logs is specified. This option can only be specified when -`destination_media=\"ra\"`.

Example 1

The following example schedules a daily backup to a Recovery Appliance using the Oracle suggested strategy. This does not include archived logs in the backup.

```
emcli backup_database -suggestedBackup -target_name="prod-db" "-
target_type=" "oracle_database"
   -destination_media="ra"
   -skip_backup_archived_logs
   -db_cred="DB_SYS_CRED"
   -db_host_cred="OMS_HOST_CRED"
   -schedule="start_time:2016/12/19 16:00;tz:PST;frequency:interval;repeat:1d"
```

Example 2

The following example schedules a daily backup to disk using the Oracle suggested strategy.

```
emcli backup_database -suggestedBackup -target_name="prod-db"
   -target_type="oracle_database" -destination_media="disk"
   -db_cred="DB_SYS_CRED" -db_host_cred="OMS_HOST_CRED"
   schedule="start_time:2016/12/19 23:55;tz:PST;frequency:interval;repeat:1d"
```

bareMetalProvisioning

Assigns a test-type to a target-type. If a test-type t is assigned to target-type T, all targets of type T can be queried with tests of type t.

Format

```
emcli bareMetalProvisioning
      [-input_file="config_properties:input_XML"]

[ ]  indicates that the parameter is optional
```

Options

- **input_file**

  Input XML file confirming to the XSD for bare metal provisioning. See below for a detailed XML file used to provision BMP.

  For more information about the input_file option, see -input_file Syntax Guidelines.

Example

```
emcli bareMetalProvisioning
      [-input_file="config_properties:input XML"]
```

XML Example File

```
<?xml version="1.0"?>
<DeployedImage><ImageName>OracleLinux</ImageName><OperatingSystemType>Oracle
Enterprise Linux x86 32 bit</OperatingSystemType>

<!--Specify the Operating system type for this operation. Supported operating system
types are : Oracle Enterprise Linux x86 32 bit, Oracle Enterprise Linux x86 64 bit,
RedHat Enterprise Linux x86 32 bit, RedHat Enterprise Linux x86 64 bit, SUSE Linux
Enterprise Server x86 32 bit, SUSE Linux Enterprise Server x86 64 bit-->
<TargetInfo><TargetType>MAC</TargetType>

<!--Specify the target type for this provisioning operation as follows : MAC : If the
provisioning target type is mac address. RE_IMAGE : If reprovisioning the existing EM
targets. Subnet : If provisioning all the targets in a Subnet.-->
<Target><MACAddress>aa:bb:cc:dd:ee:ff</
MACAddress><NetworkInterface><InterfaceName>eth0</InterfaceName><Configuration>Dhcp</
Configuration>

<!--Specify the network interface type as follows : Dhcp : If the interface
configuration is to be dynamically assigned from a DHCP server. Static : If the
interface configuration is to be statically configured. Network Profile : If the
interface configuration is to be fetched from a Network Profile.-->
<Type>Non Bonding</Type>

<!--Specify the network interface type as follows : Non Bonding : If the interface is
not part of any bond. Bonding Master : If the interface is supposed to be the Bonding
master of a bond. Bonding Slave : If the interface is supposed to be a Bonding slave
as part of bond.-->
<!-- Following are bonding configuration-->
<BondingMode>1</BondingMode>

<!--Specify the Bonding Mode in case the interface has the role of Bonding Master.-->
<SlaveDevices>eth1,eth2</SlaveDevices>

<!--Specify the Slave devices as a csv string in case the interface has the role of
Bonding Master. For ex : eth1,eth2-->
<PrimarySlaveDevice>eth1</PrimarySlaveDevice>

<!--Specify the Primary Slave device in case the interface has the role of Bonding
Master.-->
<ARPInterval>200</ARPInterval><ARPIPTarget>10.177.244.121</
ARPIPTarget><ARPFrequency>400</ARPFrequency><ARPDownDelay>200</
ARPDownDelay><ARPUpDelay>200</ARPUpDelay>

<!--bonding configuration-->
<!-- If Configuration is Static <IPAddress></IPAddress> <HostName></HostName>
```

```
<Netmask></Netmask> <Gateway></Gateway> <DNSServers></DNSServers> -->
<IsBootable>true</IsBootable>

<!--Specify if the network interface is the bootable one.-->
</NetworkInterface></Target>

<!-- If TargetType is RE_IMAGE: replace <MACAddress> with following
<HostName>myhost.us.example.com</HostName> <BootableMac>aa:bb:cc:dd:ee:ff</
BootableMac> -->
<!-- If TargetType is Subnet: replace <MACAddress> with following
<SubnetIP>10.244.177.252</SubnetIP> <SubnetMask>255.255.252.0</SubnetMask> -->
</TargetInfo>

<!-- If TargetType is RE_IMAGE:
<HostTargetsPreferredCredSetName>cred1<.HostTargetsPreferredCredSetName> Specify
the preferred credentials name to be used for accessing the EM Host targetse to
be Re-imaged. -->
<StageServer>myhost.us.example.com</StageServer>

<!--Specify the Stage Server host name. For ex : myhost.mydomain.com-->
<StageStorage>/scratch/stage</StageStorage>

<!--Specify the Stage Storage on the stage server. For ex : /scratch/stage-->
<StageServerPreferredCredSetName>Cred1</StageServerPreferredCredSetName>

<!--Specify the preferred credentials name to be used for accessing the stage
server.-->
<StageServerPrereqs>false</StageServerPrereqs>

<!--Specify if the pre-requisties check should be run on the stage server before
starting the provisioning.-->
<BootServer>myhost.us.example.com</BootServer>

<!--Specify the Boot Server host name. For ex : myhost.mydomain.com-->
<BootServerPreferredCredSetName>Cred1</BootServerPreferredCredSetName>

<!--Specify the preferred credentials name to be used for accessing the boot
server.-->
<BootServerPrereqs>false</BootServerPrereqs>

<!--Specify if the pre-requisties check should be run on the boot server before
starting the provisioning.-->
<DhcpServer>myhost.us.example.com</DhcpServer>

<!--Specify the Dhcp Server host name. This is required only if DHCP automation
is required. For ex : myhost.mydomain.com Note : Dhcp automation is supported
only for the MAC and RE_IMAGE provisioning types.-->
<DhcpServerPreferredCredSetName>Cred1</DhcpServerPreferredCredSetName>

<!--Specify the preferred credentials name to be used for accessing the Dhcp
server.-->
<DhcpServerPrereqs>false</DhcpServerPrereqs>

<!--Specify if the pre-requisties check should be run on the Dhcp server before
starting the provisioning.-->
<RpmRepository>oelrepos</RpmRepository>

<!--Specify the RPM repository name to be used for provisioning operation. For
ex : OEL4U8repos-->
<RootPassword>password</RootPassword>
```

```
<!--Specify the root password for the provisioned machines.-->
<TimeZone>Africa/Algiers</TimeZone>

<!--Specify the time zone for the provisioned machines.-->
<AgentInstallUser>oraem</AgentInstallUser>

<!--Specify the user name for installing EM agent on the provisioned machines. For
ex : oraem-->
<AgentInstallGroup>dba</AgentInstallGroup>

<!--Specify the agent installation user's group for installing EM agent on the
provisioned machines. For ex : dba-->
<AgentBaseInstallationDirectory>/var/lib/oracle/agent12g</
AgentBaseInstallationDirectory>

<!--Specify a directory for installing EM agent on the provisioned machines. For
ex : /var/lib/oracle/agent12g-->
<OracleInventoryLocation>/var/lib/oracle/oraInventory</OracleInventoryLocation>

<!--Specify a directory for storing oracle installed product's inventory on the
provisioned machines. For ex : /var/lib/oracle/oraInventory-->
<AgentRegistrationPassword>password</AgentRegistrationPassword>

<!--Specify agent registration password for installing EM agent on the provisioned
machines.-->
<AgentRpmUrl>http://myhost.us.example.com/oracle-agt.12.1.0.0.1-i386.rpm</AgentRpmUrl>

<!--Specify a http URL for fetching agent RPM. This is not mandatory if the agent rom
is already placed at the staged location-->
<ReferenceAnaconda/>

<!--Specify a reference anaconda as a string. It will be used to capture properties
like Keyboard, mouse. If not provided they will be defaulted to default values.-->
<PackageList>@base</PackageList>

<!--Specify the package list to be installed on the provisioned machines.-->
<ACPI>off</ACPI>

<!--Specify the ACPI value for the provisioned machines. Supported values are : on,
off-->
<ParaVirtualizedKernel>false</ParaVirtualizedKernel>

<!--Specify if the provisioned machines should be booted with paravirtualized
kernels.-->
<PostInstallScript>%post echo "post" </PostInstallScript>

<!--This section provides the option of adding commands to be run on the system once
the installation is complete. This section must start with the %post command.-->
<FirstBootScript>#!/bin/sh # chkconfig: 345 75 25 # description: Bare Metal
Provisioning First boot service # </FirstBootScript>

<!--This section provides the option of adding commands to run on the system when it
boots for the first time after installation.-->
<RequireTTY>false</RequireTTY>

<!--Specify if tty is required on the provisioned machines.-->
<SeLinux>Disabled</SeLinux>

<!--Specify the SELinux configuration for the provisioned machines. Supported values
are : Disabled, Enforcing, Permissive-->
<MountPointSettings/>
```

```
<!--Specify /etc/fstab settings for the provisioned machines.-->
<NISSettings/>

<!--Specify /etc/yp.conf settings for the provisioned machines.-->
<NTPSettings/>

<!--Specify /etc/ntp.conf settings for the provisioned machines.-->
<KernelParameterSettings/>

<!--Specify /etc/inittab settings for the provisioned machines.-->
<FirewallSettings/>

<!--Specify the firewall settings for the provisioned machines.-->
<HardDiskProfiles>

<!--Specify the Hard Disk profiles for the provisioned machines.-->
<HardDiskConfiguration>

<!--Specify the hard disk configuration details-->
<DeviceName>hda</DeviceName>

<!--Specify the device name for the disk. For ex : hda,hdb-->
<Capacity>10000000</Capacity>

<!--Specify the disk capacity in MB. For ex : 1024-->
</HardDiskConfiguration></HardDiskProfiles><PartitionConfigurations>

<!--Specify the partition configurations for the provisioned machines.-->
<PartitionConfiguration>

<!--Specify the partition configuration details.-->
<MountPoint>/</MountPoint>

<!--Specify the mount point for the partition. For ex : /, /root-->
<DeviceName>hda</DeviceName>

<!--Specify the disk name on which this partition has to be configured. For ex :
hda,hdb-->
<SystemDeviceName>/dev/hda1</SystemDeviceName>

<!--For ex : /dev/hda1-->
<FileSystemType>ext3</FileSystemType>

<!--Specify the File System type for this partition. Supported file system types
are : ext2, ext3, ocfs2, swap, Raid, LVM-->
<Size>4096</Size>

<!--Specify the size in MB for this partition. For ex : 5120-->
</PartitionConfiguration></PartitionConfigurations><RaidConfigurations>

<!--Specify the RAID configurations for the provisioned machines.-->
<RaidConfiguration>

<!--Specify the RAID configuration details-->
<MountPoint>raid.100</MountPoint>

<!--Specify the raid id . For ex : raid.100-->
<RaidLevel>0</RaidLevel>

<!--Specify the RAID Level for this raid device. Supported RAID Levels are :
```

```
Raid 0, Raid 1, Raid 5, Raid 6-->
<Partitions>/dev/hda1,/dev/hda2</Partitions>

<!--Specify the raid partitions for this raid device as a csv string. For ex : /dev/
hda1, /dev/hda2-->
<FileSystemType>ext3</FileSystemType>

<!--Specify the File System type for this partition. Supported file system types are :
ext2, ext3, ocfs2, swap, LVM-->
</RaidConfiguration></RaidConfigurations><LogicalVolumeGroups>

<!--Specify the Logical Volume Groups for the provisioned machines.-->
<LogicalVolumeGroup>

<!--Specify the logical volume group configuration details-->
<GroupName>LVG1</GroupName>

<!--Specify the Logical group name. For ex : mygrp-->
<Partitions>/dev/hda1</Partitions>

<!--Specify the partitions that take part in this logical volume group as a csv
string. For ex : /dev/hda1, /dev/hda2-->
<Raids>raid.100</Raids>

<!--Specify the RAIDs that take part in this logical volume group as a csv string. For
ex : raid.100, raid.200-->
</LogicalVolumeGroup></LogicalVolumeGroups><LogicalVolumes>

<!--Specify the Logical Volumes for the provisioned machines.-->
<LogicalVolume>

<!--Specify the logical volume configuration details.-->
<MountPoint>/u01</MountPoint>

<!--Specify the mount point for this logical volume. For ex : /, /root-->
<LogicalVolumeName>LV1</LogicalVolumeName>

<!--Specify the logial volume name. For ex : myvols-->
<LogicalGroupName>LVG1</LogicalGroupName>

<!--Specify the logical group name where this volume should be created. For ex :
mygrp-->
<FileSystemType>ext3</FileSystemType>

<!--Specify the File System type for this partition. Supported file system types are :
ext2, ext3, ocfs2, swap-->
<Size>4096</Size>

<!--Specify the size in MB for this partition. For ex : 5120-->
</LogicalVolume></LogicalVolumes></DeployedImage>
```

cancel_cloud_service_requests

Cancels scheduled cloud service request(s) initiated by the specified user. Note that only
scheduled requests can be cancelled.

Format

```
emcli cancel_cloud_service_requests
      -user="username"
```

```
            [-family="family"]
            [-ids="id1;id2..."]
```

```
[ ]  indicates that the parameter is optional
```

Options

* **user**

  Name of the user who initiated the requests.

* **family**

  Service family name to use to filter cloud requests.

* **ids**

  List of Request IDs to use to filter cloud requests. Separate each ID with a
  semicolon (;).

Examples

Example 1

This example cancels all scheduled cloud requests owned by user1.

```
emcli cancel_cloud_service_requests
      -user="user1"
```

Example 2

This example cancels all cloud requests owned by user1 and belonging to the family1
service family.

```
emcli cancel_cloud_service_requests
      -user="user1"
      -family="family1"
```

change_ruleset_owner

Assigns the ownership of an enterprise rule set to a new owner.

Privilege Requirements:

The new owner must have the following privileges:

* *Edit Enterprise Rule set* on the rule set.

* If specific targets are selected in the rule set, the new user must have *View Target*
  privilege on those targets.

* If some of the rules either creates or updates incidents, the new user must have
  Manage Target Event privilege on those targets as well.

Format

```
emcli change_ruleset_owner
      -rule_set_name='rule set name'
      -rule_set_owner=<rule set owner>
      -new owner=<new owner>
```

Options

- rule_set_name

  Name of an enterprise rule set. This option only applies to rule sets that are associated with a list of targets.

- rule_set_owner

  Current owner of the rule set.

- new_owner

  New owner of the rule set.

Example

The following example removes USER1 as the owner of rule set "My rule set" and assigns USER2 as the new owner.

```
emcli change_ruleset_owner -rule_set_name 'My rule set' -rule_set_owner USER1 -
new_owner USER2
```

change_service_system_assoc

Changes the system that hosts a given service.

Format

```
emcli change_service_system_assoc
      -name='name'
      -type='type'
      -systemname='system_name'
      -systemtype='system_type'
      -keycomponents='keycomp1name:keycomp1type[;keycomp2name:keycomp2type;...]'

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Service name.

- **type**

  Service type.

- **systemname**

  System on which the service resides.

- **systemtype**

  System type.

- **keycomponents**

  Name-type pair (such as `keycomp_name:keycomp_type`) list of key components in the system used for the service.

Example

This example changes the system for a generic service named `my service` to a
generic system named `my system` with specified key components.

```
emcli change_service_system_assoc
      -name='my service' -type='generic_service'
      -systemname='my system' -systemtype='generic_system'
      -keycomponents='database:oracle_database; mytestbeacon:oracle_beacon'
```

change_target_owner

Changes the owner of the target.

Format

```
emcli change_target_owner
        -target="target_name:target_type"
        [-target="target_name:target_type"]
        -owner="current_target_owner_name"
        -new_owner="new_owner_name"
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **target**

    Target name and target type to change the owner.

*   **owner**

    Name of the existing owner of the target. The default value for this option is the
    currently logged in user.

*   **new_owner**

    New owner name of the target.

Example

This example changes the ownership of two targets from admin to admin2.

```
emcli change_target_owner
        -target="abc.oracle.com:host"
        -target="testDBSystem:oracle_database"
        -owner="admin1"
        -new_owner="admin2"
```

clean_down_members

Removes all members of a Coherence cluster whose status is down including the
node and cache targets.

Format

```
emcli clean_down_members
      -coherence_cluster_name:<Coherence Cluster Target Name>      [-debug]
```

Options

- coherence_cluster_name

  Fully qualified name of the Coherence cluster target.

- debug

  Runs the verb in verbose mode for debugging purposes.

Example

This example deletes all members whose status is down in the `TestCluster` target.

```
emcli clean_down_members
    -coherence_cluster_name:"TestCluster"
```

cleanup_dbaas_requests

Cleans up requests from the host and Enterprise Manager. Depending on the options specified, this verb:

- Cleans up all failed requests from a pool.

- Cleans up all requests from a pool.

- Cleans up failed requests for a specific user.

- Cleans up all requests for a specific user.

Format

```
emcli help cleanup_dbaas_requests
emcli cleanup_dbaas_requests        [-ids="<request id>"]          [-pool_name="<pool
name>" -pool_type="
                      <database|schema|pluggable_database>"
        [-user="<SSA user name>"]
        [-all]]

[ ]  indicates that the parameter is optional.
```

Options

- **ids**

  Request ID to be used for filtering Cloud requests, separated by semicolons(;).

- **pool_name**

  Name of the pool from which requests must be cleaned up.

- **pool_type**

  Type of pool. Enter one of the following values:

  – For database pools: database

  – For schema pools: schema

  – For pluggable database pools: pluggable_database

- **user**

  User name to be used for filtering requests for deletions.

- **all**

  If specified, cleans up all requests (successful and failed), cancels the requests that are in a scheduled state, and leaves the requests that are in progress as is. If this option is not specified, cleanup is performed on failed requests only.

  > **Note:**
  >
  > The ids and pool_name options cannot be used together.

  Examples

  Example 1

  This example performs a cleanup of all requests, both failed and successful. This process essentially resets the pool.

  ```
  emcli cleanup_dbaas_requests -pool_name="database_pool" -pool_type="database" -
  all
  ```

  Example 2

  This example performs a cleanup of all requests (failed and successful) for a specific user. This option is useful in cases where the user is no longer in the system and the administrator wants to clean up all of the service instances owned by this user.

  ```
  emcli cleanup_dbaas_requests -pool_name="database_pool" -pool_type="database" -
  all -user="SSA_USER"
  ```

cleanup_config_extension

Cleans up corrupted configuration extensions by removing all associations of a given Configuration extension with the given target.

Format

```
emcli cleanup_config_extension
-ce_name="<ce_name>"
-target_name="<target_name>"
-target_type="<target_type>"
```

Options

- `-ce_name`
  The name of the Configuration Extension to be cleaned up.

- `-target_name`
  Name of the target.

- `-target_type`
  Target type associated with the compliance standard rule.

Example

```
emcli cleanup_config_extension -ce_name="Custom STIG Configuration"
    -target_name="my_target" -target_type="oracle_database"
```

clear_compliance_ca

Clears all corrective actions for a specified compliance standard rule and target type.

Format

```
emcli clear_compliance_ca
    -rule_iname="<rule_internal_name>"
    -target_type="<target_type>"
```

Options

*   rule_iname

    Internal name of the compliance standard rule from which all corrective actions are being removed. The compliance standard rule internal names are available in the MGMT$CS_RULE_ATTRS view.

*   target_type

    Target type associated with the compliance standard rule.

Example

The following example clears all corrective actions associated with the my_rule compliance standard rule located on the host target.

```
emcli clear_compliance_ca
    -rule_iname="my_rule"
    -target_type="host"
```

clear_credential

Clears preferred or monitoring credentials for given users.

Format

```
emcli clear_credential
    -target_type="ttype"
    [-target_name="tname"]
    -credential_set="cred_set"
    [-user="user"]
    [-oracle_homes="home1;home2"]

[ ]  indicates that the parameter is optional
```

Options

*   **target_type**

    Type of target, which must be "host" if you specify the oracle_homes option.

*   **target_name**

Name of the target. Omit this option to clear enterprise-preferred credentials. The target name must be the host name if you specify the oracle_homes option.

• **credential_set**

Credential set affected.

• **user**

Enterprise Manager user whose credentials are affected. If omitted, the current user's credentials are affected. This value is ignored for monitoring credentials.

• **oracle_homes**

Name of Oracle homes on the target host. Credentials are cleared for all specified homes.

Example

The following example clears the credential set DBCredsNormal with the user admin1 for the myDB target.

```
emcli clear_credential
      -target_type=oracle_database
      -target_name=myDB
      -credential_set=DBCredsNormal
      -user=admin1
```

clear_default_pref_credential

Clears the named credential set as the default preferred credential for the user. The named credential is not deleted from the credential store. Only the user preference to use the named credential as the default preferred credential is cleared.

Format

```
emcli clear_default_pref_cred
      -set_name="set_name"
      -target_type="ttype"
```

Options

• **?set_name**

Clears the default preferred credential for this credential set.

• **target_type**

Target type for the credential set.

Example

This example clears the default preferred credential set for the host target type for the HostCredsNormal credential set.

```
emcli clear_default_pref_cred
      -set_name=HostCredsNormal
      -target_type=host
```

clear_default_privilege_delegation_setting

Clears the default privilege delegation settings for a specified platform.

Format

Standard Mode

```
emcli clear_default_privilege_delegation_setting
        -platform_list="PLATFORM_DEFAULT"
```

Interactive or Script Mode

```
clear_default_privilege_delegation_setting(
        platform_list="PLATFORM_DEFAULT"
        )
```

```
[ ]  indicates that the parameter is optional
```

Options

- **platform_list**

  Comma-separated list of platforms for which default privilege delegation settings are removed. Supported platforms: Linux, HP-UX, SunOS, and AIX.

Exit Codes

0 on success. A non-zero value means verb processing was not successful.

Example

This example clears the default privilege delegation setting for Linux, HP-UX, SunOS, and AIX platforms.

```
emcli clear_default_privilege_delegation_setting
-platform_list="Linux,HP-UX,SunOS,AIX"
```

clear_monitoring_credential

Clears the monitoring credential set for the target.

Format

```
emcli clear_monitoring_credential
        -set_name="set_name"
        -target_name="target_name"
        -target_type="ttype"
```

Options

- **set_name**

  Clears the monitoring credential for this credential set.

- **target_name**

  Clears the preferred credential for this target.

- **target_type**

    Target type for the target/credential set.

Example

This example clears the monitoring credential set for the target testdb.example.com for the DBCredsMonitoring credential set.

```
emcli clear_monitoring_credential
        -set_name=DBCredsMonitoring
        -target_name=testdb.example.com
        -target_type=oracle_database
```

clear_preferred_credential

Clears the named credential set as the target preferred credential for the user. The named credential is not deleted from the credential store. Only the user preference to use the named credential as the preferred credential is cleared.

Format

```
emcli clear_preferred_credential
        -set_name="set_name"
        -target_name="target_name"
        -target_type="ttype"
```

Options

- **set_name**

    Sets the preferred credential for this credential set.

- **target_name**

    Clears the preferred credential for this target.

- **target_type**

    Target type for the target/credential set.

Example

This example clears the preferred credential set for the host target test.example.com for the HostCredsNormal credential set.

```
emcli clear_preferred_credential
        -set_name=HostCredsNormal
        -target_name=test.example.com
        -target_type=host
```

clear_privilege_delegation_setting

Clears the privilege delegation setting from a given host or hosts.

Format

```
emcli clear_privilege_delegation_setting
        -host_names="name1;name2;..."
        [-input_file="FILE:file_path"]
        [-force="yes/no"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **host_names**

  Names of the hosts.

- **input_file**

  Path of the file that has the list of hosts. The file should have one host name per line.

  For more information about the input_file option, see -input_file Syntax Guidelines.

- **force**

  If set to yes, invalid and unreachable targets are ignored and the setting is removed from all valid and up targets. If set to no, invalid and down targets raise an error. The default is no.

Example

The following example force clears the privilege delegation setting from the host listed in the file.text input file.

```
emcli clear_privilege_delegation_setting
      -input_file="FILE:/home/user/file.txt"
      -force=yes
```

clear_problem

Clears problems matching the specified criteria (problem key, target type, and age). Only users with Manage Target privilege can clear the problems for a target. When a problem is cleared, the underlying incidents and events are also cleared.

By default, the problem notification is not sent out. You can override this by specifying the send_notification option. Clearing the underlying incidents and events does not send out a notification.

Format

```
emcli clear_problem
      -problem_key="problem_key"
      -target_type="target_type"
      -older_than="age_of_problem"
      [-target_name="target_name"]
      [-unacknowledged_only="clear_unacknowledged_problems"]
      [-send_notification="send_notifications_for_problems"]
      [-preview]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **problem_key**

  Problem key of the problem to be cleared

- **target_type**

  Internal type name, such as oracle_database for "Oracle Database." You can use the get_target_types command to get the internal name for a target type.

- **older_tha n**

  Specify the age (in days) of the problem.

- **target_name**

  Name of an existing non-composite target. For example, the name of a single database. You cannot use the name of composite targets (target group).

- **unacknowledged_only**

  If provided, only the unacknowledged problems are cleared. This option does not require any value.

- **send_notification**

  If provided, any applicable notification is sent out for cleared problems. By default, no notification is sent for cleared problems. This optiondoes not require any value.

- **preview**

  Gets the number of problems that the command would clear.

Examples

Example 1

This example clears ORA-600 problems across all databases that have occurred (based on the occurrence date of the first incident) for at least 3 days.

```
emcli clear_problem -problem_key="ORA-600" -target_type="oracle_database" -
older_than="3"
```

Example 2

This example sends applicable notifications when the problem clears. By default, a notification is not sent for the cleared problems.

```
emcli clear_problem -problem_key="ORA-600" -target_type="oracle_database"
-older_than="3" - send_notification
```

clear_stateless_alerts

Clears the stateless alerts associated with the specified target. Only a user can clear these stateless alerts; the Enterprise Manager Agent does not automatically clear these alerts. To find the metric internal name associated with a stateless alert, use the get_metrics_for_stateless_alerts verb.

You cannot use this command to clear stateless alerts associated with diagnostic incidents. You can only clear these alerts in the Enterprise Manager console by clearing their associated Incident or Problem.

Format

```
emcli clear_stateless_alerts        -older_than=number_in_days        -
target_type=target_type       -target_name=target_name         [-
include_members]        [-
metric_internal_name=target_type_metric:metric_name:metric_column]       [-
unacknowledged_only]       [-ignore_notifications]       [-preview]

[ ]  indicates that the parameter is optional
```

Options

- **older_than**

  Specify the age of the alert in days. (Specify 0 for currently open stateless alerts.)

- **target_type**

  Internal target type identifier, such as host, oracle_database, and emrep.

- **target_name**

  Name of the target.

- **include_members**

  Applicable for composite targets to examine alerts belonging to members as well.

- **metric_internal_name**

  Metric to be cleaned up. Use the get_metrics_for_stateless_alerts verb to see a complete list of supported metrics for a given target type.

- **unacknowledged_only**

  Only clear alerts if they are not acknowledged.

- **ignore_notifications**

  Use this option if you do not want to send notifications for the cleared alerts. This may reduce the notification sub-system load.

- **preview**

  Shows the number of alerts to be cleared on the target(s).

Example

This example clears alerts generated from the database alert log over a week old. In this example, no notifications are sent when the alerts are cleared.

```
emcli clear_stateless_alerts -older_than=7 -target_type=oracle_database -tar
get_name=database -metric_internal_name=oracle_database:alertLog:genericErrStack -
ignore_notifications
```

clone_as_home

Clones the specified Application Server Oracle Home or S/W Library component from the target host to specified destinations. For a Portal and Wireless installation, the OID user and password are also needed. For a J2EE instance connected to only a DB-based repository, a DCM Schema password is needed.

Passing Variables Through EM CLI

When working with variables such as `%perlbin%` or `%oracle_home%`, EM CLI passes variable values from the current local environment instead of the variables themselves. To pass variables through an EM CLI command, as might be the case when using the `-prescripts` or `-postscripts` options, you can place the EM CLI command in a batch file and replace all occurrences of `%` with `%%`.

Format

```
emcli clone_as_home
        -input_file="dest_properties:file_path"
        -list_exclude_files="list of files to exclude"
        -isSwLib="true/false"
        -tryftp_copy="true/false"
        -jobname="name of cloning job"
        -iasInstance=instance
        -isIas1013="true/false"
        [-oldIASAdminPassword=oldpass]
        [-newIASAdminPassword=newpass]
        [-oldoc4jpassword=oldpass]
        [-oc4jpassword=newpass]
        [-oiduser=oid admin user]
        [-oidpassword=oid admin password]
        [-dcmpassword=dcm schema password]
        [-prescripts="script name to execute"]
        [-run_prescripts_as_root="true/false"]
        [-postscripts="script to execute"]
        [-run_postscripts_as_root="true/false"]
        [-rootscripts="script name to execute"]
        [-swlib_component ="path:path to component;version:rev"]
        [-source_params="TargetName:name;HomeLoc:loc;HomeName:name;
          ScratchLoc:Scratch dir Location"
        [-jobdesc="description"]
```

```
[ ] indicates that the parameter is optional
```

Options

* **input_file="dest_properties:file_path"**

  File containing information regarding the targets.

  Each line in the file corresponds to information regarding one destination.

  Format:

  ```
  Destination Host Name1;Destination Home Loc; Home Name; Scratch
  Location;
  ```

  For more information about the input_file option, see -input_file Syntax Guidelines.

* **list_exclude_files**

  Comma-separated list of files to exclude. Not required if the source is software lib.
  "*" can be used as a wild card.

* **isSwLib**

  Specifies whether it is an Oracle Home database or Software Library.

* **ryftp_copy**

  Try FTP to copy or not. You should set the FTP copy option to false when using
  EM CLI from the command line.

* **jobname**

  Name of the cloning job.

* **iasInstance**

  Name of instance.

- **islas1013**

  Specifies whether this is a 10.2.3 Ias home.

- **oldoc4jpassword**

  Old OC4j password. (Required for 10.1.3 Ias homes.)

- **oc4jpassword**

  New OC4J password. (Required for 10.1.3Ias homes.)

- **oldIASAdminPassword**

  Old Application Server administrator password. (Not required for 10.1.3 Ias homes.)

- **newIASAdminPassword**

  New Application Server administrator password. (Not required for 10.1.3 Ias homes.)

- **oiduser**

  OID admin user.

- **oidpassword**

  OID admin password.

- **dcmpassword**

  DCM schema password.

- **prescripts**

  Path of script to execute.

  > **Note:**
  >
  > Double-quoted options can be passed using an escape (\) sequence. For example:
  >
  > ```
  > prescripts=" <some value here>=\"some value here\" "
  > ```

- **run_prescripts_as_root**

  Run prescripts as "root". By default, the option is set to false.

- **postscripts**

  Path of script to execute.

- **run_postscripts_as_root**

  Run postscripts as "root". By default, the option is set to false.

- **rootscripts**

  Path of the script to execute. The job system environment variables (`%oracle_home%`, `%perl_bin%`) can be used for specifying script locations.

- **swlib_component**

  Path to the Software Library to be cloned. "isSwLib" must be true in this case.

- **source_params**

  Source Oracle home information. "isSwLib" must be false in this case.

- **jobdesc**

  Description of the job. If not specified, a default description is generated automatically.

Example

```
emcli clone_as_home
   -input_file="dest_properties:/home/destinations.txt"
   -list_exclude_files="centralagents.lst"
   -isSwLib="false"
   -tryftp_copy="false"
   -jobname="clone as home"
   -iasInstance="asinstancename"
   -isIas1013="false"
   -oldIASAdminPassword="oldpassword"
   -newIASAdminPassword="newpassword"
   -prescripts="/home/abc/myscripts"
   -run_prescripts_as_root="true"
   -rootscripts="%oracle_home%/root.sh"
   -source_params="TargetName:host.domain.com;HomeLoc=/home/oracle/appserver1;
    HomeName=oracleAppServer1;ScratchLoc=/tmp"
```

clone_crs_home

Creates an Oracle Clusterware cluster given a source Clusterware home location or a Clusterware S/W Library component for specified destination nodes.

Format

```
emcli clone_crs_home
      -input_file="dest_properties:file_path"
      -list_exclude_files="list of files to exclude"
      -isSwLib="true/false"
      -tryftp_copy="true/false"
      -jobname="name of cloning job"
      -home_name="name of home to use when creating Oracle Clusterware cluster"
      -home_location="location of home when creating Oracle Clusterware cluster"
      -clustername=name of cluster to create
      [-isWindows="false/true"]
      [-ocrLoc=ocr location]
      [-vdiskLoc=voting disk location]
      [-prescripts="script name to execute"]
      [-run_prescripts_as_root="true/false"]
      [-postscripts="script to execute"]
      [-run_postscripts_as_root="true/false"]
      [-rootscripts="script name to execute"]
      [-swlib_component ="path:path to component;version:rev"]
      [-source_params="TargetName:name;HomeLoc:loc;HomeName:name;
        ScratchLoc:Scratch dir Location"]      [-jobdesc="description"]
```

[ ] indicates that the parameter is optional

Options

- **input_file="dest_properties:file_path"**

  File containing information regarding the targets.

  Each line in the file corresponds to information regarding one destination.

Format:

```
Destination Host Name;Destination Node Name;Scratch
Location;PVTIC;VirtualIP;
```

For more information about the input_file option, see -input_file Syntax Guidelines.

- **list_exclude_files**

  Comma-separated list of files to exclude. Not required if the source is software lib. An asterisk "*" can be used as a wildcard.

- **isSwLib**

  Specifies whether it is an Oracle Home database or Software Library.

- **tryftp_copy**

  Try FTP to copy or not. You should set the FTP copy option to false when using `emcli` from the command line.

- **jobname**

  Name of the cloning job.

- **home_name**

  Name of the home to use for all homes in the Oracle Clusterware cluster.

- **home_location**

  Location of the home to use for all homes in the Oracle Clusterware cluster.

- **clustername**

  Name of the cluster to create.

- **isWindows**

  Specify whether the cloning source is on a Windows Platform. This option only applies for creating CRS cloning from a Gold Image source. The default value is false.

- **ocrLoc**

  Oracle Cluster Registry Loaction.

- **vdiskLoc**

  Voting disk location.

- **prescripts**

  Path of the script to execute.

  > **✎ Note:**
  >
  > Double-quoted options can be passed using an escape (\) sequence. For example:
  >
  > ```
  > prescripts=" <some value here>=\"some value here\" "
  > ```

- **run_prescripts_as_root**

  Run prescripts as "root". By default, this option is set to false.

- **postscripts**

Path of the script to execute.

- **run_postscripts_as_root**

  Run postscripts as "root". By default, it is false.

- **rootscripts**

  Path of the script to execute.

- **swlib_component**

  Path to the Software Library to be cloned. "isSwLib" must be true in this case.

- **source_params**

  Source Oracle home info. "isSwLib" must be false in this case.

- **jobdesc**

  Description of the job. If not specified, a default description is generated automatically.

Example

```
emcli clone_crs_home -input_file="dest_properties:crs.prop"  -isSwLib="true"
  -tryftp_copy="true" -jobname="crs cloning job2" -home_name="cloneCRS1"
  -home_location="/scratch/scott/cloneCRS1 " -clustername="crscluster"
  -ocrLoc="/scratch/shared/ocr" -vdiskLoc="/scratch/shared/vdisk"
  -postscripts="%perlbin%/perl%emd_root%/admin/scripts/cloning/samples/
   post_crs_ create.pl ORACLE_HOME=%oracle_home%"
  -run_postscripts_as_root="true" -rootscripts="%oracle_home%/root.sh"
  -swlib_component="path:Components/crscomp;version:.1"
```

Passing Variables Through EM CLI

When working with variables such as `%perlbin%` or `%oracle_home%`, EM CLI passes variable values from the current local environment instead of the variables themselves. To pass variables through an EM CLI command, as might be the case when using the `-prescripts` or `-postscripts` options, you can place the EM CLI command in a batch file and replace all occurrences of `%` with `%%`.

clone_database

Clones a database.

Format

```
emcli clone_database
      -source_db_name="source_database_name"
      -dest_global_dbname="global_name_of_clone_database"
      -dest_oracle_sid="clone_database_instance_name"
      [-dest_host_name="clone_host_name"]
      [-dest_oracle_home="clone_database_oracle_home"]
      [-source_db_creds_name="source_database_credential_name"]
      [-source_host_creds_name="source_database_host_credential_name"]
      [-dest_host_creds_name="clone_database_host_credential_name"]
      [-asm_inst_creds_name="asm_instance_credential_name"]
      [-dest_target_name="clone_database_name"]
      [-clone_type="clone_type"]
      [-source_staging_area="source_staging_directory"]
      [-dest_staging_area="clone_database_staging_directory"]
      [-dest_storage_type="clone_database_storage_type"]
```

```
[-dest_database_area="clone_database_files_location"]
[-dest_recovery_area="clone_database_fast_recovery_area"]
[-dest_listener_selection="clone_database_listener_selection"]
[-dest_listener_name="clone_database_listener_name"]
[-dest_listener_port="clone_database_listener_port"]
[-configure_with_oracle_restart"]
[-job_name="job_name"]
[-job_desc="job_description"]
[-src_ssh_tunnel_port="<ssh tunnel port used by standby to connect to primary>"]
[-dest_ssh_tunnel_port="<ssh tunnel port used by primary to connect to standby>"]
[-src_gateway_cred_name="<primary database host hybrid gateway agent credential
name>"]
[-dest_gateway_cred_name="<standby database host hybrid gateway agent credential
name>"]
[-dest_GI_host_creds_name="<standby database grid infrastructure credential
name>"]
[-tde_wallet_creds_name="<transparent data encryption wallet credentials of the
source database>"]
```

`[ ]` indicates that the parameter is optional

Options

- **source_db_name**

  Source database Enterprise Manager target name. Can be either a single-instance database or a cluster database instance.

- **dest_global_dbname**

  Clone database global database name. Usually specified as <name>.<domain>, with <name> being used for the db_unique_name and <domain> for the db_domain_name options.

- **dest_oracle_sid**

  Clone database instance name.

- **dest_host_name**

  Clone database host name. If not specified, the clone database is created on the same host as the source database.

- **dest_oracle_home**

  Clone database Oracle home. If not specified, the Oracle Home of the source database is used.

- **source_db_creds_name**

  Source database named credential.

- **source_host_creds_name**

  Source database host named credential.

- **dest_host_creds_name**

  Destination (clone) host named credential.

- **asm_inst_creds_name**

  Automatic Storage Management (ASM) named credential.

- **dest_target_name**

  Clone database Enterprise Manager target name.

ORACLE®

- **clone_type**

  Type of source database backup that will be used for cloning. Valid values are:

  – DUPLICATE — Database files are moved directly to the clone database host by the Recovery Manager (RMAN).

  – STAGING — Database files are backed-up into the staging area and moved to the clone database host through HTTP.

  – EXISTING_BACKUP — Database files are restored from existing backups to the clone database host by the Recovery Manager (RMAN).

- **pitr_date**

  Clone database as of the specified date in MM/dd/yyyy hh:mm:a (Month/Date/Year Hours:Minutes:AM/PM marker) format. For example: 03/22/2014 08:25:AM. If not specified, the clone database is created as of the latest point-in-time. This option is applicable when the clone_type is EXISTING_BACKUP.

- **pitr_scn**

  Clone database as of the specified System Change Number of the source database. If not specified, the clone database is created as of the latest point-in-time. This option is applicable when the clone_type is EXISTING_BACKUP.

- **encryption_mode**

  Encryption mode of the existing source database backups. If not specified, the default value is NONE. This option is applicable when the clone_type is EXISTING_BACKUP. Valid values are:

  – WALLET — Backups are encrypted using Oracle Encryption Wallet.

  – PASSWORD — Backups are encrypted using a password.

  – DUAL — Backups are encrypted using both Oracle Encryption Wallet and a password.

- **backups_encryption_creds_name**

  Database named credential for the encrypted backups. This option is applicable if encryption_mode is PASSWORD or DUAL.

  > **Note:**
  >
  > this option is applicable only if the clone type is EXISTING_BACKUP and the database backups are encrypted using a password. This database credential should be created in Enterprise Manager of scope GLOBAL with the user name specified as "backup_admin".

- **tape_settings**

  Media management vendor settings if the database backups are on tape. This option is applicable when clone_type is EXISTING_BACKUP.

- **db_backups_location**

  The location of the backups to be transferred to the destination host. Multiple values can be specified using "," as a delimiter. This option is applicable when cloning to a different host and clone_type is EXISTING_BACKUP.

> ✎ **Note:**
>
> this option is applicable only if the clone type is EXISTING_BACKUP and the database clone occurs on a different host where the source database backups are not visible. If the backups are visible from the destination host, this option should NOT be specified.
>
> – It is recommended that if the size of the database backups is very large, the backups should be taken in a common location visible from the destination host.
>
> – If the source database backups are on ASM diskgroups, ensure that the diskgroups are mounted at the destination host as these backups are not transferred.
>
> – When you specify this option is specified, all of the available files at this location are transferred to a temporary staging location at the destination host.
>
> – You can specify multiple values for this option with comma (,) as a delimiter.

- **source_staging_area**

  Staging area used to store the backup of source database. This option is applicable when clone_type is STAGING.

- **dest_staging_area**

  Staging area used to store backup files transferred from source host. This option is applicable when clone_type is STAGING.

- **dest_storage_type**

  Clone database storage type. Valid values are:

  – FILE_SYSTEM — Clone database files will be in a regular file system (using Oracle-managed Files).

  – ASM — The clone database will use Automatic Storage Management (ASM).

- **dest_database_area**

  Oracle-managed files (OMF) location for clone database files. This can be a regular file system (if storage_type is FILE_SYSTEM) or an ASM disk group (if storage_type is ASM). If not specified, a default value is used.

- **dest_recovery_area**

  Fast recovery area location. If not specified, a default value is used.

- **dest_listener_selection**

  Clone database listener selection. Valid values are:

  – GRID_INFRA — Use Grid Infrastructure Home listener.

  – DEST_DB_HOME — Use the listener from the clone database Oracle Home.

- **dest_listener_name**

  Clone database listener name. This option is applicable only if dest_listener_selection is DEST_DB_HOME. If not specified, the first existing TCP listener found in the clone database Oracle Home is used. If you specify this option, you must also specify dest_listener_port.

- **dest_listener_port**

  Clone database listener port. This option is applicable only if
  dest_listener_selection is DEST_DB_HOME. If you specify this option, you must
  also specify dest_listener_name.

- **configure_with_oracle_restart**

  Configure the clone database with Oracle Restart if the clone host has Oracle
  Restart configured. Oracle Restart automatically starts the database when
  required.

- **job_name**

  Unique job name for the clone job in the Enterprise Manager repository.

- **job_desc**

  Job description.

- **src_ssh_tunnel_port**

  SSH Tunnel port used by the standby database to connect to the primary
  database. This is the port created on the standby host to forward the connection
  request to the primary database listener port.

- **dest_ssh_tunnel_port**

  SSH Tunnel port used by the primary database to connect to the standby
  database. This is the port created on the primary host to forward the connection
  request to the standby database listener port.

- **src_gateway_creds_name**

  Hybrid Gateway Agent named credential for the primary database host.

- **dest_gateway_creds_name**

  Hybrid Gateway Agent named credential for the standby database host.

- **dest_GI_host_creds_name**

  Grid Infrastructure named credentials for an operating system user who can
  access the grid infrastructure Oracle Home.

- tde_wallet_creds_name

  Transparent Data Encryption wallet credentials for the source database. Use
  create_named_credential verb to create these credentials as shown below:

```
emcli create_named_credential
    -cred_name=WC1 -cred_type=GenericPassword
    -auth_target_type='<system>'
    -attributes="GENERIC_PASSWORD:<Source Database TDE Wallet Password>"
```

Example 1

```
emcli clone_database
    -source_db_name="database"
    -dest_target_name="dbClone1"
    -dest_host_name="host1"
    -dest_oracle_home="/ade/ngade_gct/oracle"
    -dest_oracle_sid="dbClone1"
    -dest_global_dbname="dbClone1"
    -dest_listener_selection="DEST_DB_HOME"
    -clone_type="DUPLICATE"
    -dest_storage_type="FILE_SYSTEM"
```

Example 2

```
emcli clone_database
    -source_db_name="database"
    -source_db_creds_name="NC_DBCREDS1"
    -source_host_creds_name="NC_HOST_CREDS1"
    -dest_host_name="host1"
    -dest_host_creds_name="NC_HOST_CREDS2"
    -dest_oracle_home="/u01/app/oracle/product/11.2.0/dbhome_2"
    -dest_oracle_sid="TESTDB1" -dest_global_dbname="TESTDB1"
    -dest_listener_selection="GRID_INFRA"
    -clone_type="EXISTING_BACKUP"
    -dest_storage_type="FILE_SYSTEM"
    -db_backups_location="/oracle/dir1"
```

Example 3

The following command creates a clone database on the Cloud host 'cloudhost.oracle.com' for a source database that is encrypted with TDE. Communication between the source and cloned databases will be established using the specified tunnel port.

```
emcli clone_database
    -source_db_name="database"
    -source_db_creds_name="NC_DBCREDS1"
    -source_host_creds_name="NC_HOST_CREDS1"
    -dest_host_name="cloudhost.oracle.com"
    -dest_host_creds_name="OPC_SSH_NAMED_CREDS"
    -dest_oracle_home="/u01/app/oracle/product/12.1.0/dbhome_2"
    -dest_oracle_sid="TESTDB1"
    -dest_global_dbname="TESTDB1"
    -dest_listener_selection="GRID_INFRA"
    -clone_type="DUPLICATE"
    -dest_storage_type="FILE_SYSTEM"
    -dest_gateway_creds_name='DEST_GATEWAY_CREDS'
    -dest_GI_host_creds_name="DEST_GRID_CREDS"
    -src_ssh_tunnel_port="4001"
    -dest_ssh_tunnel_port="4001"
    -configure_with_oracle_restart"
    -tde_wallet_creds_name='WC1'
```

clone_database_home

Clones the specified Oracle Home or S/W Library from the target host to specified destinations. If the isRac option is true, a RAC cluster is created. If the isRac option is true, the home name and location of the RAC cluster are needed.

Format

```
 emcli clone_database_home
      -input_file="dest_properties:file_path"
      -list_exclude_files="files_to_exclude"
      -isSwLib="true|false"
      -isRac="true|false"
      -tryftp_copy="true|false"
      -jobname="name_of_cloning_job"
      [-home_name="home_when_creating_RAC_cluster"]
      [-home_location="location_of_home_when_creating_RAC_cluster"]
      [-prescripts="script_name_to_execute"]
      [-run_prescripts_as_root="true|false"]
      [-postscripts="script_to_execute"]
```

```
[-run_postscripts_as_root="true|false"]
[-rootscripts="script_name_to_execute"]
[-swlib_component ="path:path_to_component;version:rev"]
[-source_params="TargetName:name;HomeLoc:loc;HomeName:name;
  ScratchLoc:scratch_dir_location"
[-jobdesc="description"]
```

`[ ]` indicates that the parameter is optional

Options

- **input_file=dest_properties**

  File containing information regarding the targets. Each line in the file corresponds to information regarding one destination.

  Format if cloning a database (isRac is false):

  ```
  Destination Host Name1;Destination Home Loc; Home Name; Scratch
  Location;
  ```

  Format if cloning a RAC cluster (isRac is true):

  ```
  Host Name;Node Name;Scratch Location;
  ```

  For more information about the input_file option, see -input_file Syntax Guidelines.

- **list_exclude_files**

  Comma-separated list of files to exclude. This is not required if the source is software lib. "*" can be used as a wild card.

- **isSwLib**

  Specifies whether the source is an Oracle Home database or Software Library.

- **isRac**

  Specifies whether cloning in RAC mode. If the isRac option is true, a RAC cluster is created. If the isRac option is true, the home name and location of the RAC cluster are needed.

- **tryftp_copy**

  Try FTP to copy or not. You should set the FTP copy option to false when using EM CLI from the command line.

- **jobname**

  Name of the cloning job.

- **home_name**

  Name of the home to use when creating a RAC cluster.

- **home_location**

  Location of the home to use when creating a RAC cluster.

- **prescripts**

  Path of the script to execute.

> **Note:**
>
> Double-quoted options can be passed using an escape (\) sequence. For example:
>
> ```
> prescripts=" <some value here>=\"some value here\" "
> ```

- **run_prescripts_as_root**

  Run prescripts as "root". By default, it is false.

- **postscripts**

  Path of the script to execute.

- **run_postscripts_as_root**

  Run postscripts as "root". By default it is false.

- **rootscripts**

  Path of the script to execute. You can use the job system environment variables (`%oracle_home%`, `%perl_bin%`) to specify script locations.

- **swlib_component**

  Path to the Software Library to be cloned. "isSwLib" must be true in this case.

- **source_params**

  Source Oracle home info. "isSwLib" must be false in this case.

- **jobdesc**

  Description of the job. If not specified, it is automatically generated.

Example

```
emcli clone_database_home
     -input_file="dest_properties:clonedestinations"
     -list_exclude_files="*.log,*.dbf,sqlnet.ora,tnsnames.ora,listener.ora"
     -isSwLib="false"
     -isRac="false"
     -tryftp_copy="false"
     -jobname="clone database home"
     -prescripts="/home/joe/myScript"
     -run_prescripts_as_root="true"
     -rootscripts="%oracle_home%/root.sh"
     -source_params="TargetName:host.domain.com;HomeLoc=/oracle/database1;
HomeName=OUIHome1;ScratchLoc=/tmp"
```

Passing Variables Through EM CLI

When working with variables such as `%perlbin%` or `%oracle_home%`, EM CLI passes variable values from the current local environment instead of the variables themselves. To pass variables through an EM CLI command, as might be the case when using the `-prescripts` or `-postscripts` options, you can place the EM CLI command in a batch file and replace all occurrences of `%` with `%%`.

collect_metric

Performs an immediate collection and threshold evaluation of a set of metrics associated with the specified internal metric name. Metric data collection and threshold evaluation occur asynchronously to the EM CLI call.

You typically use this command when you believe you have resolved an open metric alert or error and would like to clear the event by immediately collecting and reevaluating the metric. This command applies to most metrics except server-generated database metrics.

Use the get_on_demand_metrics verb to see a complete list of supported metrics for a given target.

Format

```
 emcli collect_metric         -target_name=name         -target_type=type        -
metric_name=metric_name | -collection_name=user_defined_metric_name

[ ]  indicates that the parameter is optional
```

Options

*   **target_name**

    Name of the target.

*   **target_type**

    Internal target type identifier, such as host, oracle_database, and emrep.

*   **metric_name**

    Internal name that represents a set of metrics that are collected together. Use the get_on_demand_metrics verb to see the supported list of metrics for a given target.

*   **collection_name**

    Name of the metric extension. This option only applies to metric extensions.

Examples

Example 1

If you want to collect the "CPU Utilization (%)" metric, look for the appropriate metric internal name (which is Load) using the get_on_demand_metrics command, then run the command as follows:

```
emcli collect_metric  -target_type=host  -target_name=hostname.example.com  -
metric_name=Load
```

Example 2

This example immediately collects and evaluates thresholds for the metric extension called MyUDM:

```
emcli collect_metric -target_type=host -target_name=hostname.example.com  -
collection=MyUDM
```

compare_sla

Compares two SLAs as defined by two XML files. This utility outputs the difference trees as sla1_compare.dif and sla2_compare.dif in the specified directory. You can use a diff utility to diff these two files. Compare two sla.xml's to find out the difference.

Format

```
emcli compare_sla    -input_file=sla1:'first_xml    -input_file=sla2:'second_xml'    [-
dir='directory']
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **input_file=sla1**

    File name for the first XML file.

    For more information about the input_file option, see -input_file Syntax Guidelines.

*   **input_file=sla2**

    File name for the second XML file.

    For more information about the input_file option, see -input_file Syntax Guidelines.

*   **dir**

    The default is the current directory. If you need to specify another directory, use this option for the output files sla1_compare.dif and sla2_compare.dif.

Example

This example compares two SLAs as defined in sla1.xml and sla2.xml, and outputs sla1_compare.dif and sla2_compare.dif in the current directory.

```
emcli compare_sla
          -input_file=sla1:sla1.xml -input_file=sla2:sla2.xml
```

You can use a standard diff tool to diff the files, such as This example for Linux:

```
diff sla1_compare.dif sla2_compare.dif
```

complete_post_pdb_relocation

Performs cleanup operations and complete pluggable database relocation in the maximum availability mode.

> **Note:**
>
> This verb is applicable only for database versions 12.2 (and above).

> **✏️ Note:**
>
> When a pluggable database of version 12.2 (and above) is relocated in
> maximum availability mode, the source pluggable database target still
> continues to exist in Enterprise Manager after the successful relocation. In
> the Maximum Availability mode, the existing client connections is redirected
> by the source CDB's listener to the relocated PDB. To facilitate this
> connection redirection, the source pluggable database exists in mounted
> state. Eventually, after all the client connect descriptors are migrated by the
> user manually to point to the relocated pluggable database, the source can
> be removed. This verb should be run only on the source pluggable database
> target to remove it and also to delete the pluggable database from backend.

Format

```
emcli complete_post_pdb_relocation
      -pdb_target_name="source pluggable database target name"
      -input_file="path of the input file"
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   **pdb_target_name**

    Name of the source pluggable database target.

*   **input_file**

    Path of the file containing the following input properties:

    ```
        SRC_CDB_CRED = SYSDBA source container database credentials (format -
    CRED_NAME:OWNER)
        SRC_HOST_CRED = Source container database host credentials (format -
    CRED_NAME:OWNER)
    ```

Example

Example 1

This example completes the post relocation actions to delete the source pluggable
database "SICDB_GEN", which was relocated to another container database in the
maximum availability mode.

```
emcli complete_post_pdb_relocation
      -pdb_target_name="SICDB_GEN"
      -input_file=data:/u01/post_relocate.props
```

config_compare

Submits the configuration comparison job.

Format

```
emcli config_compare
      -target_type="oracle_database"
      -first_config="Test Database"
      -second_config="SYSMAN"
```

```
            -job_name="Test Compare Job"
            [-schedule=
            {
                start_time:yyyy/MM/dd HH:mm;
                tz:{java timezone ID};
                frequency:interval/weekly/monthly/yearly;
                repeat:#m|#h|#d|#w;
                months:#,#,...;
                days:#,#,...;
                end_time:yyyy/MM/dd HH:mm;
                grace_period:;
            }]
            [-template_id="18"]
            [-job_description="Test Description"]
            [-mapping_display="Tree"]
            [-email_address]
            [-save_mode="save_all|save_only_diffs"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- target_type

  Target type on which the comparison job is being submitted. The value should be the internal name. To get the internal name, execute the following EM CLI command:

  ```
  emcli get_target_types
  ```

- first_config

  Name of the first configuration, which can be either the latest configuration or a saved configuration of a target. If submitting the latest configuration, provide the target name. If submitting a saved configuration, the format should be:

  ```
  "target_name|saved_configuration_name(which is the "name" field from the output of
  "emcli get_saved_configs"
  ```

- second_config

  Names of the second and subsequent configurations, which can contain one or more latest configurations and/or one or more saved configurations of one or more targets. Multiple configurations can be specified, separated by a comma. If the latest configuration needs to be submitted, provide the target name. If the saved configuration needs to be submitted, then the format should be:

  ```
  "target_name|saved_configuration_name(which is the "name" field from the output of
  "emcli get_saved_configs"
  ```

- job_name

  Name of the comparison job.

- schedule

  Schedule with which the comparison job must be scheduled. If the schedule option is not provided, the comparison job runs immediately.

  – start_time - Time when comparison job has to start executing. The format is "yyyy/MM/dd HH:mm"

    tz - Timezone ID (optional)

  – frequency - Valid values are once/interval/weekly/monthly/yearly. (optional)

If frequency is set to interval, repeat must be specified.

If frequency is set to weekly or monthly, days must be specified.

If frequency is set to yearly, both days and months must be specified.

– repeat - Frequency with which the comparison job must be repeated. (Required only if frequency is set to interval.)

– days - Comma-separated list of days. (Required only if frequency is weekly, monthly, or yearly.) Example: "repeat=1d"

If frequency is weekly, then the valid range is 1 to 7 inclusive.

If frequency is monthly or yearly, then the valid range is 1 to 30 inclusive.

– months - Comma-separated list of months. (Required only if frequency is yearly). Valid range is 1 to 12 inclusive.

– end_time - End time for comparison job executions. (optional). If it is not specified, the comparison job runs indefinitely. The format is "yyyy/MM/dd HH:mm"

– grace_period - grace period in minutes (optional)

- template_id

  ID of the template. The value is an integer.

- job_description

  Description of the comparison job.

- mapping_display

  Can be either "tree" or "table." The default value is "tree". This option is only for composite targets. Note: When "template_id" is specified, do not specify mapping_display.

- email_address

  Email address to which notification mail is to be sent, if differences are found.

- save_mode

  Tells the comparison engine whether to save all the results or only the differences. Valid inputs are "save_all" and "save_only_diffs". The save_only_diffs option saves the differences to the Management Repository. Otherwise, all the comparison results are saved. The default value is "save_only_diffs".

Checking the Job Status:

Once submitted, the comparison job's status can be viewed by issuing the following EM CLI command:

```
emcli get_jobs -name="jobName"
```

Aborting the Job:

Once submitted, the comparison job can be aborted by issuing the following EM CLI command:

```
emcli stop_job -name="jobName"
```

Examples

Example 1

This example compares the latest configuration of one target to the latest configurations of multiple targets. All the comparison results will be saved.

```
emcli config_compare
    -target_type="oracle_database"
    -first_config="Test Database"
    -second_config="Test Database","Test_Database"
    -job_name="Test Job" -template_id="18"
    -save_mode="save_all"
```

Example 2

This example compares the latest configuration with the saved configuration specifying a start_time.

```
emcli config_compare
    -target_type="oracle_database"
    -first_config="Test Database"
    -second_config="Test Database|Test Database|oracle_
        database|20140101224530","Test_Database" -job_name="Test Job"
    -schedule="start_time:2014/06/10 15:45"
```

configure_db_ha

Perform database high availability (HA) configuration tasks. This verb has multiple subcommands that perform different HA-related operations. The available subcommands are mentioned in detail as follows:

configure_db_ha -configureBackupToRA

Configures one or more databases for protection by one or more Recovery Appliance(s). The command schedules an Enterprise Manager deployment procedure that processes multiple databases and configures each to send backups to a designated Recovery Appliance. In addition, the command also enables redo transport to the Recovery Appliance(s) either from individual databases or from within a Data Guard broker configuration. The procedure steps include configuring a database backup wallet, installing the Recovery Appliance Backup Module in the database Oracle homes (if necessary), and configuration of all required backup and redo settings in the database and in Enterprise Manager. Before running this command, the specified databases must already be enrolled with the Recovery Appliance as protected databases (as performed by the Recovery Appliance administrator via the *emcli manage_ra -addProtectedDatabase* command or the Enterprise Manager Recovery Appliance management console).

The -backup_config parameter controls the type of databases that will be processed and how they will be configured for backup and redo transport. If this parameter is specified with a value of "NO_DG", only databases that are not in a Data Guard configuration will be configured. If the parameter is specified with a value of "ALL_DG" or "CUSTOM_DG", only databases in a Data Guard broker configuration will be configured to send backups to one or more Recovery Appliances (either the same Recovery Appliance for all databases or different Recovery Appliances designated for different databases), and the Recovery Appliance(s) will subsequently be added as members of the Data Guard broker configuration and redo transport will be enabled to the Recovery Appliance(s).

The command can also be used to change the existing configuration of one or more databases, either for the same Recovery Appliance or for a different Recovery Appliance than the one currently configured. The procedure works identically in the initial configuration and reconfiguration cases, except that in the latter case any existing protected database configuration is overwritten with the new settings.

Wallet Handling

Configuration of the Recovery Appliance virtual private catalog credentials in an Oracle wallet is a requirement for a database to send backups and redo to a Recovery Appliance. Since this configuration must coexist with other database features that use the wallet, it must either be integrated into an existing wallet if present or a new wallet must be created in the proper location. Since the WALLET_LOCATION parameter in SQLNET.ORA is used by some database features (such as redo transport) to determine the wallet location, the location of SQLNET.ORA must also be determined. The configuration procedure submitted by this command will use the following rules to determine the location of SQLNET.ORA and the database wallet:

- The location of SQLNET.ORA for each database will be determined as follows:

  1. Use the value of the TNS_ADMIN environment variable set via the database SRVCTL utility, if it is set.

  2. Otherwise, use the value of TNS_ADMIN in the environment of the Enterprise Manager Agent that is monitoring the database, if it is set.

  3. Otherwise, if the database Oracle home is a read-only Oracle home, use $ORACLE_BASE_HOME/network/admin.

  4. Otherwise, use $ORACLE_HOME/network/admin.

- The location of the backup wallet to be created/updated for each database will be determined as follows:

  1. Create/update the wallet in the location specified by WALLET_LOCATION in SQLNET.ORA.

  2. Otherwise, update the default wallet in $ORACLE_BASE/admin/$ORACLE_UNQNAME/wallet, if it exists.

  3. Otherwise, if -wallet_location is specified, create/update the wallet in the specified location (which is $ORACLE_BASE/admin/$ORACLE_UNQNAME/wallet if USE_RECOMMENDED is specified).

  4. Otherwise, if the database Oracle home is a read-only Oracle home, create/update the wallet in $ORACLE_BASE_HOME/dbs/zdlra.

  5. Otherwise, create/update the wallet in $ORACLE_HOME/dbs/zdlra.

- In case 2 above, WALLET_LOCATION in SQLNET.ORA will be set to the associated wallet location only if the -force_update_wallet_loc argument is specified, in order to avoid inadvertently overriding a default wallet for databases features that use it. In cases 3 through 5, WALLET_LOCATION will be set to the associated value by default, as in these cases there is no default wallet and there is no existing value for WALLET_LOCATION. If WALLET_LOCATION cannot be set for any reason, the following will occur:

  - The *Set Redo Transport User* step of the configuration procedure will fail with an error indicating that the required WALLET_LOCATION setting is not present.

  - All other steps of the procedure will proceed.

  - The databases will be configured to only send backups to the Recovery Appliance(s) and redo transport will not be enabled.

  - If the databases being processed are in a Data Guard configuration, the Recovery Appliance(s) will not be added to the Data Guard broker configuration.

– The status of the procedure will be reported as *Completed with Errors*.

Format

```
emcli configure_db_ha -configureBackupToRA
    (
      (-target_name="<database or group target name>" -target_type="oracle_database|
rac_database|composite")
      | -input_file="<full path name of input file>"
    )
    -backup_config="NO_DG|ALL_DG|CUSTOM_DG"
    [-ra_target_name="<Recovery Appliance or Recovery Appliance Monitoring View target
name>"]
    [-ra_vpc_username="<Recovery Appliance virtual private catalog username>"]
    [-ra_override_conn_desc="<Recovery Appliance database override connect
descriptor>"]
    [-db_cred="<database target named credential>"]
    [-db_host_cred="<database host target named credential>"]
    [-download_backup_module]
    [-force_install_backup_module]
    [-backup_module_directory="<full pathname where backup module will be installed on
database hosts>"]
    [-wallet_location=USE_RECOMMENDED|"<full pathname of backup wallet location>"]
    [-force_crs_setenv]
    [-force_update_wallet_loc]
    [-wallet_cred="<named credential containing wallet credentials for a password-
protected wallet>"]
    [-ship_redo=YES|NO]
    [-skip_controlfile_autobackup]
    [-update_snapshot_controlfile_loc]
    [-parallelism=<# of channels to set in database RMAN settings for Recovery
Appliance backups>]
    [-force_restart_db]
    [-skip_configured_dbs]
    [-force_serial_execution]
    [-schedule=
      {
        start_time:yyyy/MM/dd HH:mm;
        tz:{java timezone ID};
        frequency:interval/weekly/monthly/yearly;
        repeat:#m|#h|#d|#w;
        months:#,#,...;
        days:#,#,...;
        end_time:yyyy/MM/dd HH:mm;
      }
    ]
```

Options

The following conventions are used for the attribute values in the argument descriptions:

- Required: Whether the argument is required on either the command line or in an input file, and if so under what conditions.

- Default: For optional arguments, whether there is a default value.

- Scope:

  – Command Line Only: The argument can be specified only on the command line, not in an input file, and will apply globally to all database targets involved in the command.

– Input File Only: The argument can be specified only in the input file, at a per-target level.

– Both: The argument can be specified on either the command line, in the input file, or both. If specified in both places, the value in the input file will override the corresponding command line argument value.

- backup_config="NO_DG|ALL_DG|CUSTOM_DG"

Specifies the databases that will be configured by this command. A single invocation of the command can operate only on a set of databases that are either Data Guard databases (i.e., primary or standby databases in a Data Guard configuration) or non-Data Guard databases (i.e., not primary or standby databases and not members of a Data Guard configuration). If the set of databases to be configured include a combination of Data Guard and non-Data Guard databases, the command should be invoked twice: once for the Data Guard databases and once for the non-Data Guard databases.

– Required: Yes

– Scope: Command Line Only

– NO_DG:

* Only non-Data Guard databases will be configured to send backups and redo to the Recovery Appliance(s) specified.

– ALL_DG:

* Only the primary databases in one or more Data Guard configurations can be specified as a target (on the command line or in an input file). The Recovery Appliance(s) specified for the database(s) will be integrated into the Data Guard configuration via the following steps:

* All the databases in a Data Guard configuration will be configured to send backups to the Recovery Appliance specified for the primary database. (Backups can later be scheduled from any of the databases.)

* The Recovery Appliance will be added as a member of the Data Guard broker configuration.

* Redo transport will be enabled from the primary database to the Recovery Appliance. (Although all databases in the Data Guard configuration will have the basic settings required to send redo, redo transport will only be enabled from the primary database. After the initial configuration performed by this command, redo transport settings within the Data Guard configuration can be changed via Data Guard broker commands.)

* Restrictions:

* If multiple databases in a Data Guard configuration are specified as targets, these databases will be ignored and no configuration is attempted.

* This option is not compatible with group targets. If a group target is specified on the command line or in the input file in conjunction with this option, an error will be thrown.

* Preferred database credentials and preferred database host credentials must be present for all databases. If this is not available for all databases in a Data Guard configuration, the configuration

procedure will fail and report an error for all databases in this configuration.

– CUSTOM_DG:

* One or more of the databases in one or more Data Guard configurations – the primary and some/all standbys – can be specified as targets in an input file. Each database can be designated to send backups to the same or different Recovery Appliances. The specified Recovery Appliance(s) will be integrated into the Data Guard configuration via the following steps:

  * Each specified database will be configured to send backups to its designated Recovery Appliance. (Databases in a Data Guard configuration not explicitly specified as targets will not be configured for backup.)

  * The Recovery Appliance(s) will be added as members of the associated Data Guard broker configuration.

  * If there are no redo routes currently defined for a Data Guard configuration, redo transport will be enabled from the primary database to all Recovery Appliances. If redo routes are currently defined for a Data Guard configuration, the Recovery Appliances will be added to the broker configuration but will not receive redo from any member. After the initial configuration performed by this command, the redo transport settings within the Data Guard configuration can be changed via Data Guard broker commands.

  * Regardless of which databases in a Data Guard configuration are specified, the redo transport user setting will be changed on the primary database to accommodate the integrated Recovery Appliance(s), and that setting (along with the primary database password file) will be propagated to all standby databases.

* Restrictions:

  * If a group target is specified on the command line, only targets that are actually part of that group will be configured. The primary database in a Data Guard configuration must be included in the group.

  * An individual group can only be targeted to send backups to a single Recovery Appliance. If the primary and standby databases need to send backups to different Recovery Appliances, they must be members of different groups and these groups should be specified in an input file. The primary database must be a member of one of the groups specified in the input file.

- target_name="<database or group target name>"

  Enterprise Manager target name of a single-instance database or cluster database that will be configured as a protected database, or else a group for which all member databases will be configured. (Multiple databases can be configured either by specifying a group target or by using the input_file option below.)

  – Required: Yes

  – Scope: Both

- target_type="oracle_database|rac_database|composite"

  Target type corresponding to the target specified by -target_name. Allowed target types are single-instance database (oracle_database), cluster database (rac_database), or group (composite).

  – Required: Yes

  – Scope: Both

- input_file="target_list:<full pathname of input file>"

A file containing information for multiple databases and/or group targets. This is an alternative to the -target_name parameter that can be used when there are multiple databases to be configured with one or more Recovery Appliances. The entries in the file mirror the command-line parameters.

- Required:

  * Command Line: Yes, unless -target_name is specified. (Either -target_name or -input_file must be specified.)

  * Input File: Not applicable.

- Scope: Command Line Only

The format of this file is as follows:

- target_name and target_type are required for each database or group.

- The following parameters are optional (conditionally if noted, otherwise entirely). They can be specified for some or all of the targets. If an option is not specified for a particular target, values specified on the command line for that option will be used for that target. If an option is present in both the input file and command line, the input file value overrides the command-line value.

  * ra_target_name (optional only if corresponding command line argument is specified)

  * ra_vpc_username (optional only if corresponding command line argument is specified)

  * ra_override_conn_desc

  * db_cred

  * db_host_cred

  * force_install_backup_module

  * backup_module_directory

  * wallet_location

  * force_update_wallet_loc

  * wallet_cred

  * ship_redo

  * skip_controlfile_autobackup

  * update_snapshot_controlfile_loc

  * force_restart_db

- Input file format, showing optional parameters specified across three databases:

```
target.0.target_name="<database #1 target name or group target name>"
target.0.target_type=<oracle_database|rac_database|composite>
target.0.db_cred="<database named credential for database #1 or common
credential for all databases in group target>"
target.0.db_host_cred="<database host named credential for database #1>"
target.0.ra_target_name="<target name of Recovery Appliance for which
database #1 is to be configured (or multiple databases if target_name is
group)>"
target.0.ra_vpc_username="<Recovery Appliance virtual private catalog
```

```
username for database #1>"
target.0.force_install_backup_module
target.0.wallet_location=USE_RECOMMENDED
target.0.ship_redo=YES
target.1.target_name="<database #2 target name or group target name>"
target.1.target_type=<oracle_database|rac_database|composite>
target.1.db_cred="<database named credential for database #2>"
target.1.db_host_cred=<database host named credential for database #2>
target.1.ra_target_name="<target name of Recovery Appliance for which database
#2 is to be configured (or multiple databases if target_name is group)>"
target.1.ra_vpc_username="<Recovery Appliance virtual private catalog username
for database #2>"
target.1.skip_controlfile_autobackup
target.1.force_restart_db
target.1.wallet_location="<full pathname of backup wallet location for
database #2>"
target.1.wallet_cred="<named credential containing wallet credentials for a
password-protected backup wallet for database #2>"
target.2.target_name=<database #3 target name or group target name>"
target.2.target_type=<oracle_database|rac_database|composite>
target.2.db_cred="<database named credential for database #3>"
target.2.db_host_cred=<database host named credential for database #3>
target.2.ra_target_name="<target name of Recovery Appliance for which database
#3 is to be configured (or multiple databases if target_name is group)>"
target.2.ra_vpc_username="<Recovery Appliance virtual private catalog username
for database #3>"
target.2.update_snapshot_controlfile_loc
```

- ra_target_name="<Recovery Appliance or Recovery Appliance Monitoring View target name>"

  The target name of the Recovery Appliance that the specified databases will be configured to send backups to. If the -input_file option is used, this setting can be optionally overridden for an individual database as noted above. Alternatively, a Recovery Appliance Monitoring View target can be specified, in which case the databases are configured to send backups to the Recovery Appliance associated with the Recovery Appliance Monitoring View.

  – Required: Yes

  – Scope: Both

- ra_vpc_username="<Recovery Appliance virtual private catalog username>"

  The name of the Recovery Appliance database virtual private catalog user that will be used to send backups to and ship redo to the Recovery Appliance for all specified databases. This must be a virtual private catalog user, not the Recovery Appliance administrator user.

  – Required: Yes

  – Scope: Both

- ra_override_conn_desc="<Recovery Appliance database override connect descriptor>"

  A TNS connect descriptor for the Recovery Appliance that will be used by the database to send backups and redo to the Recovery Appliance. If specified, this connect descriptor overrides the default Enterprise Manager connect descriptor for the Recovery Appliance target. The value can be a full connect descriptor, an Easy Connect string, or a TNS alias.

  – Required: No

- – Default: Use the default Enterprise Manger connect descriptor for the Recovery Appliance target.
- – Scope: Both

- db_cred="<database target named credential>"

  The name of an existing Enterprise Manager database named credential for a SYSDBA or SYSBACKUP role user that can be used to connect to all the specified target databases. If this argument is not specified, preferred credentials will be used. If multiple databases are specified in an input file, this should be a global named credential.

  - – Required: No
  - – Default: Preferred database credentials for database target type.
  - – Scope: Both

- db_host_cred="<database target host named credential>"

  The name of an existing Enterprise Manager database host named credential that can be used to run operating system commands on the specified target database hosts. The credential should be for a user that has write permission for all Oracle homes. If this argument is not specified, preferred credentials will be used. If multiple database hosts are specified in an input file, this should be a global named credential.

  - – Required: No
  - – Default: Preferred host credentials for database target type.
  - – Scope: Both

- download_backup_module

  Download the latest version of the Recovery Appliance backup module for all supported operating systems from the Oracle Cloud and upload them to the Enterprise Manager software library during the deployment procedure, so that they will then be available for deployment and installation for all databases being processed later in the procedure.

  - – Required: No
  - – Default: Do not download new backup module versions.
  - – Scope: Both

  The following conditions apply to this argument:

  - – If it is desired to install the latest available backup module version on all databases, this argument should be specified in conjunction with -force_install_backup_module.
  - – This argument should not be specified if it is desired to maintain strict control of the backup module version on all databases. In that case, the desired backup module versions can be obtained and uploaded manually to the software library using the emcli configure_db_ha -uploadBackupModule command.
  - – If this argument is specified but Enterprise Manager does not have external internet connectivity, backup module download will be skipped.

- force_install_backup_module

Force installation of the version of the Recovery Appliance backup module stored in the Enterprise Manager software library into the Oracle homes of the specified target databases, regardless of whether there is an existing backup module installed in the Oracle homes. This will overwrite any existing backup module, so this option should be selected only if it is known that the backup module version in the software library is the version desired to be installed on all the specified target databases. If this argument is not specified, the backup module will be installed only if there is no existing backup module installed in the Oracle home and there is a backup module version uploaded to the software library.

– Required: No

– Default: Do not install the backup module.

– Scope: Both

• backup_module_directory="<full pathname where backup module will be installed on database hosts>"

The directory where the backup module will be installed on the database hosts. The directory must exist on all hosts.

– Required: No

– Default: ORACLE_HOME/lib

– Scope: Both

• wallet_location=USE_RECOMMENDED|"<full pathname of backup wallet location>"

The location of an existing wallet that will be configured as the backup wallet, or if a wallet doesn't exist, the location where a new backup wallet will be created. WALLET_LOCATION in SQLNET.ORA will be set to the specified location. If USE_RECOMMENDED is specified, the wallet location will be set to the Oracle-recommended value: $ORACLE_BASE/admin/$ORACLE_UNQNAME/wallet.

– Required: No

– Default: Do not alter WALLET_LOCATION in SQLNET.ORA.

– Scope: Both

The following conditions apply to this argument:

– If -force_update_wallet_loc is not specified in conjunction with this argument, and there is an existing WALLET_LOCATION setting in SQLNET.ORA or an existing default wallet, this argument will be ignored in order to avoid perturbing other database features that use these existing settings.

– If -force_update_wallet_loc is specified in conjunction with this argument, the wallet creation/usage and WALLET_LOCATION update behavior described above will occur regardless of whether there is an existing default wallet or WALLET_LOCATION setting.

• force_crs_setenv

Set the values of the required environment variables in the database Cluster Ready Services (CRS) settings via the SRVCTL command. This argument should be specified if a wallet location using these variables is being configured (as per the above wallet handling description) and these variables are not already set in CRS. This will require a database restart.

– Required: No

– Default: Do not set the environment variables.

     – Scope: Both

- force_update_wallet_loc

  Set the WALLET_LOCATION parameter in SQLNET.ORA to the value specified by the -wallet_location argument.

  – Required: No

  – Default: Do not forcibly update WALLET_LOCATION in SQLNET.ORA.

  – Scope: Both

  The following conditions apply to this argument:

  – This argument should be specified if it is desired to change WALLET_LOCATION to the location specified by the -wallet_location argument. (In the absence of this argument, WALLET_LOCATION will be set only under the conditions described above in the description for the -wallet_location argument.)

  – This option should be used with caution, as it should be verified that the new wallet location will not perturb other database features that may be using an existing wallet.

- wallet_cred="<named credential containing wallet credentials for a password-protected wallet>"

  Named credential used to create a new password-protected wallet or to update an existing password-protected wallet with the Recovery Appliance virtual private catalog user credentials used to connect to the Recovery Appliance

  – Required: No

  – Default: Create auto-login, non-password-protected wallet or assume existing wallet is not password-protected.

  – Scope: Both

- ship_redo=YES|NO

  Enable or disable real time redo transport from all specified databases to their respective Recovery Appliances. This argument is only valid when specified with -backup_config=NO_DG. If NO is specified and redo transport is currently enabled for any databases being configured, it will be disabled for those databases.

  – Required: No

  – Default: Do not alter the current redo transport settings.

  – Scope: Both

- skip_controlfile_autobackup

  Do not enable RMAN control file autobackup for the databases. (Control file autobackup is recommended to be enabled when backing up to Recovery Appliance. If control file autobackup is already enabled for any databases, that setting is not altered.)

  – Required: No

  – Default: Enable controlfile autobackup.

  – Scope: Both

- update_snapshot_controlfile_loc

Configure the database snapshot control file location to the Fast Recovery Area if it is set, or else the same location as the database control file.

- – Required: No

- – Default: Do not alter the snapshot control file location.

- – Scope: Both

- • parallelism=<# of SBT channels to set in database RMAN settings for Recovery Appliance backups>

  Configure RMAN settings for all databases with the specified number of SBT channels for Recovery Appliance backups. This will result in the specified number of backup pieces being processed in parallel by each backup.

  - – Required: No

  - – Default: Do not alter the channel settings in the database RMAN settings.

  - – Scope: Both

- • force_restart_db

  Restart database(s) after the configuration process, regardless of whether it was required by the configuration steps actually performed. A rolling restart will be performed for Real Application Cluster (RAC) databases. This argument should be used only for individual databases in cases where it is known those databases need to be restarted. Specifying this argument for a recurring procedure will cause all the databases to which it is applied to be restarted in every execution.

  - – Required: No

  - – Default: If this option is not specified, a rolling restart will be automatically performed only if a configuration operation was performed that requires a restart.

  - – Scope: Both

- • skip_configured_dbs

  While processing the databases in the deployment procedure, skip databases that were already configured as protected databases with their respective Recovery Appliances. All such skipped databases will be listed in the deployment procedure Initialization step output. This option is only applicable for group targets. If this option is specified when scheduling a recurring configuration procedure against one or more group targets, databases that were already configured are not needlessly reconfigured in subsequent procedure executions. In this scenario, each recurring execution of the procedure will process only databases that have joined the group since the last execution.

  - – Required: No

  - – Default: Process all databases in groups.

  - – Scope: Command Line Only

- • force_serial_execution

  Run the Enterprise Manager configuration deployment procedure in serial mode, processing one database at a time.

  - – Required: No

  - – Default: Process all databases in parallel.

  - – Scope: Command Line Only

- • schedule

Schedule the deployment procedure. If this argument is not provided, the procedure will run immediately.

  – Required: No

  – Default: Schedule procedure for immediate execution.

  – Scope: Command Line Only

Sub-arguments:

  – start_time - Time when the procedure has to start execution.

    * Format should be "yyyy/MM/dd HH:mm"

  – tz - The timezone ID (optional)

  – frequency - Valid values are once/interval/weekly/monthly/yearly. (optional)

    * If frequency is set to interval then repeat has to be specified.

    * If frequency is set to weekly or monthly, days has to specified.

    * If frequency is set to yearly, both days and months have to specified.

  – repeat - Frequency with which the procedure has to be repeated. (Required only if frequency is set to interval)

  – days - Comma separated list of days. (Required only if frequency is weekly, monthly, or yearly)

    * If frequency is weekly, then valid range is 1 to 7

    * If frequency is monthly or yearly, then valid range is 1 to 30

  – months - Comma separated list of months. (Required only if frequency is yearly)

    * Valid range is 1 to 12.

  – end_time - End time for procedure executions. (optional)

    * If it is not specified, procedure will run indefinitely.

    * Format should be "yyyy/MM/dd HH:mm"

  – grace_period - Grace period in minutes(optional)

Example 1

Configure one non-Data Guard single-instance database to send backups to and ship redo to a Recovery Appliance. Use named database and host credentials.

```
    emcli configure_db_ha -configureBackuptoRA -backup_config=NO_DG -
ra_target_name="Chicago
    ZDLRA" -ra_vpc_username="rauser1" -target_name="Finance" -
target_type="oracle_database"
    -db_cred="DB_USER" -db_host_cred="DB_HOST_USER" -ship_redo=YES
```

Example 2

Configure one non-Data Guard cluster database to send backups to a Recovery Appliance without transport redo, download and force installation of the backup module in the Oracle home of each cluster database instance. Use preferred database and host credentials.

```
    emcli configure_db_ha -configureBackuptoRA -backup_config=NO_DG
ra_target_name="Chicago
```

```
        ZDLRA" -ra_vpc_username="rauser1" -target_name="Finance" -
target_type="rac_database"
        -download_backup_module -force_install_backup_module
```

Example 3

Configure multiple non-Data Guard databases specified in an input file to send backups to and ship redo to a Recovery Appliance and force the installation of the backup module in the Oracle home of each database. Use named database and database host credentials.

```
    emcli configure_db_ha -configureBackuptoRA -backup_config=NO_DG -
ra_target_name="Chicago
        ZDLRA" -ra_vpc_username="rauser1" -input_file="target_list:/tmp/dblist"  -
db_cred="DB_USER"
        -db_host_cred="DB_HOST_USER" -ship_redo=YES -force_install_backup_module
```

Example 4

Configure multiple Data Guard databases specified in an input file to send backups to a Recovery Appliance and force the installation of the backup module in the Oracle home of each database. Use named database and database host credentials. The primary database in the Data Guard configuration must be specified in the input file.

```
    emcli configure_db_ha -configureBackuptoRA -backup_config=CUSTOM_DG
        -ra_target_name="Chicago ZDLRA" -ra_vpc_username="rauser1" -
input_file="target_list:/tmp/dblist"
        -db_cred="DB_USER" -db_host_cred="DB_HOST_USER" -force_install_backup_module
```

Example 5

Configure multiple non-Data Guard databases specified in a input file to send backups to and ship redo to a Recovery Appliance, but do not force the installation of the backup module. Use global named database and host credentials. Schedule the operation for a future time.

```
    emcli configure_db_ha -configureBackuptoRA -backup_config=NO_DG -
ra_target_name="Chicago
        ZDLRA" -ra_vpc_username="rauser1" -input_file="target_list:/tmp/dblist" -
db_cred="DB_USER"
        -db_host_cred="DB_HOST_USER" -ship_redo=YES -schedule="start_time:2020/10/28
18:31;tz:PST;"
```

Example 6

Configure database members of a group target to send backups and redo to a Recovery Appliance. Use preferred credentials for all databases. Schedule the procedure to execute on a daily recurring schedule, and do not process any databases in the group that were already enrolled during previous executions of the procedure (i.e., only databases that have joined the group since the last procedure execution are processed). The first invocation of the command will work on all non-Data Guard databases in the group while the second invocation will work on all Data Guard databases in the group. The primary database in the Data Guard configuration must be a part of the group.

```
    emcli configure_db_ha -configureBackuptoRA -backup_config=NO_DG -
ra_target_name="Chicago
        ZDLRA" -ra_vpc_username="rauser1" -target_name="Backup_Group" -
target_type=composite
        -ship_redo=YES -skip_configured_dbs -schedule="start_time:2020/10/10
01:00;tz:PST;frequency:interval;repeat:1d"

    emcli configure_db_ha -configureBackuptoRA -backup_config=CUSTOM_DG
        -ra_target_name="Chicago ZDLRA" -ra_vpc_username="rauser1" -
```

```
target_name="Backup_Group"
     -target_type=composite  -skip_configured_dbs -
schedule="start_time:2020/10/10 04:00;tz:PST;frequency:interval;repeat:1d"
```

Example 7

Configure multiple non-Data Guard databases with multiple Recovery Appliances using an input file. Provide command line values for database credentials applicable to all databases, and specify that the latest backup module should be downloaded to the software library. Provide per-database values in the input file for backup module install, redo transport, wallet location, and snapshot control file settings.

```
emcli configure_db_ha -configureBackuptoRA -backup_config=NO_DG -
input_file="target_list:/tmp/dblist"
     -db_cred="DB_USER" -download_backup_module
```

The input file used in this example specifies three databases, two associated with one Recovery Appliance and one associated with a different Recovery Appliance. None of the databases are in a Data Guard configuration. The options vary across the databases as follows:

- FinanceDB

    – ZDLRA: Montreal ZDLRA

    – VPC user: rauser1

    – Redo transport: Enable.

    – Backup module: Install.

    – Wallet: Use recommended location, force update of WALLET_LOCATION in SQLNET.ORA.

    – Snapshot control file: Set to shared location.

    – Parallel backup channels: 2

- SalesDB

    – ZDLRA: Montreal ZDLRA

    – VPC user: rauser2

    – Redo transport: Disable.

    – Backup module: Do not install.

    – Wallet: Use default wallet if databases already configured with default wallet, otherwise set to ZDLRA default location; do not force WALLET_LOCATION update.

- MarketingDB

    – ZDLRA: Boston ZDLRA

    – VPC user: rauser1

    – Redo transport: Keep existing settings.

    – Backup module: Install.

    – Wallet: Use custom location, force update of WALLET_LOCATION in SQLNET.ORA.

    – Parallel backup channels: 4

The input file is as follows:

```
target.0.ra_target_name="Montreal ZDLRA"
target.0.ra_vpc_username="rauser1"
target.0.target_name="FinanceDB"
target.0.target_type="rac_database"
target.0.force_install_backup_module
target.0.ship_redo=YES
target.0.wallet_location=USE_RECOMMENDED
target.0.force_update_wallet_loc
target.0.update_snapshot_controlfile_loc
target.0.parallelism=2
target.1.ra_target_name="Montreal ZDLRA"
target.1.ra_vpc_username="rauser2"
target.1.target_name="SalesDB"
target.1.target_type="oracle_database"
target.1.ship_redo=NO
target.2.ra_target_name="Boston ZDLRA"
target.2.ra_vpc_username="rauser1"
target.2.target_name="MarketingDB"
target.2.target_type="rac_database"
target.2.force_install_backup_module
target.2.wallet_location="/prod/db/wallet/$ORACLE_UNQNAME"
target.2.force_update_wallet_loc
target.2.parallelism=4
```

Example 8

Configure all databases in one Data Guard configuration to send backups to a Recovery Appliance by specifying the primary database in this Data Guard configuration. Download and force installation of the backup module in the Oracle home of each database instance. Use preferred database and host credentials.

```
emcli configure_db_ha -configureBackupToRA -backup_config="ALL_DG"
   -ra_target_name="Chicago ZDLRA" -ra_vpc_username="rauser1"
   -target_name="OrclPrimary" -target_type="oracle_database" -download_backup_module
   -force_install_backup_module
```

Example 9

Configure multiple groups of Data Guard databases with multiple Recovery Appliances using an input file that contains multiple group targets. Schedule the procedure to execute on a daily recurring schedule, and do not process any databases in the groups that were already enrolled during previous executions of the procedure (i.e., only databases that have joined the groups since the last procedure execution are processed). Provide command line values specifying that the latest backup module should be installed for all databases. Provide per-database values in the input file for database credentials, redo transport, wallet location, and snapshot control file settings.

```
emcli configure_db_ha -configureBackupToRA -backup_config="CUSTOM_DG"
   -input_file="target_list:/tmp/dblist" -skip_configured_dbs -
download_backup_module -force_install_backup_module
   -schedule="start_time:2020/10/10 01:00;tz:PST;frequency:interval;repeat:1d"
```

Multiple Data Guard configuration databases are involved in the groups specfied in this input list. Group membership for these Data Guard databases is shown below. Data Guard databases in groups G1, G2 and G3 will be configured when this command is run with the input file shown. Data Guard databases that are standalone or in a group not included in the input file are not configured. DGConfiguration4 databases are all ignored since the primary database in this configuration is not present in one of the groups specified in the input file.

- DGConfiguration1

    – Primary DB: ORCL12 - Group G1

    – Standby DB: ORCL12Stby - Group G2

    – Standby DB: ORCLStby2 - Group G3

    – Standby DB: ORCLStby3 - standalone

- DGConfiguration2

    – Primary DB: DB19 - Group G1

    – Standby DB: DB19Stby - Group G1

    – Standby DB: DB19Stby2 - Group G2

- DGConfiguration3

    – Primary DB: DBx - Group G2

    – Standby DB: DBxStby - Group G1

    – Standby DB: DBxStby2 - Group G3

    – Standby DB: DBxStby3 - Group G4

- DGConfiguration4

    – Primary DB: DB18 - Group G4

    – Standby DB: DB19Stby - Group G1

    – Standby DB: DB19Stby2 - Group G2

The input file used in this example specifies three groups, each associated with a different Recovery Appliance. Each group is also associated with a specific set of parameter values. The options vary across the groups as follows:

- Group G1

    – ZDLRA: Montreal ZDLRA

    – VPC user: rauser1

    – Wallet: Use recommended location, force update of WALLET_LOCATION in SQLNET.ORA.

    – Snapshot control file: Set to shared location.

    – Parallel backup channels: 2

- Group G2

    – ZDLRA: Boston ZDLRA

    – VPC user: rauser1

- Group G3

    – ZDLRA: Barcelona ZDLRA

    – VPC user: rauser2

    – Wallet Use recommended location

The input file is as follows:

```
target.0.ra_target_name="Montreal ZDLRA"
target.0.ra_vpc_username="rauser1"
```

```
target.0.target_name="G1"
target.0.target_type="composite"
target.0.db_cred="DB_SYSDBA_MONTREAL"
target.0.db_host_cred="HOST_MONTREAL"
target.0.wallet_location=USE_RECOMMENDED
target.0.force_update_wallet_loc
target.0.update_snapshot_controlfile_loc
target.0.parallelism=2
target.1.ra_target_name="Boston ZDLRA"
target.1.ra_vpc_username="rauser1"
target.1.target_name="G2"
target.1.target_type="composite"
target.1.db_cred="DB_SYSDBA_BOSTON"
target.1.db_host_cred="HOST_BOSTON"
target.2.ra_target_name="Barcelona ZDLRA"
target.2.ra_vpc_username="rauser2"
target.2.target_name="G3"
target.2.target_type="composite"
target.2.db_cred="DB_SYSDBA_BARCELONA"
target.2.db_host_cred="HOST_BARCELONA"
target.2.wallet_location=USE_RECOMMENDED
```

configure_db_ha -configureRABackup

> **Note:**
>
> The `configureRABackup` sub-command will be deprecated soon. Use the sub-command `configureBackupToRA` to perform this task.

Configures one or more databases to be protected by a Recovery Appliance, with the ability to send backups and redo to the Recovery Appliance. If necessary, installs the Recovery Appliance Backup Module in the database Oracle homes. (The specified databases must be already enrolled with the Recovery Appliance as protected databases.)

> **Note:**
>
> Databases specified here must be enrolled as protected databases with the Recovery Appliance.

Format

```
emcli configure_db_ha -configureRABackup
      -ra_target_name="<Recovery Appliance target name>"
      -ra_user="<Recovery Appliance database user name>"
      (
       (-target_name="<database target name>" -target_type="oracle_database|
rac_database")
       | input_file="target_list:<full pathname of input file>"

      )
      [-db_cred="<database named credential>"]
      [-db_host_cred="<database host named credential>"]
      [-enable_redo_ship]
      [-force_backup_module_install]
      [-staging_directory="<full pathname where Backup Module will be staged on
```

```
database hosts>"]
        [-schedule=
            {
                    start_time:yyyy/MM/dd HH:mm;
                        tz:{java timezone ID};
                }]
[ ]  indicates that the parameter is optional.
```

Options

- ra_target_name

  The target name of the Recovery Appliance that the specified databases will be configured to send backups to.

- ra_user

  The Recovery Appliance database user that will be used by all the specified databases to send backups and redo to the Recovery Appliance. This must be a virtual private catalog user, not the Recovery Appliance administrator user.

- target_name

  The target name of a single-instance or cluster database that will be configured to send backups to the Recovery Appliance. A multi-database operation can be performed by using the –input_file option instead of –target_name/–target_type.

- target_type

  The type of the target specified by –target_name, either a single–instance database (oracle_database) or a cluster database (rac_database).

- input_file

  The input file that contains information for each database that is to be configured. This option is an alternative to -target_name/-target_type. The target_name and target_type lines are required for each database. The db_cred and db_host_cred lines are optional; if these are present for a database, they override the command-line named credential settings (if any) for that database. The format of the file is as follows:

  ```
  target.0.target_name=<database #1 target name>
  target.0.target_type=oracle_database|rac_database
  target.0.db_cred=<database named credential for database #1>
  target.0.db_host_cred=<database host named credential for database #1>
  target.1.target_name=<database #1 target name>
  target.1.target_type=oracle_database|rac_database
  target.1.db_cred=<database named credential for database #2>
  target.1.db_host_cred=<database host named credential for database #2>
  ```

- db_cred

  The name of an existing Enterprise Manager database named credential that can be used to connect to all the specified target databases. If this argument is not specified, preferred credentials will be used. If multiple databases are specified, this should be a global named credential.

> **Note:**
>
> If the -input_file option is used, this option can be optionally overridden for individual databases.

- db_host_cred

  The name of an existing Enterprise Manager database host credential that has been created against the Database Instance or Cluster Database target type, (rather than the Host target type) that can be used to run operating system commands on the specified target database hosts. The credential should be for a user that has write permission for all Oracle Homes. If this argument is not specified, preferred credentials will be used. If multiple databases are specified, this should be a global named credential.

> **Note:**
>
> If the -input_file option is used, this option can be optionally overridden for individual databases.

- enable_redo_ship

  Enables real-time redo transport to the Recovery Appliance from all specified target databases.

- force_backup_module_install

  Installs the version of the Recovery Appliance Backup Module stored in the Enterprise Manager Software Library into the Oracle Homes of the specified target databases, even if an existing Backup Module has already been installed in the Oracle Homes. Since this option overwrites any existing Backup Module, select this option only if the Backup Module version in the Software Library is at the same or later version than the version installed on all the specified target databases. If this flag is not specified, the Backup Module will be installed only if there is no existing Backup Module present in the Oracle Home.

- staging_directory

  The directory where the Backup Module installation files will be staged on the database hosts. This directory must exist on all hosts. A temporary subdirectory will be created in this location, then deleted after the installation of the Backup Module and/or file system backup agent is complete. The default directory is <Agent installation root>/EMStage.

- schedule

  Schedules the customized backup deployment procedure. If schedule option is not provided, the procedure will run immediately.

  – start_time: Time when the procedure has to start execution. The format should be \"yyyy/MM/dd HH:mm\"

  – tz: The timezone ID (optional)

Example 1

The following example configures a single-instance database "Finance" to send backups and ship redo to the Recovery Appliance "Chicago ZDLRA". It does not install the Backup Module in the Oracle Home of each database if the Backup Module is already present. This example uses named database and host credentials:

```
    emcli configure_db_ha -configureRABackup -ra_target_name="Chicago ZDLRA"
    -ra_user="rauser1" -target_name="Finance" -target_type="oracle_database"
    -db_cred="DB_USER" -db_host_cred="DB_HOST_USER" -
enable_redo_ship
```

Example 2

The following example configures a cluster database "Finance" to send backups to the Recovery Appliance "Chicago ZDLRA" without shipping redo. It performs a forced installation of the Backup Module in the Oracle Home of each cluster database instance. This example uses preferred database and host credentials:

```
    emcli configure_db_ha -configureRABackup -ra_target_name="Chicago ZDLRA"
    -ra_user="rauser1" -target_name="Finance" -target_type="rac_database"
    -force_backup_module_install
```

Example 3

The following example configures multiple databases specified in the input file "/tmp/dblist" to send backups and ship redo to the Recovery Appliance "Chicago ZDLRA". It performs a forced installation of the Backup Module in the Oracle Home of each database. The backup module installation files are staged in a custom directory location (/tmp/stage). This example uses named database and database host credentials:

```
    emcli configure_db_ha -configureRABackup -ra_target_name="Chicago ZDLRA"
    -ra_user="rauser1" -input_file="target_list:/tmp/dblist"
    -db_cred="NC_PDB_SYSDBA" -db_cred="DB_USER"
    -db_host_cred="DB_HOST_USER" -enable_redo_ship
    -force_backup_module_install -staging_directory="/tmp/stage"
```

Example 4

The following example configures the databases specified in the input file "/tmp/dblist" to send backups and ship redo to the Recovery Appliance "Chicago ZDLRA". It does not install the Backup Module if it is already present . The example uses global named database and host credentials. The operation is scheduled for a future time as specified:

```
emcli configure_db_ha -configureRABackup -ra_target_name="Chicago ZDLRA"
-ra_user="rauser1"  -input_file="target_list:/tmp/dblist"
-db_cred="DB_USER" -db_host_cred="DB_HOST_USER" -enable_redo_ship
-schedule="start_time:2016/06/28 18:31;tz:PST;"
```

configure_db_ha -configureCloudBackup

Configures one or more databases to send backups to Database Backup Cloud Service. If necessary, installs the Database Cloud Backup Module in the database Oracle homes.

Format

```
emcli configure_db_ha -configureCloudBackup
    (
     (-target_name="<database target name>" -target_type="oracle_database|
rac_database")
    | input_file="target_list:<full pathname of input file>"
    )
    [-cloud_account="<Database Backup Cloud Service account name>"]
    [-db_cred="<database named credential>"]
    [-db_host_cred="<database host named credential>"]
```

```
     [-force_backup_module_install]
     [-staging_directory="<full pathname where Backup Module will be staged on database
host>"]
     [-schedule=
         {
          start_time:yyyy/MM/dd HH:mm;
          tz:{java timezone ID};
         }]
[ ]  indicates that the parameter is optional.
```

Options

- target_name

  The target name of a single-instance or cluster database. A multi-database operation can be performed by using the –input_file option instead of –target_name/–target_type.

- target_type

  The type of the target specified by –target_name, either a single–instance database (oracle_database) or a cluster database (rac_database).

- input_file

  The file containing the information for each database that is to be configured. This option is an alternative to –target_name/–target_type. The target_name and target_type lines are required for each database. The db_cred and db_host_cred lines are optional; if these are present for a database, they override the command-line named credential settings (if any) for that database. The format of the file is as follows:

  ```
  target.0.target_name=<database #1 target name>
  target.0.target_type=oracle_database|rac_database
  target.0.db_cred=<database named credential for database #1>
  target.0.db_host_cred=<database host named credential for database #1>
  target.1.target_name=<database #1 target name>
  target.1.target_type=oracle_database|rac_database
  target.1.db_cred=<database named credential for database #2>
  target.1.db_host_cred=<database host named credential for database #2>
  ```

- cloud_account

  The name of a cloud account configured under the Enterprise Manager Hybrid Cloud Setup console. All the information that is needed to connect to the Backup Service will be obtained from this account and its associated Backup Service settings, including the service name, identity domain, username, password, and container (optional). If this argument is not specified, the global cloud account settings configured in the Database Backup Cloud Service Settings page will be used. (This argument is applicable only if the Hybrid Cloud Setup console has been used to configure accounts and Backup Service settings).

- db_cred

  The name of an existing Enterprise Manager database named credential that can be used to connect to all the specified target databases. If this argument is not specified, preferred credentials will be used. If multiple databases are specified, this should be a global named credential.

> **Note:**
>
> If the -input_file option is used, this option can be optionally overridden for individual databases.

- db_host_cred

  The name of an existing Enterprise Manager database host credential that has been created against the Database Instance or Cluster Database target type, (rather than the Host target type) that can be used to run operating system commands on the specified target database hosts. The credential should be for a user that has write permission for all Oracle Homes. If this argument is not specified, preferred credentials will be used. If multiple databases are specified, this should be a global named credential.

  > **Note:**
  >
  > If the -input_file option is used, this option can be optionally overridden for individual databases.

- force_backup_module_install

  Install the version of the Oracle Database Cloud Backup Module stored in the Enterprise Manager Software Library into the Oracle Homes of the specified target databases, even if an existing Backup Module has already been installed in the Oracle Homes. Since this option overwrites any existing Backup Module, select this option only if the Backup Module version in the Software Library is at the same or later version than the version installed on all the specified target databases. If this flag is not specified, the Backup Module will be installed only if there is no existing Backup Module present in the Oracle Home.

- staging_directory

  The directory where the Backup Module installation files will be staged on the database hosts. This directory must exist on all hosts. A temporary subdirectory will be created in this location, then deleted after the installation of the Backup Module and/or file system backup agent is complete. The default directory is <Agent installation root>/EMStage.

- schedule

  Schedules the customized backup deployment procedure. If schedule option is not provided, the procedure will run immediately.

  – start_time: Time when the procedure has to start execution. The format should be "yyyy/MM/dd HH:mm"

  – tz: The timezone ID (optional)

Example 1

The following example configures a single-instance database "Finance" to send backups to Database Backup Cloud Service. It does not install the Backup Module if the module is already installed. This example uses named database and host credentials:

```
emcli configure_db_ha -configureCloudBackup
-target_name="Finance" -target_type="oracle_database"
-db_cred="DB_USER" -db_host_cred="DB_HOST_USER"
```

Example 2

The following example configures a cluster database "Finance" to send backups to Database Backup Cloud Service. It performs a forced installation of the Backup Module in the Oracle Home of each cluster database instance. This example uses preferred database and host credentials:

```
emcli configure_db_ha -configureCloudBackup
-target_name="Finance" -target_type="rac_database"
-force_backup_module_install
```

Example 3

The following example configures the databases specified in the input file "/tmp/dblist" to send backups to Database Backup Cloud Service. It performs a forced installation of the Backup Module in the Oracle Home of each database. The backup module installation files are staged in a custom directory location (/tmp/stage). This example uses named database and database host credentials.

```
emcli configure_db_ha -configureCloudBackup
-input_file="target_list:/tmp/dblist"
-db_cred="DB_USER" -db_host_cred="DB_HOST_USER" -force_backup_module_install
-staging_directory="/tmp/stage"
```

configure_db_ha -installSoftware

Installs the Recovery Manager (RMAN) backup module in the Oracle Homes or installs the Recovery Appliance file system backup agent on the hosts of one or more databases.

Format

```
emcli configure_db_ha -installSoftware
    (
     (-target_name="<database target name>" -target_type="oracle_database|
rac_database")
    | input_file="target_list:<full pathname of input file>"
  )
   (
     (-install_backup_module -module_type="ra|cloud" [-force_backup_module_install]
     [-db_host_cred="<database host named credential>"])
    | (-install_fs_agent [-fs_agent_host_cred="<host named credential>"]
     [-fs_agent_install_directory="<full pathname of OSB client installation
directory>"])
   )
    [-staging_directory="<full pathname where Backup Module and file system backup
agent installation will be staged on database hosts>"]
    [-schedule=
        {
         start_time:yyyy/MM/dd HH:mm;
         tz:{java timezone ID};
        }]

[ ] indicates that the parameter is optional.
```

Options

• target_name

The target name of a single-instance or cluster database that is configured to send backups to a Recovery Appliance or Oracle Cloud. You can perform a multi-database operation by using the –input_file option instead of –target_name/–target_type.

- target_type

   The type of the target specified by –target_name, either a single–instance database (oracle_database) or a cluster database (rac_database).

- input_file

   The input file that contains information for each database that is configured. This option is an alternative to –target_name/–target_type. The target_name and target_type lines are required for each database. The db_cred and db_host_cred lines are optional; if these are present for a database, they override the command-line named credential settings (if any) for that database. The following format is used while installing a backup module with the –install_backup_module option:

```
target.0.target_name=<database #1 target name>
target.0.target_type=oracle_database|rac_database
target.0.db_host_cred=<database host named credential for database #1>
target.1.target_name=<database #1 target name>
target.1.target_type=oracle_database|rac_database
target.1.db_host_cred=<database host named credential for database #2>
```

   The following format is used while installing the file system backup agent with the install_fs_backup_agent:

```
target.0.target_name=<database #1 target name>
target.0.target_type=oracle_database|rac_database
target.0.fs_agent_host_cred=<host named credential for the host(s) of
database #1>
target.1.target_name=<databse #1 target name>
target.1.target_type=oracle_database|rac_database
target.1.fs_agent_host_cred=<host named credential for the host(s) of
database #2>
```

- install_backup_module

   Installs the version of the Backup Module (of the type specified by the –module_type argument) stored in the Enterprise Manager software library in the Oracle Homes of the specified target databases. By default, the module will be installed if there is no existing module in the database Oracle Home, unless the –force_backup_module_install flag is specified. This module cannot be specified in combination with –install_osb_client.

- module_type

   The type of RMAN backup module to be installed. The possible values are: ra – Recovery Appliance Backup Module and cloud – Oracle Database Cloud Backup Module

- force_backup_module_install

   Installs a version of the Recovery Appliance Backup Module stored in the Enterprise Manager Software Library into the Oracle Homes of the specified target databases, even if an existing Backup Module has already been installed in the Oracle Homes. Since this option overwrites any existing Backup Module, select this option only if the Backup Module version in the Software Library is at the same or later version than the version installed on all the specified target databases. If

this flag is not specified, the Backup Module will be installed only if there is no existing Backup Module present in the Oracle Home.

- db_host_cred

  The name of an existing Enterprise Manager database host credential that has been created against the Database Instance or Cluster Database target type, (rather than the Host target type) that can be used to run operating system commands on the specified target database hosts. The credential should be for a user that has write permission for all Oracle Homes. If this argument is not specified, preferred credentials will be used. If multiple databases are specified, this should be a global named credential.

  > **Note:**
  >
  > If the -input_file option is used, this option can be optionally overridden for individual databases.

- install_fs_backup_agent

  Installs the file system backup agent on all hosts of the specified target databases.

- fs_agent_host_cred

  The name of an existing Enterprise Manager host privileged named credential (not a database host credential) that can be used to perform the file system backup agent installation on all the hosts of the specified target databases. The credential should specify the root user or a user with root privilege delegation. If this argument is not specified, preferred credentials for the host targets associated with the databases is used. If multiple databases are specified, this should be a global named credential.

- fs_agent_install_directory

  The directory where the file system backup agent is installed on all database hosts. The directory must exist on all the hosts. The default directory is /usr/local/oracle/backup.

- staging_directory

  The full place-name where Backup Module and file system backup agent installation will be staged on database hosts.

- schedule

  Schedules the customized backup deployment procedure. If schedule option is not provided, the procedure will run immediately.

  - start_time: Time when the procedure has to start execution. The format should be "yyyy/MM/dd HH:mm"

  - tz: The timezone ID (optional)

Example 1

The following example installs the Recovery Appliance Backup in the Oracle Home of one single-instance database "Finance". It does not install the Backup Module if the module is already installed. This example uses named database and host credentials:

```
emcli configure_db_ha -installSoftware
-target_name="Finance" -target_type="oracle_database" -db_host_cred="DB_HOST_USER"
-install_backup_module -module_type="ra"
```

Example 2

The following example installs the Oracle Database Cloud Backup Module in the Oracle Home of one RAC database "Finance". It performs a forced installation if the module is already installed. This example uses named database host credentials:

```
emcli configure_db_ha -installSoftware
-target_name="Finance" -target_type="rac_database"
-install_backup_module -module_type="cloud" -force_backup_module_install -
db_host_cred="DB_HOST_USER"
```

Example 3

The following example installs the Recovery Appliance file system backup agent on all cluster database nodes of one cluster database "Finance" in a non-default location. This example uses privileged host credentials:

```
emcli configure_db_ha -installSoftware
-target_name="Finance" -target_type="rac_database"
-install_osb_client -osb_install_directory="/usr/local/osb12"
-osb_host_cred="HOST_PRIV_CRED"
```

Example 4

The following example installs the Recovery Appliance Backup Module in the Oracle Homes of multiple databases specified in the input file "/tmp/dblist". It performs a forced installation if the module is already installed. This example uses preferred database host credentials. The operation scheduled for a future time as specified:

```
emcli configure_db_ha -installSoftware
-input_file="target_list:/tmp/dblist" -install_backup_module
-module_type="ra" -force_backup_module_install
-schedule="start_time:2016/06/28 18:31;tz:PST;"
```

Example 5

The following example installs the Recovery Appliance file system backup agent on all hosts of multiple databases specified in an input file in the default location. This example uses preferred privileged host credentials. The file system backup agent installation files are staged in a custom staging area location:

```
emcli configure_db_ha -installSoftware
-install_fs_backup_agent -staging_director="tmp/stage"
```

configure_db_ha -uploadFSBackupAgentInstall

Uploads Recovery Appliance file system backup agent installation media into the Enterprise Manager Software Library. This installation media is used by the -installSoftware subcommand when it is run with the -install_fs_backup_agent option to push a file system backup agent installation to one or more hosts. Separate software library components will be created according to the platform of the uploaded installation media.

Format

```
-media_location="<pathname of installation media zip file>"
```

Option

• media_location

The local path-name of the file system backup agent installation media zip file.

Example

Uploads the Recovery Appliance file system backup agent installation media for the Linux 64 platform:

```
emcli configure_db_ha -uploadFSBackupAgentInstall
-media_location="/home/osb_shiphome/releases/osb-12.1.0.2/
osb_12.1.0.2.0_linux.x64_release.zip"
```

configure_db_ha -uploadBackupModule

Uploads Cloud or Recovery Appliance Backup Modules into the Enterprise Manager Software Library. Separate software library components will be created according to the platform of the uploaded Backup Module.

Format

```
emcli configure_db_ha -uploadBackupModule
        -module_location="<comma separated list of pathnames of Backup Module zip
files>"
        -module_type="ra|cloud"
```

Options

- module_location

  ```
  <comma separated list of pathnames of Backup Module zip files>
  ```

  The list of local pathname of the Backup Module zip file.

- module_type

  The type of RMAN backup module to be uploaded. The possible values: ra: Recovery Appliance Backup Module and cloud: Oracle Database Cloud Backup Module.

Example 1

The following example uploads the Cloud Backup Module for multiple platforms:

```
emcli configure_db_ha -uploadBackupModule -module_type="cloud"
-module_location="/home/opc_linux64.zip,/home/opc_zlinux64.zip"
```

Example 2

The following example uploads the Recovery Appliance Backup Module for the Linux 64 platform:

```
emcli configure_db_ha -uploadBackupModule -module_type="ra"
-module_location="/home/ra_linux64.zip"
```

config_db_service_target

Creates a Database as a Service (DBaaS) target for Oracle Public Cloud.

Format

```
emcli config_db_service_target
     -database_unique_name="database unique name"
     -service_grp_name="service group name"
     -cloud_service_name="cloud service name"
     -operation="operation to be performed"
     -schema_name="schema name"
     -tablespace_name="tablespace name"
     [-subscription_id="subscription ID"]
```

```
[-customer_name="customer name"]
[-csi_number="CSI number"]
[-connection_service_name="connection service name"]
[-cloud_service_version="cloud service version"]
[-l_o_b="line of business"]
```

Options

- **database_unique_name**

  Identifies the `DBName` property of a database target on which the DBaaS target will be based.

- **service_grp_name**

  Identifies the DBaaS target service group name. The DBaaS target name will be `service_grp_name_cloud_service_name`.

- **cloud_service_name**

  Identifies the Oracle Public Cloud service name.

- **operation**

  Identifies the operation to be performed on the DBaaS target (for example, "create").

- **schema_name**

  Identifies the name of the schema associated with the DBaaS target.

- **tablespace_name**

  Identifies the name of the tablespace associated with the DBaaS target.

- **subscription_id**

  Provides a value for the Cost Center property of a DBaaS target.

- **customer_name**

  Provides a customer point of contact for the DBaaS target.

- **csi_number**

  Identifies the Customer Support Identifier (CSI) of the DBaaS target.

- **connection_service_name**

  Identifies the name of the Database Service associated with the DBaaS Target.

- **cloud_service_version**

  Shows the Oracle Public Cloud Service version of the DBaaS target.

- l_o_b

  Identifies the Line of Business (LOB) of the DBaaS target.

Exit Codes

```
0 On success
Non-zero value means verb processing was not successful.
```

Example

Creates a new Database as a Service (DBaaS) target (`db_serv1`). Specifies the schema, tablespace name, and connection service name. The new DBaaS target is based on a Database target using `db.example.com` as the `DBName` property:

```
emcli config_db_service_target
     -database_unique_name="db.example.com"
     -service_grp_name="db"
     -cloud_service_name="serv1"
     -operation="Create"
     -schema_name="HR"
     -tablespace_name="SYSTEM"
     -connection_service_name="nservice1"
```

configure_log_archive_locations

Configures Log Archive Locations for the root target and its children. To configure Log Archive Location for a target, you should know the configuration parameters, like host name, from where the log archive files are accessible, the credentials to access the host, and the location of the log archive files.

Format

```
emcli configure_log_archive_locations
     -root_target_name="<target_name>"
     -root_target_type="<target_type>"
     [-archive_config_file="<target_archive_config_file_location>"]
     [-no_update]
     [-debug]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- root_target_name

  Name of the root target. A configurable tree target hierarchy will be created with this root target. Example root targets are WebLogic Domain and Fusion Application Instance.

- root_target_type

  Target type of the root target name.

- archive_config_file

  Location of the archive config file. Every line in this file should contain the following 7 fields in the same order.

  target_name,target_type,host_target_name,host_cred_type,host_cred_name or new_cred_user_name,new_cred_password,archive_dir_location

  – target_name

    If this target is part of the root hierarchy, then this target and it's children will be updated with the archive parameters specified.

  – target_type

    Target type of the above target.

  – host_target_name

Host name from where archive location is accessible. The Management Agent monitoring this target should have Oracle Fusion Middleware plug-in release 12.1.0.4 or later installed.

– host_cred_type

Credential type. Possible values are preferred_credentials or named_credentials or new_credentials.

– host_cred_name

Credential set name for preferred_credentials type or Named credential name for named_credentials type.

– new_cred_user_name

New credential user name for new_credentials type.

– new_cred_password

New credential password for new_credentials type.

– archive_dir_location

Directory location where log archive files available.

• no_update

If this flag is provided, targets which are already configured with the archive properties, will not be updated again.

• debug

Runs the verb in verbose mode for debugging purposes.

Example

The following example configures Log Archive Locations for Fusion Instance target and its children.

```
 emcli configure_log_archive_locations
        -root_target_name=fa1
        -root_target_type=fusion_apps_instance
        -archive_config_file=/scratch/config.txt
```

Sample Archive Config File:

In the case of new credentials:

```
fa1,fusion_apps_instance,adc123.oracle.com,new_credentials,user1,pwd1,/
scratch/fa1
```

In the case of preferred credentials:

```
fa1,fusion_apps_instance,adc123.oracle.com,preferred_credentials,credential_set1,
,/scratch/fa1
```

(Because this is preferred credentials, ',,' means the new_cred_password field is not valid and therefore skipped.)

configure_siteguard_lag

Configures the limit of Apply lag and Transport lag for all or selected databases of the system.

Format

```
emcli configure_siteguard_lag
        [-system_name="name_of_the_system"]
        [-target_name="name_of_the_target_database"]
        [-property_name="lag_type"]
        [-value="max_limit_in_seconds"]
```

[ ] indicates that the parameter is optional.

Options

*   **system_name**

    Name of the system on which lag limits need to be configured.

*   **target_name**

    Name of the database on which lag limits need to be configured.

*   **property_name**

    Name of the lag property to be configured. Valid values are `ApplyLag` and `TransportLag`.

*   **value**

    Limit of the lag. These values are specified in seconds.

Example

This example configures the Apply lag limit of 1000 seconds on all of the databases of `austin-system`:

```
emcli configure_siteguard_lag
        -system_name="austin-system"
        -property_name="ApplyLag"
        -value="1000"
```

confirm_instance

Confirms a manual step. An instance cannot be confirmed when its status is suspended, stopped, completed, or completed with an error.

Format

```
emcli confirm_instance
      [-instance=<instance_guid>]
      [exec=<execution_guid>]
      [-name=<execution name>]
      [-owner=<execution owner>]
      -stateguid=<state_guid>
```

[ ] indicates that the parameter is optional

Options

*   **instance**

    Instance GUID.

*   **exec**

Execution GUID.

- **name**

  Execution name.

- **owner**

  Execution owner.

- **stateguid**

  Comma-separated list of state GUIDs.

Examples

```
emcli confirm_instance -instance=16B15CB29C3F9E6CE040578C96093F61 -
stateguid=51F762417C4943DEE040578C4E087168

emcli confirm_instance -instance=16B15CB29C3F9E6CE040578C96093F61 -
stateguid='51F762417C4943DEE040578C4E087168,51F762417C4944DEE040578C4E087168'
```

continue_add_host

Performs resume/continue operations of a previously submitted add host session that has failed at some phase.

Format

```
emcli continue_add_host
      -session_name="session_name"
      -continue_all_hosts | -continue_ignoring_failed_hosts"
      [-wait_for_completion]

[ ]  indicates that the parameter is optional
```

Options

- **session_name**

  Name of the session you want to continue to the next phase of Agent deployment.

- **continue_all_hosts**

  Continues the session on all hosts, including those on which the current deployment phase failed.

- **continue_ignoring_failed_hosts**

  Continues the session for only the hosts on which the current deployment phase succeeded.

- **wait_for_completion**

  Specifies whether the command should run in synchronous or asynchronous mode. If you specify this option (for synchronous mode), the command waits until the add host session completes before returning control to you on the command line.

Examples

Example 1

This example continues the session
'ADD_HOST_SYSMAN_Dec_17_2012_2:02:28_AM_PST' to the next phase of deployment
on all hosts.

```
emcli continue_add_host -session_name='ADD_HOST_SYSMAN_Dec_17_2012_2:02:28_AM_PST' -
continue_all_hosts
```

Example 2

This example continues the session
'ADD_HOST_SYSMAN_Dec_17_2012_2:02:28_AM_PST' synchronously to the next phase
of deployment only on hosts on which the current phase was successful.

```
emcli continue_add_host -session_name='ADD_HOST_SYSMAN_Dec_17_2012_2:02:28_AM_PST' -
continue_ignoring_failed_hosts -wait_for_completion
```

convert_to_cluster_database

Converts a single-instance database to a Real Application Cluster (RAC) database.

Format

```
emcli convert_to_cluster_database
      -sourceTargetName="Single instance database target to be converted to RAC"
      -sysdbaCreds="Named credentials for SYSDBA user"
      -hostCreds="Named credentials for Host"
      [-newOracleHome="RAC Oracle Home, if moving to differnt home"]
      [-racConfigType="ADMIN_MANAGED | POLICY_MA NAGED"]
      [-nodeList="Comma-separated node names for Admin Managed RAC database"]
      [-serverPoolList="Comma-separated list of server pools for Policy Managed
         database"]
      [-databaseArea="Shared storage location for database files"]
      [-recoveryArea="Shared storage location for recovery files"]
      [-listenerPort="RAC Listener port"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **sourceTargetName**

  Enterprise Manager target name of the single-instance database to be converted to a
  RAC database. Database versions 10.2.0.1.0 and above are supported for conversion.
  The single-instance database target should exist on one of the nodes of the cluster where
  the RAC database will be created, and the cluster should be an Enterprise Manager
  target.

- **sysdbaCreds**

  Named database credentials with SYSDBA privileges on the database to be converted to
  a RAC database.

- **hostCreds**

  Named host credentials of the user who owns the Oracle home installation.

- **newOracleHome**

  RAC Oracle home location of the converted database. You only need to provide this if
  different from the Oracle home of the single-instance database to be converted.

- **racConfigType**

RAC configuration type. Valid values are POLICY_MANAGED and ADMIN_MANAGED. POLICY_MANAGED is valid only for database versions 11.2 or higher. The default is ADMIN_MANAGED if not provided.

- **nodeList**

    List of valid node names for an ADMIN_MANAGED RAC database. It should include the node where the single instance database to be converted exists. If not provided, all the nodes in the cluster are used.

- **serverPoolList**

    Comma-separated list of server pool names for a POLICY_MANAGED RAC database. Applicable only for database versions 11.2 or higher.

- **databaseArea**

    New location for data files of the RAC database. This location should be shared across the nodes of the cluster. It can either be a Cluster File System location or an Automatic Storage Management diskgroup. If not specified, the existing database files should already be on shared storage, and files are not moved during RAC conversion.

- **recoveryArea**

    Fast recovery area location of the RAC database. This location should be shared across the nodes of the cluster. It can either be a Cluster File System location or an Automatic Storage Management diskgroup. If not specified, the existing recovery area location should already be on shared storage, and it does not change during RAC conversion.

- **listenerPort**

    Port of the new RAC listener to be created for the new RAC database. If not provided, the existing listener is used. This option is only applicable to 10.2 and 11.1 database versions. For 11.2 or higher database versions, this value is ignored and the RAC database is always registered with the existing listener in the Cloud Infrastructure home.

Examples

Example 1

```
emcli convert_to_cluster_database -sourceTargetName=sidb
 -sysdbaCreds=sysCreds  -hostCreds=hostCreds racConfigType=ADMIN_MANAGED
```

Example 2

```
emcli convert_to_cluster_database -sourceTargetName=sidb
 -sysdbaCreds=sysCreds  -hostCreds=hostCreds racConfigType=POLICY_MANAGED
 -serverPoolList=sp1,sp2 -databaseArea=+DATA -recoveryArea=+RECOVERY
```

create_aggregate_service

Defines an aggregate service: name and its sub-services. After the aggregate service is created, you can edit it from the Enterprise Manager Cloud Control console to configure performance and usage metrics to be collected and displayed.

Format

```
emcli create_aggregate_service
      -name='name'
```

```
-type='type'
-availType=SUB-SERVICE|SYSTEM|TESTS
-add_sub_services="name1:type1;name2:type2;..."
-avail_eval_func=and|or
[-hostName=<host_name>
[-agentURL=<agent_url>
[-properties='pname1|pval1;pname2|pval2;...']
[-timezone_region=<gmt_offset>]
[-systemname=<system_name>]
[-systemtype=<system_type>]
[-keycomponents='keycomp1name:keycomp1type;keycomp2name:keycomp2type;...']
[-beacons='bcn1name:bcn1isKey;bcn2name:bcn2isKey;...']
[-input_file='template:Template_file_name;[vars:Variables_file_name]']
[-sysAvailType=<availability_type>]
```

[ ] indicates that the parameter is optional

Options

* **name**

  Aggregate service name.

* **type**

  Aggregate service type.

* **availType**

  Sets availability to either sub-service, system-based, or test-based. Valid values are SUB-SERVICE, SYSTEM, and TESTS.

  If availability is set to SYSTEM, -systemname and -systemtype are required.

  If availability is set to TESTS, -beacons, template file, and variables are required.

* **add_sub_services**

  Sub-services list.

* **avail_eval_func**

  Operator to evaluate availability. If "and" is used, all sub-services, tests, and system-components must be up in order for this aggregate_service to be up. If "or" is used, only one of the sub-services, tests, and system-components needs to be up for this aggregate_service to be up.

* **hostName**

  Network name of the system running the Management Agent that is collecting data for this target instance.

* **agentURL**

  URL of the Management Agent that is collecting data for this target instance. If you enter the host name, the Agent URL of the host is automatically entered in this field.

* **properties**

  Name-value pair (that is, prop_name|prop_value) list of properties for the service instance.

* **timezone_region**

  Accepts either long formats ("America/Los Angeles") or short formats ("PST"). If you do not provide a time zone, the default OMS time zone is used.

- **systemname**

  System name on which service resides.

- **systemtype**

  Use emcli get_targets to obtain the system type for the system name.

- **keycomponents**

  Name-type pair (that is, `keycomp_name:keycomp_type`) list of key components in the system that are used for the service.

- **beacons**

  Name-isKey pairs that describe the beacons of the service. If isKey is set to `Y`, the beacon is set as a key-beacon of the service. The service should have at least one key beacon if the availability is set to test-based.

- **input_file**

  Template file name is the XML file that includes the template definition. Variable file defines the values for the template.

  For more information about the input_file option, see -input_file Syntax Guidelines.

  See below for an example of an XML file for this option.

- **sysAvailType**

  Type of availability when the availType is system-based. Sets the availability to either SYSTEM_TARGET_DIRECTLY or SELECTED_COMPONENTS_OF_A_SYSTEM.

  If availability is set to SYSTEM_TARGET_DIRECTLY, the system needs to have availability[status] defined. -systemname and -systemtype are required options.

  If availability is set to SELECTED_COMPONENTS_OF_A_SYSTEM, -systemname, -systemtype and -keycomponents are required options.

  If availability is set to SYSTEM_TARGET_DIRECTLY and if the system does not have availability[status] defined, the availability set is invalid. Therefore, the only option that can be set is SELECTED_COMPONENTS_OF_A_SYSTEM.

Example

```
emcli create_aggregate_service -name="My_Name"
            -type="aggregate_service"
            -add_sub_services="sub1:type1;sub2:type2"
            -avail_eval_func="and"
            -availType="SUB_SERVICE"
             -properties="prop1|value1;prop2|value2"
            -timezone_region="PST"
```

XML for input_file Example

The following sample XML file creates a service test of name 'EM Console Service Test' and of type 'Web Transaction'. It defines some properties, such as readTimeout, Collection Interval, and so forth under the <properties> section, which are related to this service test.

This service test has defined step information under <mgmt_bcn_step_with_props>. The name of the step is '1.Access Logout page'. The URL to be monitored under this

step is https://myhost.in.domain.com:14513/em/console/logon/logoff?event=load, which is defined under the properties section of the step.

This XML file also defines some threshold levels for this service test on the transaction level under <txn_thresholds>. For the metric avg_response_time, it states that if the metric value is greater than 6000.0, raise a warning alert, and if the metric value is greater than 12000.0, raise a critical alert.

```
<?xml version = '1.0' encoding = 'UTF-8'?> <transaction-template
template_type="aggregate_service" xmlns="template">
<variables>
<variable name="HOST1" value="myhost.in.domain.com"/>
<variable name="PORT1" value="14513"/>
<variable name="PROTOCOL1" value="https"/>
</variables>
<transactions>
<mgmt_bcn_transaction>
<mgmt_bcn_txn_with_props>
<mgmt_bcn_txn description="Test for checking the availability of EM Console/Website"
is_representative="true" name="EM Console Service Test" monitoring="true"
txn_type="HTTP"/>
<properties>
<property name="readTimeout" num_value="120000.0" prop_type="2" encrypt="false"/>
<property name="certValidationMode" string_value="1" prop_type="1" encrypt="false"/>
<property name="maxDownloadSize" num_value="1.0E8" prop_type="2" encrypt="false"/>
<property name="sensitiveValuesProtection" string_value="0" prop_type="1"
encrypt="false"/>
<property name="failureStringModes" string_value="regularText" prop_type="1"
encrypt="false"/>
<property name="UserAgent" string_value="Mozilla/4.0 (compatible; MSIE 6.0; Windows NT
5.1) OracleEMAgentURLTiming/3.0" prop_type="1" encrypt="false"/>
<property name="successStringModes" string_value="regularText" prop_type="1"
encrypt="false"/>
<property name="variablesModes" string_value="urlEncode" prop_type="1"
encrypt="false"/>
<property name="content" string_value="0" prop_type="1" encrypt="false"/>
<property name="AcceptLanguage" string_value="en" prop_type="1" encrypt="false"/>
<property name="connectionTimeout" num_value="120000.0" prop_type="2"
encrypt="false"/>
<property name="useCache" string_value="yes" prop_type="1" encrypt="false"/>
<property name="stringValidationMode" string_value="1" prop_type="1" encrypt="false"/>
<property name="granularity" string_value="transaction" prop_type="1"
encrypt="false"/>
<property name="numThreads" num_value="4.0" prop_type="2" encrypt="false"/>
<property name="retries" num_value="1.0" prop_type="2" encrypt="false"/>
<property name="timeout" num_value="300000.0" prop_type="2" encrypt="false"/>
<property name="retryInterval" num_value="5000.0" prop_type="2" encrypt="false"/>
</properties>
<per_bcn_properties/>
</mgmt_bcn_txn_with_props>
<steps_defn_with_props>
<mgmt_bcn_step_with_props>
<mgmt_bcn_step step_number="1" name="1.Access Logout page" step_type="HTTP"/>
<properties>
<property name="req_mode" num_value="1.0" prop_type="2" encrypt="false"/>
<property name="http_method" string_value="G" prop_type="1" encrypt="false"/>
<property name="url" string_value="{PROTOCOL1}://{HOST1}:{PORT1}/em/console/logon/
logoff?event=load" prop_type="1" encrypt="false"/>
</properties>
</mgmt_bcn_step_with_props>
</steps_defn_with_props>
```

```
<stepgroups_defn/>
<txn_thresholds>
<mgmt_bcn_threshold warning_threshold="6000.0" warning_operator="0"
critical_threshold="12000.0" critical_operator="0" num_occurrences="1">
<mgmt_bcn_threshold_key metric_name="http_response"
metric_column="avg_response_time"/>
</mgmt_bcn_threshold>
<mgmt_bcn_threshold warning_threshold="0.0" warning_operator="1"
critical_threshold="0.0" critical_operator="1" num_occurrences="1">
<mgmt_bcn_threshold_key metric_name="http_response" metric_column="status"/>
</mgmt_bcn_threshold>
</txn_thresholds>
<step_thresholds/>
<stepgroup_thresholds/>
</mgmt_bcn_transaction>
</transactions>
</transaction-template>
```

create_assoc

Creates target association instances.

Format

Standard Mode

```
emcli create_assoc
      -assoc_type="association type"
      -source="source"
      -dest="destination_target"
      [-separator="separator:attribute_name:character"]
      [-subseparator="subseparator:attribute_name:character"]
```

Interactive (Script) Mode

```
create_assoc(
      assoc_type="association type"
      ,source="source"
      ,dest="destination_target"
      [,separator="separator:attribute_name:character"]
      [,subseparator="subseparator:attribute_name:character"]
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **source_type**

  Source target type.

- **source**

  Source target.

- **dest**

  Destination target.

- **separator**

  By default, multi-value input attributes use a semicolon (;) as a separator.
  Specifying this option overrides the default separator value.

Example: `separator="<attribute_name=sep_char>"` where *attribute_name* is name of the attribute for which you want to override the separator character, and *sep_char* is the new separator character.Example: `separator="att=#"`

- **subseparator**

  By default, multi-value input attributes use a colon (:) as a subseparator. Specifying this option overrides the default subseparator value.

  Example: `subseparator="<attribute_name=sep_char>"` where *attribute_name* is name of the attribute for which you want to override the separator character and *sep_char* is the new subseparator character.Example: `separator="att=#"`

> **✎ Note:**
>
> The name and owner options must be used together.

Exit Codes

0 indicates that the verb processing was successful.

Non-zero values indicate that the verb processing was not successful.

Example

This example creates an association of type cluster_contains from target "abc_cluster:cluster" to targets "def.oracle.com:host" and "ghi.oracle.com:host":

```
emcli create_assoc
      -assoc_type="cluster_contains"
      -source="abc_cluster:cluster"
      -dest="def.oracle.com:host;ghi.oracle.com:host"
```

For a list of allowed pairs, enter `emcli list_allowed_pairs` .

create_blackout

Creates a scheduled blackout to suspend any data collection activity on one or more monitored targets.

Format

```
emcli create_blackout
      -name="name"
      -add_targets="name1:type1;name2:type2;..."...
      -reason="reason"
      [-description="description"]
      [-notification_only]
      [-is_sla_required]
      [-jobs_allowed]
      [-propagate_targets]
      [-full_blackout_all_hosts]
      [-dep_services_all]
      [-exclude_types]
      [-exclude_target]
      -schedule=
          [frequency:once|interval|weekly|monthly|yearly];
```

```
                        duration:[HH...][:mm...];
                        [start_time:yy-MM-dd HH:mm];
                        [end_time:yy-MM-dd HH:mm];
                        [repeat:#m|#h|#d|#w];
                        [months:#,#,...];
                        [days:#,#,...];
                        [tzinfo:specified|target|repository]
                        [tzoffset:#|[-][HH][:mm]]
                        [tzregion:...]
```

```
[ ]  indicates that the parameter is optional
```

Constraints on schedule arguments:

```
frequency:once
    requires => duration or end_time
    optional => start_time, tzinfo, tzoffset
frequency:interval
    requires => duration, repeat
    optional => start_time, end_time, tzinfo, tzoffset
frequency:weekly
    requires => duration, days
    optional => start_time, end_time, tzinfo, tzoffset
frequency:monthly
    requires => duration, days
    optional => start_time, end_time, tzinfo, tzoffset
frequency:yearly
    requires => duration, days, months
    optional => start_time, end_time, tzinfo, tzoffset
```

Options

- **name**

  Name of the blackout to create.

- **add_targets**

  Targets to add to the blackout, each specified as `target_name:target_type`. You can specify this option more than once.

- **reason**

  Reason for the blackout. If you have `SUPER_USER` privileges (you are an Enterprise Manager Super Administrator), any text string can be used for the reason. The reason is added to the list of allowable blackout reasons if it is not already in the list. If you do not have `SUPER_USER` privileges, you must specify one of the text strings returned by the `get_blackout_reasons` verb.

- **description**

  Description or comments pertaining to the blackout. The description, limited to 2000 characters, can be any text string.

- **notification_only**

  When this option is specified, by default a notification blackout for planned maintenance is created on the selected targets. Blackout duration is excluded from Availability(%) calculations.

- is_sla_required

  When this option is specified, the notification blackout is created for unplanned maintenance. Blackout duration is considered for Availability(%) calculations.

- **jobs_allowed**

  When you specify this option, jobs are allowed to run against blacked-out targets during the blackout period. If you do not specify this option, jobs scheduled to be run against these targets are not allowed to run during the blackout period. After a blackout has been created, you cannot change the "allowed jobs" from either EM CLI or the Enterprise Manager Cloud Control console.

- **propagate_targets**

  When you specify this option, a blackout for a target of type "host" applies the blackout to all targets on the host, including the Agent. This is equivalent to nodelevel in the emctl command. Regardless of whether you specify this option, a blackout for a target that is a composite or a group applies the blackout to all members of the composite or group.

- **full_blackout_all_hosts**

  When this option is specified, full blackout is enabled on all hosts included in this blackout. A full blackout places the host and all targets on the host (including the agent) under blackout. The propagate_targets option is implicitly enabled on selecting this option.

- **dep_services_all**

  When this option is specified, all of the dependent targets of the targets selected for blackout will also be blacked out.

- **exclude_types**

  A list of target types can be specified. Indirect members of that type and their members will not be part of the blackout. For example, specifying `oracle_dbsys` will exclude database systems and their members which would be otherwise indirect members of the blackout. Flags `exclude_targets` and `exclude_types` can be used in combination with one another.

- **exclude_target**

  A list of indirect member targets to exclude from the blackout can be specified. Indirect members of the blackout and their members will not be part of the blackout. For example, specifying a database system target will exclude that target and the corresponding database instance from the blackout if it would otherwise be an indirect member of the blackout. Flags `exclude_targets` and `exclude_types` can be used in combination with one another.

- **schedule**

  Blackout schedule. Note that the "frequency" argument determines which other arguments are required or optional.

- **schedule=frequency**

  Type of blackout schedule (default is "once").

- **schedule=duration**

  Duration in hours and minutes of the blackout (-1 means indefinite). Hours and minutes each can be up to 6-digits long.

- **schedule=start_time**

  Start date/time of the blackout. The default value is the current date/time. The format of the value is "yy-MM-dd HH:mm", for example: "2003-09-25 18:34"

- **schedule=end_time**

Last date/time of the blackout. When "frequency" is weekly, monthly, or yearly, only the date portion is used. When "frequency" is interval or once, the date and time are taken into account. The format of the value is "yy-MM-dd HH:mm"; for example: "2003-09-25 18:34"

- **schedule=repeat**

  Time between successive start times of the blackout. The letter following the number value represents the time units: "m" is minutes, "h" is hours, "d" is days, and "w" is weeks.

- **schedule=months**

  List of integer month values in the range 1-12. Each value must have a corresponding "day" value to fully specify (month, day) pairs that indicate the blackout starting days of the year.

- **schedule=days**

  When "frequency" is weekly, this is a list of integer day-of-week values in the range 1-7 (1 is Sunday). When "frequency" is monthly, this is a list of integer day-of-month values in the range 1-31 or -1 (last day of the month). When "frequency" is yearly, this is a list of integer day-of-month values in the range 1-31 or -1 (last day of the month); in this case, the month is taken as the corresponding "month" value for each (month, day) pair.

- **schedule=tzinfo**

  Type of timezone. The `tzinfo` argument is used in conjunction with `tzoffset`. Available timezone types are: "specified" (offset between GMT and the target timezone), "target" (timezone of the specified target), and "repository" (repository timezone -- default setting when tzinfo is not specified). See `-schedule=tzoffset` for more information.

- **schedule=tzoffset**

  Value of the timezone. When the `tzinfo` argument is not specified or is "repository", the timezone value is the repository timezone. In this case, the `tzoffset` argument must not be specified. Otherwise, the `tzoffset` argument is required. When `tzinfo` is set to "specified", the `tzoffset` argument specifies the offset in hours and minutes between GMT and the timezone. When `tzinfo` is set to "target", the `tzoffset` argument specifies an integer index (the first is 1) into the list of targets passed as arguments. For example, for a `tzoffset` setting of 1, the timezone of the first target specified in the `-add_targets` option is used.

  Note that the timezone is applied to the start time and the end time of the blackout periods. The timezones associated with each target are not taken into account when scheduling the blackout periods (except that when `tzinfo` is set to "target", the specified target's timezone is used for the blackout times).

- **schedule=[tzregion:<...>]**

  Time zone region to use. When you "specify" the tzinfo option, this option determines which timezone to use for the blackout schedule. Otherwise, it is ignored. It defaults to "GMT".

Examples

Example 1

This example creates blackout `b1` for the specified target (`database2`) to start immediately and last for 30 minutes.

```
emcli create_blackout
      -name=b1
      -add_targets=database2:oracle_database
      -schedule="duration::30"
      -reason="good reason1"
```

Example 2

This example creates blackout `b1` for the specified targets (`database2` and `database3`) to start at 2007-08-24 22:30 and last for 30 minutes. The timezone is the timezone for the `database2` target.

```
emcli create_blackout
      -name=b1
      -add_targets="database2:oracle_database;database3:oracle_database
      -schedule="frequency:once;start_time:07-08-24
22:30;duration::30;tzinfo:target:tzoffset:1"
      -reason="good reason4"
```

Example 3

The following example creates a blackout on a WebLogic domain, but excludes the database system and its member targets.

```
emcli create_blackout
      -name="wlblkout"
      -add_targets="weblogic1:weblogic_domain"
      -exclude_types="oracle_dbsys"
      -schedule="duration::30"
      -reason="good reason1"
```

Example 4

The following example creates a blackout on a group which contains hundreds of WebLogic domains. The blackout excludes database systems and its member targets (e.g. Oracle home, Listener, Database instance).

```
emcli create_blackout
      -name=Group_Blackout
      -add_targets="Weblogic_Domain_Group:group"
      -exclude_types=oracle_dbsys
      -schedule="duration:1:30"
      -reason="WebLogic Domain Maintenance"
```

create_charge_entity_type

Creates a custom entity type for an Enterprise Manager target type for which there is no current Chargeback support. There can be only one custom entity type for the specified Enterprise Manager target type.

Format

```
emcli create_charge_entity_type
        -target_type="target_type"
```

Options

- **target_type**

  Name of the custom entity type.

Example

The following example creates a new Chargeback entity type named oracle_apache for the Enterprise Manager Apache target type:

```
emcli create_charge_entity_type
       -target_type="oracle_apache"
```

create_charge_item

Creates a custom charge item for Chargeback based on the properties specified in the referenced file.

Format

```
emcli create_charge_item
       -input_file="property_file:filename"
```

Options

The option [-input_file] is the full path of a file that contains the item properties. The following properties can be defined in the file:

- **target_type**

  Target type to which the charge item applies.

- **source_data_type**

  Source data type. Valid values are metric, config, and property.

- **item_name**

  Name of the item.

- **metric_group**

  Metric group name as listed in `list_item_candidates`. This is a required property if source_data_type=metric.

- **metric_column**

  Metric column name as listed in `list_item_candidates`. This is a required property if source_data_type=metric.

- **config_view**

  Config view name as listed in `list_item_candidates`. This is a required property if source_data_type=config.

- **config_key**

  Config key name as listed in `list_item_candidates`. This is a required property if source_data_type=config.

- **config_column**

Config column name as listed in `list_item_candidates`. This is a required property if source_data_type=config.

- **config_data_source**

  Data source of configuration metric. This is a required property if source_data_type=config.

- **property**

  Property name as listed in `list_item_candidates`. This is a required property if source_data_type=property.

- **item_displayname**

  Item display name.

- **unit**

  Unit display name.

- **aggregation_type**

  Type of aggregation to use for this item. Applicable only if data type=number. Valid values are sum and avg. Default value is avg.

- **is_config_condition**

  Item used conditionally in a charge plan. Valid values are 0, 1. Default value is 0.

- **item_category**

  Category of item. Default value is instance. Valid values are cpu, storage, memory, network, and instance.

- **data_type**

  Valid values are string and number. The default value is string for config and property types, and number for metric type.

Examples

Example 1

This example creates a metric custom charge item that bases charges on the average total of processes on a particular host:

```
emcli create_charge_item -input_file="property_file:/home/user/property_file"
Contents of /home/user/property_file:
      target_type=host
      source_data_type=metric
      item_name=total_proc
      metric_group=Load
      metric_column=noOfProcs
      item_displayname=Total Processes
      unit=process
      aggregation_type=avg
      item_category=instance
      data_type=number
```

Example 2

This example creates a configuration custom charge item that can charge different rates for various usage charge items based on the instance region:

```
emcli create_charge_item -input_file="property_file:/home/user/property_file"
Contents of /home/user/property_file:
      target_type=oracle_database
      source_data_type=config
      item_name=custom_config
      config_view=myCustomCCS
      config_key=region
      config_column=country
      config_data_source=regionList.txt
      item_display_name=Region of Instance
      item_category=instance
      data_type=string
```

create_compare_check

Creates a comparison check for the specified template and target type.

Format

There are three forms of the EM CLI create_compare_check command:

```
Latest comparison:
emcli create_compare_check
      -name="<check_name>"
      -template="<template_name>"
      -target_type="<target_type>"
      -compare_type="L<ATEST>"
      -target_name="<target_name>"

Saved comparison:
emcli create_compare_check
      -name="<check_name>"
      -template="<template_name>"
      -target_type="<target_type>"
      -compare_type="S<AVED>"
      -saved_guid="<saved_guid>"

Consistency comparison:
emcli create_compare_check
      -name="<check_name>"
      -template="<template_name>"
      -target_type="<target_type>"
      -compare_type="C<ONSISTENCY>"
```

Options

• name

  Name of the comparison check being created.

• template

  Name of the template being used as the baseline for the comparison.

• target_type

  Target type associated with the template.

• compare_type

  Type of comparison. Values are:

  – L or LATEST

- S or SAVED

- C or CONSISTENCY

• target_name

  Name of the target.

• saved_guid

  Name of the saved comparison.

Example

The following example creates the mycheck1 comparison check using the 'Database Instance Template'. The check is against the Oracle database target my_target. The comparison type is L for latest.

```
emcli create_compare_check
      -name="mycheck1"
      -template="Database Instance Template"
      -target_type="oracle_database"
      -compare_type="L"
      -target_name="my_target"
```

create_config_onetimecompare

Creates a one-time comparison for the specified template and target type.

Format

There are three forms of the create_config_onetimecompare command:

```
Latest comparison:
emcli create_config_onetimecompare
      -name="<check_name>"
      -template_name="<template_name>"
      -target_type="<target_type>"
      -compare_type="L"
      -ref_target_name="<target_name>"
      -dest_target_list="<comma_separated_list_of_target_names>"

Saved comparison:
emcli create_config_onetimecompare
      -name="<check_name>"
      -template_name="<template_name>"
      -target_type="<target_type>"
      -compare_type="S"
      -saved_config_name=<saved_config_name>
      -dest_target_list="<comma_separated_list_of_target_name>" | -
dest_saved_config_list="<comma_separated_list_of_saved_configuration>"

Consistency comparison
emcli create_config_onetimecompare
      -name="<check_name>"
      -template_name="<template_name>"
      -target_type="<target_type>"
      -compare_type="C"
      -ref_target_name="<target_name>"
```

Options

- name

  Name of the one-time comparison.

- template_name

  Comparison Template name to be used for the one time comparison.

- target_type

  Target type of the one-time comparison.

- compare_type

  Comparison type:

  - L for latest comparison

  - S for saved comparison

  - C for consistency comparison

- ref_target_name

  Reference target name of the one-time comparison.

- dest_target_list

  Comma separated list of destination target names of the one-time comparison.
  Can be used with the dest_saved_config_list option when the comparison type is
  "S".

- saved_config_name

  Saved configuration name of the target.

- dest_saved_config_list

  Comma separated list of destination saved configuration names. Can be used with
  the dest_target_list option when the comparison type is "S".

Examples

Example 1

The following example creates a one-time comparison for the Latest compare type.

```
emcli create_config_onetimecompare
     -name="my_check"
     -template_name="my_template"
     -target_type="host"
     -compare_type="L"
     -ref_target_name="target_name"
     -dest_target_list="target1,target2,target3"
```

Example 2

The following example creates a one-time comparison for the Saved compare type.

```
emcli create_config_onetimecompare
     -name="my_check"
     -template_name="my_template"
     -target_type="host"
     -compare_type="S"
```

ORACLE®

```
    -saved_config_name="saved_config"
    -dest_target_list="target1,target2,target3"
```

create_cost_centers

Creates one or more cost centers and associates them with respective organizational entities.

Standard Mode

```
emcli create_cost_centers
    cost-centers="cost center name,entity name"[;"cost center name,entity name";…]
    [-tenant_name="tenant name"]
```

```
[ ]  indicates that the parameter is optional.
```

Interactive or Script Mode

```
create_cost_centers
    (cost-centers="cost center name,entity name"[;"cost center name,entity name";…]
    [,tenant_name="tenant name"])
```

```
[ ]  indicates that the parameter is optional.
```

Options

- cost_centers

  Specifies one or more cost centers to create and associate with organizational entities, which may be a line-of-business or a department. The organizational entity must already exist. Use a comma to separate the cost center name from the entity name.

- tenant_name

  Specifies the name of the tenant to which the organizational entity associated with the cost center belongs. Default is the tenant of the logged-in user.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

The following example creates two cost centers, receivables and payables, both associated with the accounting department.

```
emcli create_cost_centers
    -cost_centers="receivables,accounting;payables,accounting"
```

Example 2

The following example creates two cost centers, coders and testers, the former associated with the development department and the latter associated with the QA department. The organizational entities belong to the engineering tenant.

```
emcli create_cost_centers
    -cost_centers="coders,development;testers,QA"
    -tenant_name="engineering"
```

create_credential_set

Creates a new credential set. Only Enterprise Manager Super Administrators can create new credential sets.

Format

```
emcli create_credential_set
      -set_name="set_name"
      -target_type="ttype"
      -supported_cred_types="supported_cred_types"
      -monitoring
      [-auth_target_type = "authenticating_target_type"
      [-description ="description]"
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **set_name**

    Credential set name to be created.

*   **target_type**

    Target type of the new credential set.

*   **supported_cred_types**

    Credential types supported by this credential set. You can list the available credential types by using the command show_credential_type_info.

*   **monitoring**

    Creates a monitoring credential set.

*   **auth_target_type**

    Target type for the supported cred types. The default value is target_type.

*   **description**

    Description of the credential set.

Example

This example creates a new credential set named New_Credential_Set.

```
emcli create_credential_set
      -set_name=New_Credential_Set
      -target_type=host
      -supported_cred_types=HostCreds;HostSSHCreds
      -description="Example credential set"
```

create_custom_plugin_update

Creates a custom plug-in update using a plug-in that is already deployed to a Management Agent. Includes all of the patches that were applied to the source plug-in. Use this in place of Oracle-supplied plug-in versions for all subsequent plug-in deployments on any Management Agent.

Format

```
emcli create_custom_plugin_update
      -agent_name="agent_name"
      -plugin_id="plugin_id"
      [-overwrite]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **agent_name**

  Management Agent (host:port) on which the plug-in and its patches are deployed.

- **plugin_id**

  ID of the plug-in that should be used for creating the custom plug-in update. To view a list of plug-ins deployed on a Management Agent, run 'emcli list_plugins_on_agent'.

- **overwrite**

  Overwrites and updates an existing custom plug-in update, if a custom plug-in update already exists for that plug-in in the repository. If not provided, the new custom plug-in update is not created for that plug-in. Applies only for subsequent plug-in deployments. Does not automatically redeploy on the Management Agents where the source plug-in was previously deployed. To redeploy on such Management Agents, run 'emcli redeploy_plugin_on_agent'.

Examples

Example 1

The following example creates a custom plug-in update for the `oracle.sysman.db` plug-in that is already deployed on the Management Agent named `host.example.com`. If a custom plug-in update already exists for the `oracle.sysman.db` plug-in, then the command does not overwrite it, and therefore, does not create a new custom plug-in update.

```
emcli create_custom_plugin_update
      -agent_name="host.example.com"
      -plugin_id="oracle.sysman.db"
```

Example 2

The following example creates a custom plug-in update for the `oracle.sysman.db` plug-in, which is already deployed on the Management Agent named `host.example.com`, by overwriting and updating the custom plug-in update that already exists for the `oracle.sysman.db` plug-in in the repository.

```
emcli create_custom_plugin_update
      -agent_name="host.example.com"
      -plugin_id="oracle.sysman.db"
      -overwrite
```

create_database

Creates a database.

Format

```
emcli create_database
        [-dbType="type_of_database"]
        [-hostTargets="list_of_host_targets"]
        [-cluster="cluster_target_name"]
        -oracleHome="Oracle_Home_location"
        -gdbName="global_database_name"
        -templateName="path_and_display_name_of_the_software_library_entity"
        -hostCreds="named_credential_for_OS_user"
        -sysCreds="named_credential_for_SYS_user"
        -systemCreds="named_credential_for_SYSTEM_user"
        -dbsnmpCreds="named_credential_for_DBSNMP_user"
        [-sid="database_system_identifier"]
        [-racConfigType="RAC_configuration_type"]
        [-nodeList="comma-separated_node_names"]
        [-serverPoolList="comma-separated_list_of_server_pools"]
        [-newServerPool="new_server_pool_name_and_cardinality"]
        [-racOneServiceName="service_name_for_RAC_one-node_database"]
        [-templateInSwlib="TRUE|FALSE"]
        [-templateStageLocation="temporary_directory_on_agent_side"]
        [-storageType="FS|ASM"]
        [-dataFileLocation="Location_of_data_files "]
        [-recoveryAreaLocation="Fast_Recovery_Area_location "]
        [-enableArchiving]
        [-useOMF]
        [-listeners="comma-separated_list_of_listeners_database"]
        [-newListener="new_listener_and_port"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **dbType**

  Type of database that needs to be created. Valid values are:

  – SINGLE_INSTANCE —To create a database on one particular host or a list of hosts.

  – RAC — To create a cluster database on multiple nodes.

  – RACONE — To create a RAC One-node database.

  RAC and RACONE require the use of the cluster option.

- **hostTargets**

  Comma-separated list of host targets where a single-instance database needs to be created. This is a mandatory option for a SINGLE_INSTANCE database.

- **cluster**

  Cluster target name for the RAC database on which a cluster needs to be created. The target name should be valid and should have at least one node attached to the target. This is a mandatory option for RAC and RACONE databases.

- **oracleHome**

  Oracle home of the host targets or cluster target. The Oracle home should be present in all of the targets.

- **gdbName**

Global database name of the database.

- **templateName**

  Fully-qualified path and display name of the software library entity.

- **hostCreds**

  Named host credentials of the user who owns the Oracle Home installation.

- **sysCreds**

  Named database credentials to be used to create the SYS user.

- **systemCreds**

  Named database credentials to be used to create the SYSTEM user.

- **dbsnmpCreds**

  Named database credentials to be used to create the DBSNMP user.

- **sid**

  Database system identifier., which can be a maximum length of 12 for SINGLE_INSTANCE, 8 otherwise. This should be alphanumeric, with the first character being an alpha character.

- **racConfigType**

  RAC configuration type. Valid values are:

  – POLICY_MANAGED

  – ADMIN_MANAGED

  The default is ADMIN_MANAGED if not provided.

- **nodeList**

  List of valid node names for ADMIN_MANAGED RAC databases. If not provided, all the nodes for the given cluster target are used.

- **serverPoolList**

  Comma-separated list of server pool names for POLICY_MANAGED RAC databases.

- **newServerPool**

  > **Note:**
  >
  > You can either use serverPoolList or newServerPool, but not both. For newServerPool, cardinality is mandatory and should be a positive integer greater than 0.

- **racOneServiceName**

  Service name for the RAC One Node database.

- **templateInSwlib**

  Boolean value stating whether the template is from the software library. Valid values are TRUE if the template is from the software library, otherwise FALSE. The default is FALSE if you do not provide this option.

- **templateStageLocation**

Fully-qualified path to where the template should be staged on the host target.

- **storageType**

    Type of storage preferred for the database. Valid values are:

    – FS for File System. This is the default if the option is not provided.

    – ASM for Automatic Storage Management.

- **dataFileLocation**

    Location of the data files.

- **recoveryAreaLocation**

    Fast Recovery Area location.

- **enableArchiving**

    Indicates whether archiving of the database is required. Valid values are TRUE if archiving is required, otherwise FALSE. The default is FALSE.

- **useOMF**

    Indicates whether to use Oracle Managed Files.

- **listeners**

    Comma-separated list of listeners (name:port) to register the created database. This is for the SINGLE_INSTANCE database type only, and will be ignored for a RAC database.

- **newListener**

    New listener (name:port) creates a new listener and registers the database. This is for the SINGLE_INSTANCE database type only, and will be ignored for a RAC database.

Examples

Example 1

```
emcli create_database -oracleHome=/u01/app/oracle/product/11.2.0/dbhome_2 -
gdbName=testRACCli -hostCreds=cluster_named -sysCreds=sys -systemCreds=system -
dbsnmpCreds=dbsnmp
                     -templateName=/u01/app/oracle/product/11.2.0/
dbhome_2/assistants/dbca/templates/General_Purpose.dbc -dbType=RAC -
cluster=cluster1
                     -dataFileLocation=/u01/share/oradata -
recoveryAreaLocation=/u01/share/fra
```

Example 2

```
emcli create_database -oracleHome=/u01/app/oracle/product/11.2.0/dbhome_2 -
gdbName=testdbee -hostCreds=cluster_named
                     -sysCreds=sys -systemCreds=system -dbsnmpCreds=dbsnmp -
templateName=/u01/app/oracle/product/11.2.0/
dbhome_2/assistants/dbca/templates/General_Purpose.dbc
                     -dbType=RAC -cluster=cluster1 -
racConfigType=POLICY_MANAGED -newServerPool=sp1:2
```

create_database_size

Specify a database size that overrides values specified in the service template.

Format

```
emcli create_database_size -name="<size name>"
-description="<size description>"
[-attributes="cpu:<number of cores>;memory:<memory in GB>;processes:<max number of
processes>;storage:<Total Storage in GB allocated to database>;"]
[-source_type="Profile Source"]

[ ]  indicates that the parameter is optional.
```

Note: Use one or more attributes to specify the database size. The different attributes must be separated by a semicolon (;). Attributes specified using the database_size verb override values specified in the service template.

Options

- name

  Creates a name for the database size.

- description

  Creates a description for the database size.

- attributes

  Defines the database size. Attributes must be separated by a semicolon(;). You can specify values for the following attributes:

  cpu: Total number of cpu cores.

  memory: Total maximum in GB.

  processes: Total number of processes that can simultaneously connect to the database.

  storage: Total storage that is allocated to the database (in GB)

Example

The following example creates a database size named Small with a maximum of four CPUs, 50 GB of storage, and 4 GB of memory.

```
emcli create_database_size
     -name=Small
     -description="Small size database"
     -attributes="cpu:4;storage:50;memory:4"
     -source_type="weblogic_domain"
```

create_dbaas_quota

Creates a database quota for an SSA user role.

Format

```
emcli create_dbaas_quota
      -role_name="<SSA user role name>"
      -databases="<number of database requests>"
      -schema_services="<number of schema service requests>"
      -pluggable_databases="<number of Pluggable database service requests>"
      -memory="<memory(GB)>"
      -storage="<storage(GB)>"
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **role_name**

  Name of the SSA user role for which the quota is to be created.

- **databases**

  Number of database service requests allowed. For example, for 10 requests enter:

  ```
  -databases="10"
  ```

- **schema_services**

  Number of schema service requests allowed. For example, for 10 requests enter:

  ```
  -schema_services="10"
  ```

- **pluggable_databases**

  Number of pluggable database service requests allowed. For example, for 10 requests enter:

  ```
  -pluggable_databases="10"
  ```

- **memory**

  Amount of memory usage allowed. For example, for 10 GB enter:

  ```
  -memory="10"
  ```

- **storage**

  Amount of storage usage allowed. For example, for 10 GB enter:

  ```
  -storage="10"
  ```

Example

This example assigns the quota for the role My Role:

```
emcli create_dbaas_quota
      -role_name="My Role"
      -databases="10"
      -schema_services="10"
      -pluggable_databases="10"
      -memory="99"
      -storage="99"
```

create_dbprofile

Creates a new database profile.

Format

```
emcli create_dbprofile
                -input_file=data:"file:path"
        [-schedule=
                [frequency:interval|weekly|monthly|yearly];
                start_time:yy-MM-dd HH:mm;
                end_time:yy-MM-dd HH:mm;
                [repeat:#m];
                [months:#,#,#,...];
```

```
                        [days:#,#,#,...];
                        [tz:{timezone ID}];
                        [grace_period:xxx];
            ]
    [-purge_policy=DAYS|SNAPSHOTS: number]
```

`[ ]  indicates that the parameter is optional.`

Options

- input_file

  A property file which completely describes the type of profile that will be created and the options used.

- schedule

  frequency: Frequency type with which the Database Profile will be created. It can be interval (in minutes), weekly, monthly or yearly

  start_time: Denotes the starting time of Database Profile Component creation in the format yy-MM-dd HH:mm

  end_time: Denotes the end time of Database Profile Component Creation Repetition in the format yy-Mm-dd HH:mm

  repeat: Repetition rate at which Database Profile will be created. If the frequency is interval, then repeat will be in minutes

  months: Number of months after which repetition of Database Profile Component Creation will occur

  days: Number of days after which repetition of Database Profile Component Creation will occur

  tz: Time Zone ID for example tz:America/New_York

  grace_period: A period of time in minutes that defines the maximum permissible delay when attempting to create a Databasse Profile. If the job system cannot start the execution within a time period equal to the scheduled time + grace period, it will set the create Database Profile to be skipped.By default, grace period is indefinite

- purge_policy

  You can purge the collected data based on number of days or count of snapshots. If you do not specify purge_policy, it is defaulted to NONE. Allowed values: DAYS, SNAPSHOT

  DAYS specify the number of days after which the data component should be purged.

  SNAPSHOT specify the count or number of data components, after which older data will be purged

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example creates a new database profile based on the property file "profile.txt" with the specified schedule and purge policy.

```
emcli create_dbprofile -input_file="data:/tmp/profile.txt" -
schedule="frequency:interval;start_time:14-10-05 05:30;end_time:14-10-12
05:23;repeat:30;grace_period:60;tz:America/New_York" -purge_policy=DAYS:2
```

create_diag_snapshot

Creates a diagnostic snapshot for specified targets.

Format

```
emcli create_diag_snapshot
     -name=<name>
     -desc=<description>
     -start_time=<yyyy/MM/dd HH:mm>
     -end_time=<yyyy/MM/dd HH:mm>
     -targets=<type1:name1;type2:name2;…>
     [-diag_type_odl_target_types=<type1;type2; ….>]
     [-diag_type_odl_online_logs=<true|false>]
     [-diag_type_odl_offline_logs=<true|false>]
     [-diag_type_jvmd_target_types=<type1;type2; ….>]
     [-diag_type_jvmd_properties="<pname1:pval1;pname2:pval2;…>"]
     [-debug]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **name**

Name of diagnostic snapshot to be created. Make sure that a diagnostic snapshot does not exists for the specified name.

• **desc**

Description of the diagnostics snapshot.

• **start_time**

Start time for collecting the logs. The snapshot will contain all logs between the start time and end time. Make sure that the duration is valid for the snapshot.

• **end_time**

End time for collecting the logs. The snapshot will contain all logs between the start time and end time. Make sure that the duration is valid for the snapshot.

• **targets**

Target type and target name list for the snapshot. This list can contain all targets for the specific system. User can choose specific target types in optional options for selected diagnostic types.

• **diag_type_odl_target_types**

Target type list for the Oracle Diagnostic Logging (ODL) diagnostic type. You can select a subset of target types from the target list for snapshot creation.

• **diag_type_odl_online_logs**

By default, online logs are collected for a snapshot. You can choose to collect online, offline, or both logs for the Oracle Diagnostic Logging (ODL) diagnostic type.

• **diag_type_odl_offline_logs**

By default, offline/archive logs are not collected for a snapshot. You can choose to collect online, offline, or both logs for the Oracle Diagnostic Logging (ODL) diagnostic type.

- **diag_type_jvmd_target_types**

  Target type list for the JVMD diagnostic type. You can select a subset of target types from the target list for snapshot creation.

- **diag_type_jvmd_properties**

  Properties list to collect logs for the JVMD diagnostic type.

- **debug**

  Runs the verb in verbose mode for debugging purposes.

Examples

Example 1

This example creates a snapshot for EMGC_DOMAIN and EMGC_OMS1 targets with offline logs. The target types (weblogic_domain and weblogic_j2eeserver) belong to the Oracle Diagnostic Logging (ODL) diagnostic type.

```
emcli create_diag_snapshot
    -name=wls_snapshot
    -desc= "Snapshot for Weblogic Domains and Server"
    -start_date="2012/10/02 10:30"
    -end_date="2012/10/03 22:30"
    -targets="weblogic_domain:/EMGC_EMGC_DOMAIN/EMGC_DOMAIN;
              weblogic_j2eeserver: /EMGC_EMGC_DOMAIN/EMGC_DOMAIN/EMGC_OMS1"
```

Example 2

This example creates a snapshot for the weblogic_j2eeserver target type with offline logs. You can filter the target types on top of the target list.

```
emcli create_diag_snapshot
    -name=wls_snapshot
    -desc="Snapshot for Weblogic Domains and Server"
    -start_date="2012/10/02 10:30"
    -end_date="2012/10/03 22:30"
    -targets="weblogic_domain:/EMGC_EMGC_DOMAIN/EMGC_DOMAIN;
              weblogic_j2eeserver:/EMGC_EMGC_DOMAIN/EMGC_DOMAIN/EMGC_OMS1;
              weblogic_j2eeserver:/EMGC_EMGC_DOMAIN/EMGC_DOMAIN/EMGC_ADMIN_SERVER"
    -diag_type_odl_target_types="weblogic_j2eeserver"
    -diag_type_odl_offline_logs=true
```

create_dynamic_group

Creates a new dynamic Group. Specifies a name for the group and target selection criteria. After the group is created, you can edit the group from the Enterprise Manager Cloud Control console to configure Summary Metrics to be displayed for group members.

Format

```
emcli create_dynamic_group -name="<Name>"
[-target_types="<Targets>;"]
[-hostnames="<Hosts>;"]
[-properties="<Properties>:<Values>;"]
[-is_propagating="<True|False>"]
```

```
[ ] indicates that the parameter is optional.
```

Options

- **name**

  Name of the group.

- **target_types**

  Target types based on which targets are selected to be included in group

- **hostnames**

  Hosts where the targets will be picked up from to be included in group

- **properties**

  Additional properties to be applied in target selection criteria.

- **is_propagating**

  Specifies a flag to indicate whether or not a privilege on the group will be propagated to member targets. The default value is false.

create_far_sync

Creates a far sync instance.

Format

```
emcli create_far_sync
      -primary_target_name="<primary database target name>"
      -primary_target_type="oracle_database|rac_database"
        -far_sync_sid="<far sync instance name>"
        -far_sync_db_unique_name="<far sync instance unique name>"
        [-far_sync_em_target_name="<far sync instance target name>"]
        [-far_sync_host_name="<far sync instance host name>"]
        [-far_sync_oracle_home="<far sync instance Oracle home>"]
        [-far_sync_db_create_file_dest="<far sync instance files location>"]
        [-far_sync_fra_location="<far sync instance fast recovery area>"]
        [-far_sync_fra_size="<far sync instance fast recovery area size>"]
        [-primary_db_creds_name="<primary database credential name>"]
        [-primary_host_creds_name="<primary database host credential name>"]
        [-far_sync_host_creds_name="<far sync instance host credential name>"]
        [-far_sync_asm_cred_name="<far sync instance asm credential name>"]
        [-far_sync_storage_type="<far sync instance storage type>"]
        [-far_sync_listener_name="<far sync instance listener name>"]
        [-far_sync_listener_port="<far sync instance listener port>"]
        [-redo_source="<far sync instance redo source>"]
        [-redo_mode="<far sync instance redo shipping mode>"]
        [-redo_dest="<far sync instance redo destination>"]
```

Options

- primary_target_name

  Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- primary_target_type

  Primary database target type. Allowed values:

- – oracle_database: Single-instance database

- – rac_database: Cluster database

- far_sync_sid

  Far sync instance name.

- far_sync_db_unique_name

  Far sync instance unique name.

- far_sync_em_target_name

  Far sync instance Enterprise Manager target name. The default value is far_sync_db_unique_name followed by db_domain.

- far_sync_host_name

  Far sync instance host name. The default value is primary host name.

- far_sync_oracle_home

  Far sync instance host Oracle home location. The default value is primary database Oracle home location.

- far_sync_db_create_file_dest

  Oracle-managed files (OMF) location for far sync instance files. Can be a regular file system (if storage_type is FILE_SYSTEM) or an ASM diskgroup (if storage_type is ASM_STORAGE). The default values are:

  - – File system - Far sync instance's <oracle_base>/oradata

  - – ASM - Mandatory

- far_sync_fra_location

  Far sync instance fast recovery area.

  The default values are:

  - – File system - Far sync instance's <oracle_base>/fast_recovery_area

  - – ASM - Mandatory

- far_sync_fra_size

  Far sync instance fast recovery area size in MB. The default value is Primary fra_size.

- primary_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Preferred credentials will be used as the default value.

- primary_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle home. Preferred credentials will be used as the default value.

- far_sync_host_creds_name

  Far sync instance host named credential for an operating system user who can access the far sync instance Oracle home. Preferred credentials will be used as the default value.

- far_sync_asm_cred_name

  Automatic Storage Management named credential.

- far_sync_storage_type

  Far sync instance storage type. The allowed values are:

  – FILE_SYSTEM: Far sync instance files will be in a regular file system

  – ASM_STORAGE: Far sync instance will use Automatic Storage Management

  The default value is FILE_STORAGE.

- far_sync_listener_name

  Far sync instance listener name. If the default value is not specified, the first existing TCP listener found in the far sync instance Oracle home or a running GRID INFRA listener will be used. If listener_name is specified, listener_port must also be specified.

- far_sync_listener_port

  Far sync instance listener port. If listener_port is specified, listener_name must also be specified.

- redo_source

  db_unique_name of the database that will act as a redo source for the new far sync instance. The default value is Primary database db_unique_name.

- redo_mode

  The mode of redo shipping to the new far sync instance. The default value is sync and the allowed value is sync or async.

- redo_dest

  The list of db_unique_names of the standby databases to which the new far sync instance will ship redo. For example "standby1,standby2". There is default value for this option.

Examples

Example 1

The following command creates a file system based far sync instance.

```
emcli create_far_sync
        -primary_target_name=database
        -primary_target_type="oracle_database" -far_sync_sid="fsd4"
        -far_sync_db_unique_name="fsd4" -primary_db_creds_name=SYS
        -primary_host_creds_name=HOST_CRED
        -db_create_file_dest=/scratch/orabase/ORADATA -
far_sync_em_target_name=fsd4
        -far_sync_host_name=abc.us.oracle.com
        -far_sync_oracle_home=/scratch/orabase/product/12.1.0/dbhome_1
        -fra_location=/scratch/orabase/fra -fra_size=3000
        -far_sync_host_creds_name=HOST_CRED -listener_name=LIST_L -
listener_port=1531
```

Example 2

The following command creates an ASM based far sync instance.

```
emcli create_far_sync
        -primary_target_name="database"
        -primary_target_type="oracle_database" -far_sync_sid="asmfs2"
        -far_sync_db_unique_name="asmfs2" -primary_db_creds_name=SYS
```

```
-primary_host_creds_name=HOST_CRED -db_create_file_dest='+DATA'
-far_sync_em_target_name=asmfs2 -far_sync_host_name=abc.us.oracle.com
-far_sync_oracle_home=/scratch/orabase/product/12.1.0/dbhome_1
-fra_location=+DATA -fra_size=3000 -far_sync_host_creds_name=HOST_CRED
-far_sync_asm_cred_name=ASM_CRED -redo_source=farsync1 -redo_mode=async
-redo_dest="stdb1,stdb2"
```

create_fmw_domain_profile

Creates a Fusion Middleware provisioning profile from a WebLogic Domain.

Format

```
emcli create_fmw_domain_profile
        -name="profile_name"
        -ref_target="reference_target_name"
        [-description="profile_description"]
        [-oh_cred="Oracle_home_owner_credentials"]
        [-includeOh]
        [-schedule=
                start_time:yyyy/MM/dd HH:mm;
                [tz:{java_timezone_ID}];
                [grace_period:xxx];
        ]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **name**

    Name of the profile to be created.

*   **ref_target**

    Name of the WebLogic Domain target to be used as a reference to create the profile.

*   **description**

    Description of the profile to be created.

*   **oh_cred**

    Named credential to be used to access the reference host. The format is:

    ```
    CREDENTIAL_NAME:CREDENTIAL_OWNER
    ```

    All operations are performed on the Administration Server host. Credentials of the Oracle Home owner on the Administration Server host are required. If no named credential is provided, preferred host credentials for the Oracle Home target are used.

*   **includeOh**

    Includes the Oracle Home binaries in the profile.

*   **schedule**

    Schedule for the Deployment Procedure. If not specified, the procedure executes immediately.

    –   start_time: Time when the procedure should start.

    –   tz: Time zone ID.

    –   grace_period: Grace period in minutes.

Examples

Example 1

This example creates a WebLogic Domain profile for the specified schedule from the given WebLogic Domain target using preferred credentials.

```
emcli create_fmw_domain_profile
     -name="BitlessDomainProfile"
     -ref_target="/Farm01_base_domain/base_domain"
     -description="A domain profile without software bits"
     -schedule="start_time:2014/6/21 21:23;tz:America/New_York;grace_period:60"
```

Example 2

This example immediately creates a WebLogic Domain plus Oracle Home from the given WebLogic Domain target using given named credentials.

```
emcli create_fmw_domain_profile
     -name="DomainProfileWithBits"
     -ref_target="/Farm01_base_domain/base_domain"
     -oh_cred="MY_HOST_CRED:SYSMAN"
     -includeOh
```

create_fmw_home_profile

Creates a Fusion Middleware provisioning profile from an Oracle Home.

Format

```
emcli create_fmw_home_profile
     -name="profile_name"
     -ref_target="reference_target_name"
     [-description="profile_description"]
     [-oh_cred="Oracle_home_owner_credentials"]
     [-schedule=
          start_time:yyyy/MM/dd HH:mm;
          [tz:{java timezone ID}];
          [grace_period:xxx];
     ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the profile to be created.

- **ref_target**

  Name of the Oracle Home target to be used as a reference to create the profile.

- **description**

  Description of the profile to be created.

- **oh_cred**

  Named credential to be used to access the reference host. The format is:

  ```
  CREDENTIAL_NAME:CREDENTIAL_OWNER
  ```

If no named credential is provided, preferred host credentials for the Oracle Home target are used.

• **schedule**

Schedule for the Deployment Procedure. If not specified, the procedure executes immediately.

– start_time: Time when the procedure should start.

– tz: Time zone ID.

– grace_period: Grace period in minutes.

Examples

Example 1

This example creates a profile on the specified schedule from the given Oracle Home target using preferred credentials.

```
emcli create_fmw_home_profile
      -name="OhProfile1"
      -ref_target="/Farm01_base_domain/base_domain"
      -description="An Oracle Home profile"
      -schedule="start_time:2014/6/21 21:23;tz:America/New_York;grace_period:60"
```

Example 2

This example immediately creates a profile from the given Oracle Home target using given named credentials.

```
emcli create_fmw_home_profile
      -name="OhProfile2"
      -ref_target="WebLogicServer_10.3.6.0_myhost.mycompany.com_5033"
      -oh_cred="MY_HOST_CRED:SYSMAN"
```

create_gold_agent_image

Creates a Management Agent gold image using the specified source Management Agent or by importing an already created image from another Enterprise Management System.

Format

```
emcli create_gold_agent_image
        -image_name="gold_image_name"
        -version_name="gold_image_version_name"
        -source_agent|-import_location="source_agent|import_location"
        [-gold_image_description="gold_image_description"]
        [-working_directory="working_directory_location"]
        [-config_properties= " agent_configuration_properties"]
[ ]  indicates that the parameter is optional.
```

Options

• image_name

Specifies the gold image name to which the created Management Agent gold image must be added.

• version_name

Specifies the version name of the Management Agent gold image.

- source_agent

  Specifies the Management Agent that must be used as the source to create the Management Agent gold image. To view a list of the Management Agents that can be used as a source to create a gold image, run `emcli get_targets -target="oracle_emd"`.

- import_location

  Specifies the location where gold agent image software is staged for creating the gold agent image version and is accessible from all OMSes.

- gold_image_description

  Provides a description of the Management Agent gold image.

- working_directory

  Specifies the working directory that must be used to create the Management Agent gold image. The default working directory is `$AGENT_INSTANCE_HOME/install`.

- config_properties

  Specifies the Management Agent configuration properties separated by ";" that must be captured while creating the Management Agent gold image.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

The following example creates a Management Agent gold image OPC_AGI_DB_JUL_13, using example.com:3872 as the source Management Agent, and adds the gold image version to the gold image OPC_DB_MONITORING:

```
emcli create_gold_agent_image
                -source_agent=example.com:3872 -version_name=OPC_AGI_DB_JUL_13 -
image_name=OPC_DB_MONITORING
```

Example 2

The following example creates a Management Agent gold image OPC_AGI_DB_JUL_13, using example.com:3872 as the source Management Agent, /tmp as the working directory, and adds the gold image version to the gold image OPC_DB_MONITORING:

```
emcli create_gold_agent_image
                -source_agent=example.com:3872 -version_name=OPC_AGI_DB_JUL_13 -
image_name=OPC_DB_MONITORING -working_directory=/tmp
```

create_group

Defines a group name and its members. After you create the group, you can edit it from the Enterprise Manager Cloud Control console to configure Summary Metrics to be displayed for group members.

Standard Mode

```
emcli create_group
      -name="name"
      [-type=<group>]
      [-add_targets="name1:type1;name2:type2;..."]...
      [-is_propagating="true/false"]
```

```
[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
create_group
      (name="name"
      [,type=<group>]
      [,add_targets="name1:type1;name2:type2;..."]...
      [,is_propagating="true/false"])
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the group.

- **type**

  Group type: group. Defaults to "group".

- **add_targets**

  Add existing targets to the group. Each target is specified as a name-value pair `target_name:target_type`. You can specify this option more than once in Standard Mode.

- **is_propagating**

  Flag that indicates whether or not privilege on the group will be propagated to member targets. The default is false.

Example

This example creates a database-only group named `db_group`. This group consists of two Oracle databases: `emp_rec` and `payroll`.

```
emcli create_group
      -name=db_group
      -add_targets="emp_rec:oracle_database"
      -add_targets="payroll:oracle_database"
```

create_inst_media_profile

Defines a group name and its members. After you create the group, you can edit it from the Enterprise Manager Cloud Control console to configure Summary Metrics to be displayed for group members.

Standard Mode

```
emcli create_inst_media_profile
      -name="profile_name"
      -host="host_target"
      -version="media_version"
      -platform="media_platform"
      [-description="profile_description"]
      [-host_cred="Oracle_home_owner_credentials"]
      -files=
            WebLogic:WLSFile1;
            SOA:SOAFile1,SOAFile2;
            OSB:OSBFile;
            RCU:RCUFile;

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the profile to be created.

- **host**

  Name of the host target that where all of the installation files are stored.

- **version**

  Version of the installation media.

- **platform**

  Platform for which the installation media is applicable.

- **description**

  Description of the profile to be created.

- **host_cred**

  Named credential to be used to access the files. The format is:

  ```
  CREDENTIAL_NAME:CREDENTIAL_OWNER.
  ```

  If you do not provide a named credential, preferred host credentials for the Oracle Home target are used.

- **files**

  List of files to be uploaded to the Software Library. Acceptable products are WebLogic, SOA, OSB and RCU. An upload for WebLogic is mandatory. The format is:

  ```
  PRODUCT1:FILE1,FILE2;PRODUCT2:FILE3,FILE4
  ```

Examples

Example 1

This example uploads the installation media file for the WebLogic Server to the Software Library from the given location on the given host. Preferred host credentials will be used to access the files.

ORACLE®

```
emcli create_inst_media_profile
     -name="WebLogic1036Installer"
     -host="myhost.mycompany.com"
     -description="WebLogic Server 10.3.6.0 installer"
     -version="10.3.6.0"
     -platform="Generic"
     -files="WebLogic:/u01/media/weblogic/wls1036_generic.jar"
```

Example 2

This example uploads the installation media files for SOA and the WebLogic Server to the Software Library from the given location on the given host. The provided named credentials are used to access the files.

```
emcli create_inst_media_profile
     -name="SOA+WLSInstaller"
     -host="myhost.mycompany.com"
     -description="SOA 11.1.1.7.0 and WebLogic Server 10.3.6.0 installer"
     -version="11.1.1.7.0"
     -platform="Generic"
     -host_cred="MY_HOST_CRED:SYSMAN"
     -files="WebLogic:/u01/media/weblogic/
      wls1036_generic.jar;SOA:/u01/media/soa/soa1.zip,/u01/media/soa/soa2.zip"
```

create_jeeappcom

Creates a Java EE Application Component in the software library. On successful creation, the entity revision is displayed under the specified folder in the software library.

This entity has not been uploaded yet, use the verb upload_jeeappcomp_file to upload it to the software library.

Format

```
emcli create_jeeappcomp
      -name="entity_name"
      -folder_id="folder_id"
      [-desc="entity_desc"]
      [-attr="<attr name>:<attr value>"]
      [-prop="<prop name>:<prop value>"]
      [-secret_prop="<secret prop name>:<secret prop value>"]
      [-note="note text"]

[ ]  indicates that the parameter is optional
```

Options

*   **name**

    Name of the entity.

*   **folder_id**

    ID of the folder where the entity will be created. The Software Library Home page exposes the identifier for folders and entities as a custom column called Internal ID. By default, this column is hidden.

*   **desc**

    A short description about the entity. The new description is visible to all existing revisions.

*   **attr**

A name:value pair for specifying the attributes of an entity. It is represented as "attr_name:attr_value". For specifying values for multiple attributes, repeat the `-attr` option.

- **prop**

  A name:value pair for specifying the configuration properties of an entity. It is represented as prop_name:prop_value. For specifying values for multiple properties, repeat the option.

- **secret_prop**

  A name:value pair for specifying the configuration property and its secret value. Do not provide the secret value on the command line. Instead, enter the property name and press the Enter key. Provide the secret value when you are prompted for it.

- **note**

  Information related to the entity such as changes being made to the entity or modification history that you want to track.

Examples

Example 1

Creates a Java EE Application component called "myJeeAppComp' in the folder identified by folder_id. You can find the folder ID using the custom column called Internal ID available on the Software Library home page. Note that this column is hidden by default.

```
emcli create_jeeappcomp
      -name="myJeeAppComp"
       folder_
id="oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19AA560C"

+E34
```

Example 2

Creates entity named 'myJeeAppComp' in the folder identified by folder_id with a short description about the entity. Entity attributes such as PRODUCT, PRODUCT_VERSION, and VENDOR are specified. Value for the DEFAULT_HOME configuration property is also specified. A note that includes information related to the entity is included.

The identifier of the newly created entity revision will be printed on the standard output.

```
emcli upload_jeeappcomp_file
emcli create_jeeappcomp
      -name="myJeeAppComp"
folder_id="oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19
AA560C"
      -desc="myJeeAppComp description"
      -attr="PRODUCT:JEEApp"
      -attr="PRODUCT_VERSION:3.0"
      -attr="VENDOR:Vendor"
      -prop="DEFAULT_HOME:/u01/myJeeAppComp3/"
      -note="myJeeAppComp for test servers"
```

create_job

Creates and schedules a job. This verb supports multi-task jobs.

> **Note:**
>
> EM CLI permits OS Script jobs to be run against database targets by setting the targetType property for -input_file in the create_job verb. For example:
>
> ```
> targetType=oracle_database
> ```
>
> You can set other target types similarly.

EM CLI supports the following job types:

```
ASMSQLScript
ASSOCIATE_CS_FA
ASSOCIATE_DOMAIN_FA
AssociateClusterASM
BlockAgent
CoherenceCacheAddition
CoherenceNodesRefresh
Config Log Archive Locations
DbMachineDashboard
DiscoverPDBEntities
FusionMiddlewareProcessControl
GlassFishProcessControl
InstallKernelModuleJob
Log Rotation
OSCommand
OpatchPatchUpdate_PA
RMANScript
RefreshFromEMStore
RefreshFromMetalink
RefreshFusionInstance
SOABulkRecovery
SQLScript
ShutdownDB
StartDepartedCohNodes
StartDepartedCohStoreNodes
StartFusionInstance
StartupDB
StatspackPurge
StopFusionInstance
Upgrade Exalogic Systems
WebLogic Domain Discover
WebLogic Domain Refresh
WLSTScript
```

> **Note:**
>
> Not all job types support all target types. Use describe_job_type to determine which target types are supported for a given job type.

ORACLE

Format

```
emcli create_job
   -input_file=property_file:"filename"
   [-name="job_name"]
   [-type="job_type"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the job.

- **job_type**

  Name of the job type. You can obtain a template property file for the job type by using the describe_job_type verb.

- **input_file**

  Provide the file name to load the properties for creating and scheduling the job. The property file must be accessible to the EM CLI client for reading. Another job of the same job type could also be used to generate the property file using the EM CLI verb describe_job.

  For more information about the input_file option, see -input_file Syntax Guidelines.

Example

This example creates and schedules a job with name MYJOB1 and job type MyJobType1 with the property file present at location /tmp/myjob1_prop.txt.

```
emcli create_job -name=MYJOB1 -job_type=MyJobType1 -
input_file="property_file:/tmp/myjob1_prop.txt"
```

create_job_from_library

Creates a job using a library job as a template. This verb supports multi-task jobs.

Format

```
emcli create_job_from_library
    -lib_job_name="library_job_name"
    -name="new_job_name"
   [-owner="library_job_owner"]
   [-input_file=property_file:"filename"]
   [-appendtargets]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **lib_job_name**

  Library job to use as a template.

- **owner**

  Owner of the job. When this option is not specified, the default job owner is the logged in Enterprise Manager administrator.

- **name**

  Name of the new job to be created. You can also specify the name in the property file. If no name is specified, a name is generated from the name of the library job.

- **input_file**

  "filename" can be provided to load the properties for creating the job.

  If you specify a property file, the values in the property file override or append to existing values in the library job. If you do not specify a property file, the library job is submitted unchanged.

  For more information about the input_file option, see -input_file Syntax Guidelines.

- **appendtargets**

  Appends targets in the property file to existing targets in the library job. Otherwise, library job targets are overwritten by targets in the property file if they are specified.

Examples

Example 1

This example creates a job named MYJOB based on the library job MYLIBJOB1.

```
emcli create_job_from_library -lib_job_name=MYLIBJOB1 -name=MyJob
```

Example 2

This example creates a job named MYJOB2 based on the library job MYLIBJOB1. Properties in /tmp/myjob1_prop.txt will override library job settings.

```
emcli create_job_from_library -lib_job_name=MYLIBJOB1 -name=MyJob2 -
input_file=property_file:/tmp/myjob1_prop.txt
```

create_library_job

Creates a library job. This verb supports multi-task jobs.

Format

```
emcli create_library_job
   [-name="job_name"]
   [-type="job_type"]
    -input_file=property_file:"filename"

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the job.

- **type**

  Name of the job type. You can obtain a template property file for the job type by using the describe_job_type verb.

- **input_file**

  Provide the file name to load the properties for creating the library job. The property file must be accessible to the EM CLI client for reading. Another library job of the same job

type could also be used to generate the property file using the EM CLI verb
describe_library_job.

For more information about the input_file option, see -input_file Syntax Guidelines.

Example

This example creates a library job with the name MYLIBJOB1 and job type
MyJobType1 with the property file present at location /tmp/myjob1_prop.txt.

```
emcli create_library_job -name=MYLIBJOB1 -job_type=MyJobType1
-input_file="property_file:/tmp/myjob1_prop.txt"
```

create_mw_profile

Creates a non-Oracle Middleware Provisioning Profile using the provided archive and
properties.

Examples of non-Oracle middleware include Apache Tomcat, JBoss, etc.

For Oracle Middleware Provisioning Profile, refer to other verbs such as
create_fmw_domain_profile or create_fmw_home_profile.

Format

```
emcli create_mw_profile
    -input_file=propertiesXml:"Properties XML"
    -host="Host Target Name"
    [-host_cred="Host Credentials"]
    [-location="Software Library Location"]
    [-input_file=parameters:"Provisioning Options"]
    -files=
        Archive1,
        Archive2,
        Script1,
        Script2

[ ]  indicates that the parameter is optional.
```

Options

* input_file

  An XML file describing the characteristics of the profile. It also contains a list of
  commands that can be executed to provision from the profile.

* host

  Name of the host target where all files are stored including the archives and, if
  required, any scripts required during provisioning.

* host_cred

  Named credentials used to access the reference host. This option is not
  mandatory. To pass the credential option, enter a name:value pair in the following
  format:

  ```
  credential_name:credential_owner
  ```

  Where:

  credential_name is the name of the named credential.

credential_owner is the credentials of the Oracle home owner on the administration server host.

If no named credential is provided, the preferred host credential for the host target will be used.

- location

  The software library location for the profile. This option is not mandatory. The default is: Middleware Provisioning/Generic Profiles.

- input_file:properties

  A properties file listing values for all of the parameters required by the commands listed in the profile properties. This is an optional parameter as the same can be provided during provisioning as well.

- files

  List of files that must be uploaded to the software library. These files are passed in the format file1,file2. All of the files necessary for provisioning from this profile should be provided.

Example

The following example uploads the archives and the provisioning scripts to the software library from the host `myhost.example.com`. The profile is created using the named credentials specified.

```
emcli create_mw_profile
      -input_file=propertiesXml:"/u11/myprofile/properties.xml"
      -host="myhost.mycompany.com"
      -host_cred="MY_HOST_CRED:SYSMAN"
      -location="My Stuff/Profiles/Middleware"
      -files="/u11/myprofile/binaries.zip,/u11/myprofile/
 configuration.zip,/u11/myprofile/cloneBinaries.sh,/u11/myprofile/cloneConfig.sh"
```

create_named_credential

Creates a named credential. You can provide input parameters using command line arguments or an input properties file. It also supports the input_file tag for passwords and parameter values.

Standard Mode

```
 emcli create_named_credential
        -cred_name=<name>
        -auth_target_type=<authenticating_target_type>
        -cred_type=<credential_type>
        -cred_scope=<credential_scope>
        -cred_desc=<credential_description>
        -target_name=<target_name>
        -target_type=<target_type>
        -test
        -test_target_name=<test_target_name>
        -test_target_type=<test_target_type>
        -input_file=<tag:value>
        -input_bfile=<tag:value>
        -properties_file=<filename>
        -attributes=<p1:v1;p2:v2;...>
```

Interactive or Script Mode

```
create_named_credential
        (cred_name=<name>
        ,auth_target_type=<authenticating_target_type>
        ,cred_type=<credential_type>
        ,cred_scope=<credential_scope>
        ,cred_desc=<credential_description>
        ,target_name=<target_name>
        ,target_type=<target_type>
        ,test
        ,test_target_name=<test_target_name>
        ,test_target_type=<test_target_type>
        ,input_file=<tag:value>
        ,input_bfile=<tag:value>
        ,properties_file=<filename>
        ,attributes=<p1:v1;p2:v2;...>)
```

Options

- **cred_name**

  Credential name, such as MyBackUpCreds. This is required if you do not use properties_file.

- **auth_target_type**

  Authenticating target type (e.g. host). This is required if you do not use properties_file.

- **cred_type**

  Credential type. This is required if you do not use properties_file.

- **cred_scope**

  Possible values are global|instance. The default is global.

- **cred_desc**

  Credential description.

- **target_name**

  This is required when cred_scope is instance.

- **target_type**

  This is required when cred_scope is instance.

- **test**

  Use this to test the credential before saving.

- **test_target_name**

  Use this to supply the target name to test a global credential. This is required when cred_scope is global and the test parameter is used.

- **test_target_type**

  Use this to supply the target type to test a global credential. This is required when cred_scope is global and the test parameter is used.

- **input_file**

  Use this to supply sensitive property values from the file.

For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **input_bfile**

  Use this to supply binary property values from the file.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **properties_file**

  Use this to pass all parameters from the file. Values given on the command line take precedence.

- **attributes**

  Specify credential columns as follows:

  ```
  colname:colvalue;colname:colvalue
  ```

  You can change the separator value using -separator=attributes=<newvalue>, and you can change the subseperator value using -subseparator=attributes=<newvalue>.

  For more information about the separator and subseperator parameters, see -input_file Syntax Guidelines.

Error Codes

0 - Success

1 and 223 - Syntax error

Examples

Example 1

This example create a HostCreds named credential with username foo and password bar:

```
emcli create_named_credential
      -cred_name=NC1
      -auth_target_type=host
      -cred_type=HostCreds
      -attributes="HostUserName:foo;HostPassword:bar"
```

Example 2

This example creates a privilege delegation credential with the user name foo, password bar, privilege delegation type SUDO, and RUNAS user root:

```
emcli create_named_credential
      -cred_name=NC1
      -auth_target_type=host
      -cred_type=HostCreds
      -attributes="HostUserName:foo;HostPassword:bar;PDPTYPE:SUDO;RUNAS:root"
```

To use Powerbroker attributes, the string should be:

```
-attributes="HostUserName:foo;HostPassword:bar;PDPTYPE:POWERBROKER;RUNAS:root;
PROFILE:EMGC
```

create_operation_plan

Creates an operational plan for the Oracle Site Guard operation.

Format

```
emcli create_operation_plan
        -primary_system_name="name_of_primary_system"
        -standby_system_name="name_of_standby_system"
        -system_name="name_of_system"
        -operation="name_of_operation"
        -name="name_of_operation_plan"
        -role="role_associated_with_system"
```

Options

- **primary_system_name**

  Name of your system associated with the primary site. Enter this **parameter** for switchover or failover operations.

- **standby_system_name**

  Name of your system associated with the standby site. Enter this **parameter** for switch-over or fail-over operations.

- **system_name**

  Name of the system. Enter this **parameter** for start or stop operations.

- **operation**

  The function of the operation. Examples: switchover, failover, start, or stop.

- **name**

  Name of the operation plan.

- **role**

  Role associated with a system when you run an operation (start or stop).

Examples

Example 1

```
emcli create_operation_plan
      -primary_system_name="BISystem1"
      -standby_system_name="BISystem2"
      -operation="switchover"
      -name="BISystem1-switchover-plan"
```

Example 2

```
emcli create_operation_plan
      -system_name="austin"
      -operation="start"
      -name="BISystem1-start-plan"
      -role="Primary"
```

See Also

submit_operation_plan

create_organizational_entity

Creates an organizational entity, which can be either a department or a line-of-business. A department is typically a division within an organization that refers to a specific responsibility. A line-of-business, or LOB, generally describes the products or services offered by a business.

You can also create cost centers when creating an organizational entity. A cost center is the smallest segment of an organization for which costs can be collected and reported.

Standard Mode

```
emcli create_organizational_entity
      -entity_name="entity name"
      -entity_type="entity type"
      [-parent_entity_name="parent entity name"]
      [-tenant_name="tenant name"]
      [-cost_centers="cost centers"[;"cost centers"...]]
      [-separator=argument_name="separator_value"])
```

```
[ ]  indicates that the parameter is optional.
```

Interactive or Script Mode

```
create_organizational_entity
      (entity_name="entity name"
      ,entity_type="department|lob"
      [,parent_entity_name="parent entity name"]
      [,tenant_name="tenant name"]
      [,cost_centers="cost centers"[;"cost centers"...]]
      [,separator=argument_name="separator_value"])
```

```
[ ]  indicates that the parameter is optional.
```

Options

- entity_name

  Name of the organizational entity to be created.

- entity_type

  Type of entity to be created. Value is either department or lob.

- parent_entity_name

  Specifies a parent of the organizational entity being created. The parent must already exist and can be either a department or LOB, regardless of the type being created. Default is no parent.

- tenant_name

  Specifies the name of the tenant to which the organizational entity being created belongs. Default is the tenant of the logged-in user.

- cost_centers

  Specifies one or more cost centers to create and associate with the organizational entity being created. Default is no cost centers. You can create cost centers and associate them independently, using the `create_cost_centers` verb.

- separator

Overrides the separator for multi-value input arguments, which is a semicolon (;). For information about overriding the separator or subseparator, see "Overriding the Separator and Subseparator".

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

The following example creates a department named finance with three cost centers, C1, C2, C3.

```
emcli create_organizational_entity
     -entity_name="finance"
     -entity_type="department"
     -cost_centers="c1;c2;c3"
```

Example 2

The following example creates a department named finance with three cost centers, C1, C2, C3, where the separator is changed to a comma (,).

```
emcli create_organizational_entity
     -entity_name="finance"
     -entity_type="department"
     -cost_centers="c1,c2,c3"
     -separator=cost_centers=","
```

create_paas_zone

Creates a PaaS Infrastructure Zone.

Format

```
emcli create_paas_zone
      -name="<PaaS Zone name>"
      -credential="<global named credential>"
      [-hosts="<Host1,Host2,Host3...>"]
      [-ovm_zones="<OVMZone1,OVMZone2,OVMZone3...>"]
      [-roles="<ssaRole1,ssaRole2,..>"]
      [-description="<PaaS Zone description>"]
      [-cpu_utilization="<value between 1 and 100>"]
      [-memory_utilization="<value between 1 and 100>"]

[ ]  indicates that the parameter is optional.
```

Options

- **name**

  Name of the PaaS Infrastructure Zone to be created.

- **credential**

  Global named credentials to be used for provisioning in this PaaS Infrastructure Zone. The credentials should be the same for all hosts. A cloud administrator can only use the named credentials that they own.

- **hosts**

  A comma-separated list of the host targets to be added as members of this Paas Infrastructure Zone.

- **ovm_zones**

  Comma-separated list of the Oracle Virtual Machine (OVM) Zone targets to be added as members of this Paas Infrastructure Zone. You must add at least one host or OVM Zone target for a PaaS Infrastructure Zone to be created.

- **roles**

  Comma-separated list of SSA roles that can access this PaaS Infrastructure Zone. A PaaS Infrastructure Zone can be made available to a restricted set of users through the use of roles. The SSA roles should already be created before executing this EM CLI command.

- **description**

  Description of the PaaS Infrastructure Zone.

- **cpu_utilization**

  Placement policy constraints enable the cloud administrator to set maximum ceilings for any host in the PaaS Infrastructure Zone. This constraint restricts the maximum resource consumption for the host members in a PaaS Infrastructure Zone. For example, a production PaaS Infrastructure Zone might limit CPU utilization to 80%, whereas a development PaaS Infrastructure Zone might allow up to 95 percent utilization. The service instance will be provisioned on the first host that satisfies the placement constraints. The value entered must be between 1 and 100. If not specified, the default value of 80% is used.

- **memory_utilization**

  Placement policy constraint for the PaaS Infrastructure Zone that restricts the percent of memory used. The value entered must be between 1 and 100. If not specified, the default value of 80% is used.

Example

This example creates a Paas Infrastructure Zone with the name My PaaS Zone:

```
emcli create_paas_zone
       -name="My PaaS Zone"
       -credential="ZoneNamedCredentials"
       -hosts="host1.mycompany.com, host2.mycompany.com"
       -roles="SSA_USER_ROLE"
       -description="This is a test PaaS Zone"
       -cpu_utilization="85"
       -memory_utilization="75"
```

Create a PDB Data Profile on Deletion

Format

```
emcli cleanup_dbaas_requests -ids=<requested> -preserve_backup_of_instance -
save_as="profile
       name " -description="profile description"
```

Options

- **-save_as**

  Profile component name

- **-description**

  Profile description

Example

```
emcli cleanup_dbaas_requests -ids=<requested> -preserve_backup_of_instance -
save_as="profile name " -description="profile description"
```

create_partition_profile

Format

```
emcli create_partition_profile
-name="Profile Name"
-ref_target="Reference Target Name"
-partition_name="Partition Name"
[-oh_cred="Oracle Home Owner Credentials"]
 [-wls_cred="WebLogic Administrator Credentials"]
 [-schedule=start_time:yyyy/MM/dd HH:mm;
 [tz:{java timezone ID}];
 [grace_period:xxx]; ]
 [-workDir="Working Directory Location"]
```

[ ] indicates that the parameter is optional.

Options

- **name**

  Name of the WebLogic Domain Partition Profile.

- **ref_target**

  Name of the reference WebLogic Domain target from which the partition will be exported.

- **partition_name**

  Name of the partition from which the profile will be created.

- **oh_cred**

  Named credential used to access the reference host. This is an optional parameter. To pass the credential parameter, enter a `name: value pair` in the following format:

  `credential_name:credential_owner`

  Credential_name is the name of the named credential.

  Credential_owner are the credentials of the Oracle home owner on the Administration Server host.

> **Note:**
>
> All the operations will be performed on the Administration Server host.
>
> If no named credential is provided, the preferred host credentials for the Oracle home target will be used.

- **wls_cred**

  Named credential used to access the Administration Server.

  This is an optional parameter. To pass the credential parameter, enter a `name: value pair` in the following format:

  ```
  credential_name:credential_owner
  ```

  Credential_name is the name of the named credential.

  Credential_owner are the credentials of the Administrator on the WebLogic Domain.

  > **Note:**
  >
  > If no named credential is provided, the preferred Administrator credentials for the domain target will be used.

- **schedule**

  Specify when to run the deployment procedure. If no value is entered, by default, the procedure runs immediately. To schedule a procedure, provide:

  - **start_time:** when the procedure should start.

  - **tz:** the timezone id.

  - **grace_period:** grace period in minutes.

- **workDir**

  Specify the working directory to be used. This is an optional parameter

Example 5-1   Examples

A WebLogic Domain Partition profile called **HRPartitionProfile** is created using the partition **HRPartition** from the domain **base_domain** at the specified schedule.

No credentials were specified, so the preferred credentials are used.

```
emcli create_fmw_domain_profile
-name="HRPartitionProfile"
-ref_target="/Farm01_base_domain/base_domain"
-partition_name="HRPartition"
-description="A partition profile for human resources"
-schedule="start_time:2016/6/21 21:23;tz:America/New_York;grace_period:60"
-workDir="/u01/mytemp"
```

create_patch_plan

Creates a new patch plan with the specified name and the patch-target map.

Format

```
emcli create_patch_plan
        -name="name"
        -input_file=data:"file_path"
        [-impact_other_targets="add_all | add_original_only | cancel"]
        [-problems_assoc_patches="ignore_all_warnings | cancel"]

[ ]  indicates that the parameter is optional
```

Options

* **name**

  Name of the setting.

* **input_file**

  Input data to create a new patch plan. You must provide the data in the property name-value pairs.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

* **impact_other_targets**

  Action to take when other targets are impacted while adding the patches to the plan. Possible values for this option are:

  add_all — Add all impacted targets to the plan. add_original_only — Only add original targets to the plan. cancel — Cancel the plan creation.

* **problems_assoc_patches**

  Action to take when there are problems associating patches to targets. Possible values for this option are:

  ignore_all_warnings — Ignore all warnings. cancel — Cancel the plan creation.

See Also

delete_patches
describe_patch_plan_input
get_connection_mode
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data
show_patch_plan
submit_patch_plan
upload_patches

Also see Patching Using EMCLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*:

Examples

```
emcli create_patch_plan -name="plan_name" -input_file=data:"/tmp/patchplan.props"

emcli create_patch_plan -name="plan name" -input_file=data:"/tmp/patchplan.props" -
impact_other_targets="add_all"
```

You can use the following sample input file to create a patch plan with two patches:

```
patch.0.patch_id=4518443
         patch.0.release_id=80102010
         patch.0.platform_id=226
         patch.0.language_id=0
         patch.0.target_name=orclws
         patch.0.target_type=oracle_database
         patch.1.patch_id=4424952
         patch.1.release_id=80102030
         patch.1.platform_id=46
         patch.1.language_id=0
         patch.1.target_name=arac
         patch.1.target_type=rac_database
```

create_pool

Creates a software pool.

Format

```
emcli create_pool
       -name="<software pool name>"
       -target_type="<software pool target type>"
       -paas_zone="<Paas Infrastructure Zone of software pool>"
       -members="<Member1, Member2...>"
       [-description="<software pool description>"]
       [-placement_constraints="<constraint1=value1, constraint2=value2...>"]
       [-member_constraints="<constraint1=value1, constraint2=value2>"]
       [-properties="<property1=value1, property2=value2>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **name**

  Name of the software pool to be created.

- **target_type**

  Target type of the software pool to be created, for example "mwaas_zone" for the middleware Pool, "oracle_cloud_zone" for the database pool, and "schaas_pool" for schema pool.

- **paas_zone**

  Name of PaaS Infrastructure Zone in which the software pool is to be created.

- **members**

Comma-separated list of targets to be added as members of the software pool. The targets to be added must satisfy the member constraints specified.

- **description**

  Description of the software pool.

- **placement_constraints**

  Comma-separated key-value pairs of the placement constraints that enable the self-service administrator to set maximum ceilings for resource utilization. This ability provides protection for the members of the software pool in terms of resource consumption. For example, a production software pool might enforce more conservative limits, whereas a development software pool might enforce more liberal limits.

- **member_constraints**

  Comma-separated key-value pairs that restrict the addition of member targets to a software pool with a set criteria. Execute "emcli get_pool_allowed_member_constraints -target_type=<Target type>" to retrieve the list of allowed possible member constraints for a pool target type.

- **properties**

  Comma-separated key-value pairs for additional properties that must be specified based on the pool target type.

Example

The following example creates the My Pool software pool:

```
emcli create_pool
      -name="My Pool"
      -target_type="mwaas_zone"
      -paas_zone="My PaaS Zone"
      -members="MyMember"
      -description="This is a test Pool"
      -placement_constraints="MAX_INSTANCES=20"
      -member_constraints="VERSION=10.3.5.0"
```

create_pluggable_database

Creates a pluggable database.

Format

```
emcli create_pluggable_database
        -cdbTargetName="Specify the CDB target name for creating new PDB"
        -cdbTargetType="Specify the CDB target type - oracle_database,
rac_database"
        -cdbHostCreds="Specify the host credentials on which the CDB target is
located. (owner:name)"
        [-cdbTargetCreds="Specify the credentials of container database on
which the new PDB will be created. (owner:name)"]
        -pdbName="Specify a name for the new PDB"
        -sourceType="Type of pdb to be created - DEFAULT, UNPLUGGED_PDB, CLONE,
PROFILE"
        [-sourcefromSWLIB="If -sourceType is 'UNPLUGGED_PDB', specify if the
dump location is SWLIB or not."]
        [-pdbTemplateInSWLIB="If -sourceFromSWLIB, specify the URN of pdb
template component in SWLIB."]
```

```
        [-sourcePDBTempStagingLocation="Specify fully qualified location for staging
temporary files. If not specified it will be defaulted to to "C:\Temp" in case of
Windows and "\tmp" otherwise."]
        [-unpluggedPDBType="If -sourceType is 'UNPLUGGED_PDB', specify pdb dump type -
 ARCHIVE, RMAN, XML."]
        [-sourcePDBArchiveLocation="If -unpluggedPDBType=ARCHIVE, this is fully
qualified archive location"]
        [-sourcePDBMetadataFile="If -unpluggedPDBType=RMAN or XML, this is fully
qualified path of the source PDB metadata file"]
        [-sourcePDBDataBackup="If -unpluggedPDBType=RMAN, this is fully qualified
path of the source PDB datafile"]
        [-moveDatafiles="If -unpluggedPDBType=XML, this will move the source PDB
datafiles to the destination"]
        [-excludeStandbys="If -unpluggedPDBType=XML and if 'moveDatafiles' is
specified, this will exclude the new PDB from standby CDBs"]
        [-sourcePDBName="If -sourceType is 'CLONE', specify the name of an existing
PDB which is a valid em target"]
        [-sourceCDBCreds="If -sourceType is 'CLONE', specify the credentials of
container database on which the -sourcePDBName is present. (owner:name)"]
        [-pdbAdminCreds="Name of pdb credentials with admin role. (owner:name)"]
        [-useOMF="Specifies that the datafiles can be stored in OMF location"]
        [-sameAsSource="Specifies that the datafiles of new PDB can be stored in the
same location as that of source CDB"]
        [-newPDBFileLocation="Specify the storage location for datafiles of the
created PDB."]
        [-createAsClone="If -sourceType is 'UNPLUGGED_PDB' and if 'createAsClone' is
specified, the PDB will be created as clone."]
        [-lockAllUsers="If -sourceType is 'UNPLUGGED_PDB' and if 'lockAllUsers' is
specified, all PDB users of the new PDB will be locked."]
        [-noUserTablespace="Specifies that the new DEFAULT PDB will not be created
with USERS tablespace."]
        [-useSnapClone="If -sourceType is 'CLONE', specify if Snap Clone should be
used for cloning"]
        [-sourceCDBHostCreds="If -sourceType is 'CLONE' and -useSnapClone is
specified, this is the host credentials for the source container database.
(owner:name)"]
        [-mountPointPrefix="If -sourceType is 'CLONE' and -useSnapClone is specified,
this is the mount point prefix for the clone volumes"]
        [-writableSpace="If -sourceType is 'CLONE' and -useSnapClone is specified,
this is the writable space in GB for the clone volumes"]
        [-saveProfile="If -sourceType is 'CLONE' and -useSnapClone is specified, -
saveProfile allows the created snapshot to be saved as profile"]
        [-profileName="If -saveProfile is specified, -profileName is required"]
        [-profileLocation="If -saveProfile is specified, -profileLocation is
required"]
        [-profileURN="If -sourceType is 'PROFILE', -profileURN is required"]
        [-privHostCreds="If -sourceType is 'CLONE' and -useSnapClone is specified,
this is the privileged host credentials to mount the volumes at the specified
locations. (owner:name)"]

[ ]  indicates that the parameter is optional
```

Options

- **cdbTargetName**

  Target container database for which pluggable database is to be created. It should be a
  valid target on enterprise manager.

- **cdbTargetType**

  Type of target container database.

- **cdbHostCreds**

  Credentials for the host on which the target container database is located. This is the name of the credentials saved in enterprise manager. To specify a credential owned by an user other than the current, use the form (owner:name).

- **cdbTargetCreds**

  Credentials for the container database on which the new PDB will be created. This is the name of the credentials stored in enterprise manager.

- **pdbName**

  Pluggable database name to be created. This works as a prefix in case of multiple pluggable database creation and will be suffixed with sequence number to generate pdb name <pdbname>#

- **numOfPdbs**

  Total number of pluggable databases to be created. The maximum number of new pluggable databases that are allowed in a given container database is 252 . If not specified, the default value is 1.

- **sourceType**

  Type of pluggable database to be created. The valid values are:

  – DEFAULT: This will create a pluggable database from seed pluggable database

  – UNPLUGGED_PDB: This will create a pluggable database from existing dump of an unplugged database like Archive, RMAN file set, or from XML

  – CLONE: This will create a pluggable database from an existing pluggable database

  – PROFILE: This will create a pluggable database from an existing PDB profile

- **sourceFromSWLIB**

  If the -sourceType is specified as 'UNPLUGGED_PDB' and the dump is available in software library, this flag must be set.

- **pdbTemplateInSWLIB**

  If the -sourceFromSWLIB, specify the valid URN of the pdb template in Software Library.

- **sourcePDBTempStagingLocation**

  Specify fully qualified location for staging temporary files. If not specified, the default location is "C:\Temp" in case of Windows and "\tmp" otherwise. Files are deleted after the operation.

- **unpluggedPDBType**

  If the -sourceType is specified as 'UNPLUGGED_PDB' and the dump is from file system, this specifies the type of dump to be used. The valid values are:

  – ARCHIVE: This will create a pluggable database from existing PDB archive

  – RMAN: This will create a pluggable database from existing PDB RMAN file set

  – XML: This will create a pluggable database from the metadata file

- **sourcePDBArchiveLocation**

If the -unpluggedPDBType is specified as ARCHIVE, this is the fully qualified location of PDB archive.

- **sourcePDBMetadataFile**

  If the -unpluggedPDBType is specified as RMAN or XML, this is the fully qualified location of the metadata file.

- **sourcePDBDataBackup**

  If the -unpluggedPDBType is specified as RMAN, this is the fully qualified location of datafile backup.

- **moveDatafiles**

  This parameter is applicable only if the parameter -unpluggedPDBType is specified as XML. This option indicates that the datafiles of the source pluggable database should be moved to the destination datafiles location.

- **excludeStandbys**

  This parameter is applicable only if the parameter -unpluggedPDBType is specified as XML and if 'moveDatafiles' is indicated. This option will exclude the new pluggable database from all standby container databases (STANDBYS=NONE).

- **sourcePDBName**

  If -sourceType is specified as clone, this specifies the name of the pluggable database from which the new pluggable database will be cloned from. This has to be a valid target in enterprise manager.

- **sourceCDBCreds**

  If -sourceType is specified as clone, provide the credentials for the container database on which the -sourcePDBName is present. This is the name of the credentials stored in enterprise manager.

- **useOMF**

  Destination of the datafiles for the new pluggable database will be the OMF location. This is valid only if the source CDB is OMF.

- **sameAsSource**

  Destination of the datafiles for the new pluggable database will be same as CDB. In case of multi PDB creation this option is invalid.

- **newPDBFileLocation**

  Destination of the datafiles for the new pluggable database.

- **pdbAdminCreds**

  New pluggable database admin credentials. This is a mandatory parameter, if the source of new pluggable database is DEFAULT, for other options, it is optional.

- **createAsClone**

  This flag specifies whether the new pluggable database should be created as clone, if the -sourcePDBType is specified as 'UNPLUGGED_PDB' and is used by default in case of multiple pluggable database creation.

- **lockAllUsers**

  This flag specifies whether all the users should be locked in case of -sourcePDBType being 'UNPLUGGED_PDB'.

- **noUserTablespace**

This flag specifies whether new DEFAULT PDB should be created without USER tablespace. This is applied only if the -sourceType is DEFAULT.

- **useSnapClone**

  This flag specifies whether Snap Clone can be used for cloning the PDB. This is applied only if the -sourceType is CLONE.

- **sourceCDBHostCreds**

  This is the host credentials to get storage information of the source container database. This is applied only if the -sourceType is CLONE and -useSnapClone is specified.

- **mountPointPrefix**

  This is the mount point prefix for the clone volumes. This is applied only if the -sourceType is CLONE and -useSnapClone is specified.

- **writableSpace**

  This is the writable space in GB for the clone volumes. This is applied only if the -sourceType is CLONE and -useSnapClone is specified.

- **saveProfile**

  This allows the snapshot created for the source PDB to be saved as a profile. This is applied only if the -sourceType is CLONE and -useSnapClone is specified.

- **profileName**

  This specifies the name of the profile to be created. This is applied only if -saveProfile is specified.

- **profileLocation**

  This specifies the location under software library where the created profile will be saved. This is applied only if -saveProfile is specified.

- **profileURN**

  This specifies the URN of the profile from which PDB will be created. This is applied only if the -sourceType is PROFILE.

- **privHostCreds**

  This is the privileged host credentials to mount the volumes at the specified locations. This is applied only if the -sourceType is CLONE and -useSnapClone is specified.

Examples

Example 1

```
emcli create_pluggable_database
        -cdbTargetName=database
        -cdbTargetType=oracle_database
        -pdbName=pdb -sourceType=UNPLUGGED_PDB
        -unpluggedPDBType=ARCHIVE
        -sourcePDBArchiveLocation=/u01/app/oracle/product/12.1.0/dbhome_2/
assistants/dbca/templates/a.tar.gz
        -cdbHostCreds=HOST_CREDS
        -cdbTargetCreds=DBSNMP
        -newPDBFileLocation=/u01/app/oradata/pdb
        -pdbAdminCreds=pdb_creds
        -lockAllUsers
```

Example 2

```
emcli create_pluggable_database
        -cdbTargetName=database
        -cdbTargetType=oracle_database
        -pdbName=pdb -numOfPdbs=1
        -sourceType=CLONE
        -cdbHostCreds=SYSMAN:HOST_CREDS
        -cdbTargetCreds=SYSMAN:DBSNMP
        -sourcePDBName=source_pdb
        -sourceCDBCreds=CDB_SYS_CREDS
        -useSnapClone
        -srcCDBHostCreds=SYS
        -mountPointPrefix=/mount/point/prefix -writableSpace=1
        -sourcePDBTempStagingLocation=/tmp
        -privHostCreds=HOST_SUDO
        -saveProfile -profileName=PROFILENAME
        -profileLocation=MyProfiles
```

Example 3

```
emcli create_pluggable_database
        -cdbTargetName=database
        -cdbTargetType=oracle_database
        -pdbName=pdb -numOfPdbs=1
        -sourceType=PROFILE
        -cdbHostCreds=HOST_CREDS
        -cdbTargetCreds=DBSNMP
        -mountPointPrefix=/mount/point/prefix
        -writableSpace=1
        -sourcePDBTempStagingLocation=/tmp
        -privHostCreds=HOST_SUDO -
profileURN=oracle:defaultService:em:provisioning:1:PROFILE:URN:RANDOM:0.1
```

Example 4

```
emcli create_pluggable_database
        -cdbTargetName=database
        -cdbTargetType=oracle_database
        -pdbName=pdb -numOfPdbs=2
        -sourceType=UNPLUGGED_PDB -unpluggedPDBType=RMAN
        -sourcePDBMetadataFile=/u01/app/oracle/product/12.1.0/dbhome_2/assistants/
dbca/templates/a.xml
        -sourcePDBDataBackup=/u01/app/oracle/product/12.1.0/dbhome_2/assistants/dbca/
templates/a.dfb
        -cdbHostCreds=HOST_CREDS
        -cdbTargetCreds=DBSNMP
        -newPDBFileLocation=/u01/app/oradata/pdb
        -pdbAdminCreds=pdb_creds
        -createAsClone
```

Example 5

```
emcli create_pluggable_database
        -cdbTargetName=database
        -cdbTargetType=oracle_database
        -cdbHostCreds=HOST_CREDS
        -cdbTargetCreds=CDB_SYS_CREDS
        -pdbName=pdb
        -sourceType=CLONE
        -sourcePDBName=source_pdb
        -sourceCDBCreds=CDB_SYS_CREDS
```

```
                -useSnapClone -srcCDBHostCreds=HOST_CREDS
                -mountPointPrefix=/oracle -writableSpace=1
                -sourcePDBTempStagingLocation=/tmp
                -privHostCreds=ROOT_CREDS
```

create_privilege_delegation_setting

Creates a privilege delegation setting template to apply later. You must create at least one setting to use the `apply_privilege_delegation_setting` verb.

Standard Mode

```
 emcli create_privilege_delegation_setting
         -setting_name="name"
         -setting_type="ttype"
         [-settings="setting"]
         [-separator=settings=";"]
         [-subseparator=settings=","]
```

```
[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
 create_privilege_delegation_setting
         (setting_name="name"
         ,setting_type="ttype"
         [,settings="setting"]
         [,separator=settings=";"]
         [,subseparator=settings=","])
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **setting_name**

    Name of the privilege delegation setting template.

*   **setting_type**

    Type of setting you want to create.

*   **settings**

    Parameter value. Choose one of the following parameters:

    %USERNAME% — Name of the user running the command. %RUNAS% — Run the command as this user. %COMMAND% — Sudo command.

    The %USER%, %RUNAS%,, %COMMAND% are tokens that the end-user has to use as-is while creating/modifying the privilege delegation settings. The system replaces these tokens with the actual values at run time depending on the command being run and for which user. Also, %command% should be upper case %COMMAND% for 10.2.0.5 GC.

*   **separator**

    Delimiter inserted between name-value pairs for the given name. The default value is a semi-colon (;).

    For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **subseparator**

  Separator inserted between the name and value in each name-value pair for the given name. The default value is a semi-colon (;).

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

Examples

Example 1

These examples create a setting named `sudo_setting`. The setting is of type SUDO, and the Sudo path used is `/usr/local/bin/sudo`. Sudo arguments are:

`-S` -u %RUNAS% %COMMAND%

```
emcli create_privilege_delegation_setting
    -setting_name=sudo_setting
    -setting_type=SUDO
    -settings="SETTINGS:/usr/local/bin/sudo -S -u %RUNAS% %COMMAND%"
```

Example 2

This example creates a setting named `pb_setting`. The setting is of type POWERBROKER, and the PowerBroker path used is `/etc/pbrun`. Arguments are:

```
%RUNAS% %PROFILE% %COMMAND%
```

```
emcli create_privilege_delegation_setting
    -setting_name="pb_setting"
    -setting_type="POWERBROKER"
    -settings="SETTINGS,/etc/pbrun %RUNAS% %PROFILE% %COMMAND%"
    -separator="settings=;"
    -subseparator="settings=,"
```

create_rbk

Creates a retroactive blackout or retroactive outage on given targets and updates their availability. Only Enterprise Manager Administrators with OPERATOR privilege on the target can perform this action.

For planned outages, where the administrator forgot to set a blackout, create a retroactive blackout without the -outage option. This increases the target's availability %. For example, 84% ==> 100%.

For unplanned outages, where Enterprise Manager did not detect the outage, create a retroactive blackout with the -outage option. This decreases the target's availability %. For example. 100% ==> 84%.

A retroactive blackout or retroactive outage can be created either from the console or using EM CLI. In both cases, it is necessary to enable retroactive blackout first through the user interface (UI). For more information on how to enable retroactive blackout from the UI, see *Enterprise Manager Cloud Control Administrator's Guide.*

Format

```
emcli create_rbk
    -add_targets="name1:type1;name2:type2;..."...
      -reason="reason"
    [-propagate_targets]
```

```
-schedule=
    start_time:<yyyy-MM-dd HH:mm:ss>;
    end_time:<yyyy-MM-dd HH:mm:ss>;
    [tzregion:<...>;]
[-outage]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **add_targets**

  Targets to add to the blackout. Each target is specified as target_name:target_type. You can specify this option more than once.

- **reason**

  Reason for the retro-active blackout. This is used for storing in backup tables.

- **propagate_targets**

  A blackout for a target of type "host" applies the blackout to all non-agent targets on the host. Regardless of whether this option is specified, a blackout for a target that is a composite or a group applies the blackout to all members of the composite or group.

- **schedule**

  Schedule for retroactive blackout. The following arguments are mandatory for providing a retroactive blackout schedule:

  – schedule=start_time - The start date/time of the blackout. The format of the value is "yyyy-MM-dd HH:mm:ss". For example: "2013-09-20 12:12:12"

  – schedule=end_time - The end date/time of the blackout. The format of the value is "yyyy-MM-dd HH:mm:ss". For example: "2013-09-20 12:15:00"

  – schedule=tzregion - The timezone region to use. For example: "UTC". If not provided, tzregion is set to UTC by default.

- **outage**

  Use this option with caution as it will lower the target availability (%). This option should be used only if Enterprise Manager does not detect the outage.

Examples

Example 1

This example creates a retroactive blackout on Oemrep_Database and updates the target's availability record from 2013-09-20 12:12:12 UTC to 2013-09-20 12:15:00 UTC as the blackout. This will result in an increased target availability %.

```
emcli create_rbk -reason="Testing"
    -add_targets="Oemrep_Database:oracle_database"
    -schedule="start_time:2013-09-20 12:12:12;end_time:2013-09-20
     12:15:00;tzregion:UTC"
```

Example 2

This example creates a retroactive blackout for all targets on host example.company.com and updates their availability records from 2013-09-20 12:12:12 UTC to 2013-09-20 12:15:00 UTC as the blackout. This will result in an increased target availability %.

```
emcli create_rbk -reason="Testing"
     -add_targets="example.company.com:host"
     -schedule="start_time:2013-09-20 12:12:12;end_time:2013-09-20
      12:15:00;tzregion:UTC"
     -propagate_targets
```

Example 3

This example creates a retroactive outage on adidev.company.com target. This will result in a decreased target availability %.

```
emcli create_rbk
    -add_targets="adidev.company.com:oracle_database"
    -reason="Testing"
    -propagate_targets
    -schedule="start_time:2017-11-02 9:25:10;end_time:2017-11-02 11:20:24;tzregion:PST"
    -outage
```

create_red_group

Defines a redundancy group name and its members. After you create the redundancy group, you can edit it from the Enterprise Manager Cloud Control console to configure charts to be displayed for redundancy group members.

Format

```
emcli create_red_group
     -name="name"
     [-type=<generic_redundancy_group>]
     -add_targets="name1:type1;name2:type2;..."...
     [-owner=<redundancy_group_owner>]
     [-timezone_region=<actual_timezone_region>]

[ ]  indicates that the parameter is optional
```

Options

*   **name**

    Name of the redundancy group.

*   **type**

    Redundancy group type. Defaults to `generic_redundancy_group`.

*   **add_targets**

    Add existing targets to the redundancy group. Each target is specified as a name-value pair `target_name:target_type`. You can specify this option more than once.

*   **owner**

    Owner of the redundancy group.

*   **timezone_region**

    Time zone region of this redundancy group.

Example

This example creates a redundancy group named `lsnr_group`. This group consists of two Oracle listeners: `emp_rec` and `payroll`.

```
emcli create_red_group -name=lsnr_group
      -add_targets="emp_rec:oracle_listener"
      -add_targets="payroll:oracle_listener"
```

create_redundancy_group

Creates a redundancy group.

Format

```
 emcli create_redundancy_group        -redundancyGroupName="redGrpName"       -
memberTargetType="tType"        -memberTargetNames="tName1;tName2"        [-
group_status_criterion=NUMBER|PERCENTAGE]       [-group_status_tracked=UP|
DOWN]        [-group_status_value=<group_status_value>]       [-
timezone_region=<valid_time_zone_region>]
        [is_propagating=true|false]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **redundancyGroupName**

    Name of the redundancy group.

*   **memberTargetType**

    Target type of the constituent member targets.

*   **memberTargetNames**

    Member targets for this redundancy group.

*   **group_status_criterion**

    this optionand the next two calculate the status of the Redundancy Group.
    Consequently, you need to specify all three options together. If this is not to be a
    capacity group, you need to specify the following combination:

    ```
    -group_status_criterion='NUMBER' -group_status_tracked='UP' -
    group_status_value='1']
    ```

*   **group_status_tracked**

    See the parameter above.

*   **group_status_value**

    See the group_status_criterion parameter.

    You can specify any value between 1 and 100 if -group_status_criterion=
    "PERCENTAGE", or any value between 1 and the number of targets present if -
    group_status_criterion="NUMBER".

*   **timezone_region**

    Time zone region of this redundancy group. For a list of valid time zone regions,
    enter the following command at SQLPLUS:

    ```
    SELECT TZNAME FROM V$TIMEZONE_NAMES
    ```

    You may need to have the SELECT_CATALOG_ROLE role to execute this
    command.

*   **is_propagating**

Indicates whether or not the privilege on the redundancy group will be propagated to member targets. The default value is false.

Examples

Example 1

This example creates a redundancy group with the name 'redGrp1' and with listener, listener2, listener3 as its member targets. The status is calculated as the redundancy group being up if 55 percent of its member targets are up.

```
    emcli create_redundancy_group -redundancyGroupName='redGrp1'           -
memberTargetType='oracle_listener'           -
memberTargetNames='listener;listener2;listener3'           -
group_status_criterion='PERCENTAGE'           -group_status_tracked='UP'           -
group_status_value='55'
```

Example 2

This example creates a 'redGrp1' redundancy group with listener, listener2, and listener3 as its member targets and time zone as PST8PDT. The status is calculated as the redundancy group being up if two of its member targets are up.

```
emcli create_redundancy_group -redundancyGroupName='redGrp1'
        -memberTargetType='oracle_listener'
        -memberTargetNames='listener;listener2;listener3'
        -timezone_region='PST8PDT'
        -group_status_criterion='NUMBER'
        -group_status_tracked='UP'
        -group_status_value='2'
```

create_resolution_state

Creates a new resolution state that describes the state of incidents or problems. Only super administrators can execute this command. The new state is always added between the New and Closed states. You need to specify the exact position of this state in the overall list of states by using the position option. The position can be between 2 and 98.

The state is applicable by default to both incidents and problems. You can use the applies_to option to indicate that the state is applicable only to incidents or problems. A success message is reported if the command is successful. An error message is reported if the create fails.

Format

```
emcli create_resolution_state
      -label="label_for_display"
      -position="display_position"
       [-applies_to="INC|PBLM"]

[ ]  indicates that the parameter is optional
```

Options

- **label**

  End-user visible label of the state. The label cannot exceed 32 characters. You can change this later if needed.

- **position**

Position of this state within the overall list of states. This is used when displaying the list of states in the user interface. The position can be between 2 and 98. You can change the position of the state later if needed.

It is recommended that you set the position with sufficient gaps to facilitate moving states around. For example, if you set the positions to 5, 10, and 15 instead of 2, 3, and 4, it is easier to move a state from position 15 to 9, for instance, in contrast to the latter scheme, in which you would have to move all states to provide space for the reordering.

- **applies_to**

  Indicates that the state is applicable only for incidents or problems. By default, states apply to both incidents and problems. Supported values are "INC" or "PBLM".

Examples

Example 1

This example adds a resolution state that applies to both incidents and problems at position 25.

```
emcli create_resolution_state -label="Waiting for Ticket" -position=25
```

Example 2

This example adds a resolution state that applies to problems only at position 35.

```
emcli create_resolution_state -label="Waiting for SR" -position=35 -
applies_to=PBLM
```

create_role

Creates a new Enterprise Manager admininistrator role.

Standard Mode

```
emcli create_role
      -name="role_name"
      [-type="type_of_role"]
      [-description="description"]
      [-roles="role1;role2;..."]
      [-users="user1;user2;..."]
      [-privilege="name[;secure_resource_details]]"
      [-separator=privilege="sep_string"]
      [-subseparator=privilege="subsep_string"]

[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
create_role
      (name="role_name"
      [,type="type_of_role"]
      [,description="description"]
      [,roles="role1;role2;..."]
      [,users="user1;user2;..."]
      [,privilege="name[;secure_resource_details]]"
      [,separator=privilege="sep_string"]
```

```
        [,subseparator=privilege="subsep_string"]
        )
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Role name.

- **type**

  Type of role. The default value for this option is EM_ROLE. Other possible values for this parameter are EM_ROLE and EXTERNAL_ROLE.

- **description**

  Description of the role.

- **roles**

  List of roles to assign to this new role. Currently, the only built-in role is PUBLIC.

- **users**

  List of users to whom this role is assigned. If the role must be granted with the WITH_ADMIN option, include the <subseparator:>WITH_ADMIN option.

- **privilege**

  Privilege to grant to this role. You can specify this option more than once. **Note:** Privileges are case-insensitive.

  Specify <secure_resource_details> as follows:

```
resource_guid|
[resource_column_name1=resource_column_value1[:resource_column_name2=resource_colum
n_value2]..]"
```

  To get the list of SYSTEM privileges, which do not require resource information, execute the following emcli command:

```
emcli get_supported_privileges -type=SYSTEM
```

  To get the complete list of privileges and resource column names, execute the following emcli command:

```
emcli get_supported_privileges
```

  To get the list of target type privileges, execute the following emcli command:

```
emcli get_supported_privileges -type=TARGET
```

  To get the list of job privileges, execute the following emcli command:

```
emcli get_supported_privileges -type=JOB
```

- **separator**

  Specify a string delimiter to use between name-value pairs for the value of the privilege option. The default separator delimiter is ";" .

  For example: separator="<*attribute_name=sep_char*>"

  where *attribute_name* is the name of the attribute for which you want to override the separator character and *sep_char* is the new separator character: separator="att=#"

- **subseparator**

  Specify a string delimiter to use between name and value in each name-value pair for the value of the privilege option. The default separator delimiter is ";" .

  For example: subseparator="<*attribute_name=sep_char*>"

  where *attribute_name* is the name of the attribute for which you want to override the separator character and *sep_char* is the new subseparator character: subseparator="att=#"

  For information about overriding the separator or subseparator, see "Overriding the Separator and Subseparator".

Examples

These examples create a role named `my_new_role` with the one-sentence description - "This is a new role called `my_new_role`". The role combines three existing roles: `role1`, `role2`, and `role3`. The role also has two added privileges: to view the job with ID 923470234ABCDFE23018494753091111 and to view the target `host1.example.com:host`. The role is granted to `johndoe` and `janedoe`.

Example 1 - Command-Line

```
emcli create_role
      -name="my_new_role"
      -desc="This is a new role called my_new_role"
      -roles="role1;role2;role3"
      -privilege="view_job;923470234ABCDFE23018494753091111"
      -privilege="view_target;host1.example.com:host"
      -users="johndoe;janedoe"
```

Example 2 - Scripting and Interactive

```
create_role
      (name="my_new_role"
      ,desc="This is a new role called my_new_role"
      ,roles="role1;role2;role3"
      ,privilege="view_job;923470234ABCDFE23018494753091111"
      ,privilege="view_target;host1.example.com:host"
      ,users="johndoe;janedoe")
```

These examples create a role named my_external_role with a role type of EXTERNAL_ROLE and one-sentence description of "This is an external role."

Example 3 - Command-Line

```
emcli create_role
      -name="my_external_role"
      -type="EXTERNAL_ROLE"
      -desc="This is an external role"
```

Example 4 - Scripting and Interactive

```
create_role
      (name="my_external_role"
      ,type="EXTERNAL_ROLE"
      ,desc="This is an external role")
```

create_service

Creates a service to be monitored by Enterprise Manager.

Format

```
emcli create_service
     -name='name'
     -type='type'
     -availType=test|system
     -availOp=and|or
     [-hostName=<host_name>
     [-agentURL=<agent_url>
     [-properties='pname1|pval1;pname2|pval2;...']
     [-timezone_region=<gmt_offset>]
     [-systemname=<system_name>]
     [-systemtype=<system_type>]
     [-keycomponents='keycomp1name:keycomp1type;keycomp2name:keycomp2type;...']
     [-beacons='bcn1name:bcn1isKey;bcn2name:bcn2isKey;...']
     [-input_file="template:Template file name]"]
     [-input_file="variables:Variable file name]"]
     [-sysAvailType=<availability_type>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Service name. Names cannot contain colons (:), semi-colons (;), or any leading or trailing blanks.

- **type**

  Service type.

- **availType**

  Sets the availability to either test-based or system-based. If availability is set to `test`, template file, beacons, and variable are required arguments. If availability is set to `system`, systemname, systemtype, and keycomponents are required.

- **availOp**

  Availability operator. If `and`, uses all key tests/components to decide availability. If `or`, uses any key tests/components to decide availability.

- **hostName**

  Network name of the system running the Management Agent that is collecting data for this target instance.

- **agentURL**

  URL of the Management Agent that is collecting data for this target instance. If you enter the host name, the Agent URL of the host is automatically entered in this field.

- **properties**

  Name-value pair (that is, prop_name|prop_value) list of properties for the service instance.

- **timezone_region**

GMT offset for this target instance (-7 or -04:00 are acceptable formats).

- **systemname**

  System name on which service resides.

- **systemtype**

  Type of system for which you want to create the service.

- **keycomponents**

  Name-type pair (that is, `keycomp_name:keycomp_type`) list of key components in the system that are used for the service.

- **beacons**

  Name-isKey pairs that describe the beacons of the service. If isKey is set to `Y`, beacon is set as a key-beacon of the service. The service should have at least one key beacon if the availability is set to test-based.

- **input_file**

  Template file name is the XML file that includes the template definition. Variable file defines the values for the template.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **sysAvailType**

  Type of availability when the availType is system-based. Sets the availability to either system target directly or selected components of a system.

  If availability is set to 'system target directly,' the system needs to have availability[status] defined. systemname and systemtype are required parameters.

  If availability is set to 'selected components of a system,' systemname, systemtype and keycomponents are required parameters.

  If availability is set to 'system target directly,' and if the system does not have availability[status] defined, the availability set is invalid. Therefore, the only option that can be set is 'selected components of a system'.

Examples

Example 1

This example creates a generic service named my_service with specified properties on a generic system named my_system. The availability is set as system-based, and the availability is based on system target status.

```
emcli create_service
     -name='my service' -type='generic_service'
     -availType='system' -availOp='or'
               -sysAvailType='system target directly'
     -properties='prop1:value1; prop2:value2'
     -timezone_region='PST8PDT'
     -systemname='my system' -systemtype='generic_system'
```

Example 2

This example creates a generic service named `my_service` with specified properties on a generic system named `my system` with specified key components. The availability is set as system-based.

```
emcli create_service
      -name='my_service' -type='generic_service'
      -availType='system' -availOp='or'
      -properties='prop1:value1; prop2:value2'
      -timezone_region='PST8PDT'
      -systemname='my system' -systemtype='generic_system'
      -keycomponents='database:oracle_database; mytestbeacon:oracle_beacon'
```

create_service_template

Creates a service template.

Format

```
emcli create_service_template
      -name="<service template>"
      -service_family="<service family>"
      -service_type="<service type>"]
      -pool_target_type="target type of software pools"
      -software_pools="<SwPool1,SwPool2,SwPool3,...>"
      [-roles="<SsaRole1,SsaRole2,..>"]
      [-description="<service template description>"]
      [-input_file="data:<service executable metadata file>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **name**

  Name of the service template to be created.

- **service_family**

  Service family for which the service template is being created, for example DBAAS for database, MWAAS for middleware.

- **service_type**

  Service type for which the service template is being created, for example PhysicalWLS for a physical middleware service template.

- **pool_target_type**

  Target type of the software pools to be associated with the service template.

- **software_pools**

  Comma-separated list of software pools to be associated with the service template.

- **roles**

  Comma-separated list of SSA roles that can access this service template. A service template can be made available to a restricted set of users through the use of roles. The SSA roles should already be created before executing this EM CLI command.

- **description**

  Description of the service template.

- **input_file**

  Contains configuration and profile data in JSON format that will be required for setting values of procedure configuration variables. For example:

```
       input_file="data:executable.json"
```

Example

This example creates the service template My Service Template:

```
emcli create_service_template
       -name="Middleware service template August"
       -service_family="MWAAS"
       -service_type="PhysicalWLS"
       -pool_target_type="mwaas_zone"
       -software_pools="MyPoolOH"
       -roles="SSA_USER_ROLE"
       -description="Middleware small instance service template."
       -input_file="data:executable.json"
```

create_siteguard_configuration

Creates a site configuration for Site Guard. It associates the systems and their roles.

Format

```
emcli create_siteguard_configuration
       -primary_system_name=<name>
       -standby_system_name=<name1;name2;...>
```

Options

* **primary_system_name**

  Name of the system associated with the primary site.

* **standby_system_name**

  Name of the system associated with the standby system. You can specify more than one system name.

Examples

```
emcli create_siteguard_configuration
       -primary_system_name="BISystem1"
       -standby_system_name="BISystem2"
```

See Also

update_siteguard_configuration
delete_siteguard_configuration

create_siteguard_credential_association

Associates the credentials with the targets in a site.

Format

```
emcli create_siteguard_credential_association
       -system_name=<name>
       [-target_name=<name>]
       -credential_type=<type>
       [-credential_name=<name>]
       [-use_preferred_credential=<type>]
```

```
            -credential_owner=<owner>
```

[ ] indicates that the parameter is optional.

Options

* **system_name**

  Name of the system.

* **target_name**

  Name of the target.

* **credential_type**

  Type of credential, which can be HostNormal, HostPrivileged, WLSAdmin, or DatabaseSysdba.

* **credential_name**

  Name of the credential. If you do not specify this option, you need to specify the use_preferred_credential parameter.

* **use_preferred_credential**

  Name of the credential. If you do not specify this option, you need to specify the credential_name parameter.

* **credential_owner**

  Owner of the credential.

Examples

Example 1

```
emcli create_siteguard_credential_association
      -system_name="BISystem1"
      -credential_type="HostNormal"
      -credential_name="HOST-SGCRED"
      -credential_owner="sysman"
```

Example 2

```
emcli create_siteguard_credential_association
      -system_name="BIsystem1"
      -target_name="database-instance"
      -credential_type="HostNormal"
      -credential_name="HOST-DBCRED"
      -credential_owner="sysman"
```

create_siteguard_script

Associates scripts (pre-script, post-script, and storage script) with the Site Guard configuration.

Format

```
emcli create_siteguard_script
      -system_name=<name>
      -operation=<name>
      -script_type=<type>
      [-host_name=[<name1;name2;...>]
```

```
-path=<path_of_script>
[-all_hosts=true|false]
[-role=Primary|Standby]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **system_name**

  Name of the system.

- **operation**

  Name of the operation. Examples: Switchover, Failover, Start, or Stop.

- **script_type**

  Type of script, which can be Mount, UnMount, Pre-Script, Post-Script, Failover, or Switchover.

- **host_name**

  Name of the host where this script will run. You can specify this option more than once.

- **path**

  Path to the script.

- **all_hosts**

  Allows the script to run on all the hosts in the system. This optionoverrides the host_name.

- **role**

  Configures the script based on the system role. By default, the script is configured for both primary and standby roles for a given system.

Examples

Example 1

```
emcli create_siteguard_script
    -system_name="BISystem1"
    -operation="Switchover"
    -script_type="Pre-Script"
    -path="/tmp/prescript"
    -all_hosts="true"
    -role="Primary"
```

Example 2

```
emcli create_siteguard_script
    -system_name="BISystem1"
    -operation="Switchover"
    -script_type="Pre-Script"
    -path="/tmp/prescript"
    -host_name="BIHOST1"
    -host_name="BIHOST2"
```

create_standby (deprecated)

Creates a standby database generated from a backup of a primary database.

> **✎ Note:**
>
> Since create_standby is deprecated, use create_standby_database instead.

Format

```
emcli create_standby
        -source_db_target_name="<standby database target name>"
      -source_db_target_type="oracle_database|rac_database"
      -dest_oracle_sid="<standby instance name>"
      -spname="<standby database unique name>"
      [-source_db_creds_name="primary database credential name"]
         [-source_host_creds_name="primary database host credential name"]
         [-dest_host_creds_name="standby database host credential name"]
         [-asm_inst_creds_name="asm instance credential name"]
          [-dest_host_name="standby host name"]
         [-dest_oracle_home="standby database oracle home"]
         [-dest_target_name="standby database target name"]
         [-use_duplicate=\"Yes|No\"]
         [-source_staging_area="primary staging directory"]
         [-storage_type="storage type"]
         [-dest_db_database_area="standby database files location"]
         [-dest_db_recovery_area="standby database fast recovery area"]
         [-dest_listener_selection="standby database listener selection"]
         [-dest_listener_name="standby database listener name"]
         [-dest_listener_port="standby database listener port"]
         [-stby_type="standby type"]
         [-use_broker=\"Yes|No\"]
         [-use_sys_dba_monitoring_creds="use sys dba monitoring creds"]
         [-dest_staging_area="standby staging directory"]
         [-configure_with_oracle_restart]
```

Options

- source_db_target_name

  Primary database Enterprise Manager target name source. Can either be a single-instance database or a cluster database instance.

- source_db_target_type

  Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- dest_oracle_sid

  Standby database instance name.

- spname

  Standby database unique name.

- source_db_creds_name

Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- source_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle Home. Default is to use preferred credential.

- dest_host_creds_name

  Standby database host named credential for an operating system user who can access the primary database Oracle Home. Default is to use preferred credential.

- asm_inst_creds_name

  ASM instance named credential.

- dest_host_name

  Standby database host name. Default is primary host name.

- dest_oracle_home

  Standby database Oracle Home location. Default is primary database Oracle Home location.

- dest_target_name

  Standby database Enterprise Manager target name. Default is standby database unique name.

- use_duplicate

  Database files moved directly to clone the database host by the Recovery Manager(RMAN). Default is yes.

- source_staging_area

  Staging area used to store the backup of a primary database. This option is applicable only when use_duplicate is not set.

- storage_type

  Standby database storage type.

  – FILE_SYSTEM: Standby database files are in a regular file system.

  – ASM_STORAGE: Standby database uses Automatic Storage Management (ASM).

  Default is FILE_SYSTEM.

- dest_db_database_area

  Standby database files Oracle-managed files (OMF) location. Can be a regular file system (if storage_type is FILE_SYSTEM) or an ASM diskgroup (if storage_type is ASM_STORAGE).

- dest_db_recovery_area

  Standby database fast recovery area.

- dest_listener_selection

  Standby database listener selection.

  – GRID_INFRA: Uses the Grid Infrastructure Home listener.

  – DEST_DB_HOME: Uses the listener from the standby database Oracle Home.

Default is GRID_INFRA.

- dest_listener_name

  Standby database listener name. This option is applicable only if dest_listener_selection is set to DEST_DB_HOME.

  If not specified, default value is the first existing TCP listener found in the standby database Oracle Home. Note that if dest_listener_name is specified, then dest_listener_port must also be specified.

- dest_listener_port

  Standby database listener port. This option is applicable only if dest_listener_selection is set to DEST_DB_HOME. Note that if dest_listener_port is specified, then dest_listener_name must also be specified.

- stby_type

  Type of the standby database to be created.

  – PHYSICAL

  – LOGICAL

  Default is PHYSICAL.

- use_broker

  Uses Data Guard broker to manage the Data Guard configuration. Default is yes.

- use_sys_dba_monitoring_creds

  Uses SYSDBA credentials to monitor the standby database. Default is no.

- dest_staging_area

  Staging area used to store the backup files transferred from the primary host. This option is applicable only when use_duplicate is not set.

- configure_with_oracle_restart

  If the destination host has Oracle Restart configured, it configures the standby database with Oracle Restart. When required, Oracle Restart automatically starts the standby database. Default is no.

Examples

Example 1

The following command creates a standby database with the unique name "database1" generated from the backup of a primary single-instance database named "Database".

```
emcli create_standby
     -source_db_target_name="database"
     -source_db_target_type="oracle_database"
     -dest_oracle_sid="database1"
     -spname="database1"
```

Example 2

The following command creates a standby database with the unique name "database1" generated from the backup of a cluster database named "primary". The standby database uses SYSDBA credentials for monitoring and uses Data Guard broker to manage the Data Guard configuration.

```
emcli create_standby
      -source_db_target_name="primary"
      -source_db_target_type="rac_database"
      -dest_oracle_sid="database1"
      -spname="database1"
      -use_broker="Yes"
      -use_sys_dba_monitoring_creds="Yes"
```

create_standby_database

Creates a standby database generated from a backup of a primary database.

Format

```
emcli create_standby_database
      -source_db_target_name="<standby database target name>"
      -source_db_target_type="oracle_database|rac_database"
      -dest_oracle_sid="<standby instance name>"
      -spname="<standby database unique name>"
      [-source_db_creds_name="<primary database credential name>"]
      [-source_host_creds_name="<primary database host credential name>"]
      [-dest_host_creds_name="<standby database host credential name>"]
      [-asm_inst_creds_name="<asm instance credential name>"]
      [-dest_host_name="<standby host name>"]
      [-dest_oracle_home="<standby database Oracle home>"]
      [-dest_target_name="<standby database target name>"]
      [-use_duplicate="Yes|No"]
      [-source_staging_area="<primary staging directory>"]
      [-storage_type="<storage type>"]
      [-dest_db_database_area="<standby database files location>"]
      [-dest_db_recovery_area="<standby database fast recovery area>"]
      [-dest_listener_selection="<standby database listener selection>"]
      [-dest_listener_name="<standby database listener name>"]
      [-dest_listener_port="<standby database listener port>"]
      [-stby_type="<standby type>"]
      [-use_broker="Yes|No"]
      [-use_sys_dba_monitoring_creds]
      [-dest_staging_area="<standby staging directory>"]
      [-configure_with_oracle_restart]
      [-src_ssh_tunnel_port="<ssh tunnel port used by standby database to
connect to primary database>"]
      [-dest_ssh_tunnel_port="<ssh tunnel port used by primary database to
connect to standby database>"]
      [-src_gateway_creds_name="<primary database host hybrid gateway agent
credential name>"]
      [-dest_gateway_creds_name="<standby database host hybrid gateway agent
credential name>"]
      [-dest_GI_host_creds_name="<standby database grid infrastructure
credential name>"]
      [-tde_wallet_creds_name="<transparent data encryption wallet credentials
of the primary database>"]
```

Options

- source_db_target_name

  Primary database Enterprise Manager target name source. Can either be a single-
  instance database or a cluster database instance.

- source_db_target_type

Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- dest_oracle_sid

  Standby database instance name.

- spname

  Standby database unique name.

- source_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- source_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle home. Default is to use preferred credential.

- dest_host_creds_name

  Standby database host named credential for an operating system user who can access the primary database Oracle home. Default is to use preferred credential.

- asm_inst_creds_name

  ASM instance named credential.

- dest_host_name

  Standby database host name. Default is primary host name.

- dest_oracle_home

  Standby database Oracle home location. Default is primary database Oracle home location.

- dest_target_name

  Standby database Enterprise Manager target name. Default is standby database unique name.

- use_duplicate

  Database files moved directly to clone the database host by the Recovery Manager(RMAN). Default is yes.

- source_staging_area

  Staging area used to store the backup of a primary database. This option is applicable only when use_duplicate is not set.

- storage_type

  Standby database storage type.

  - FILE_SYSTEM: Standby database files are in a regular file system.

  - ASM_STORAGE: Standby database uses Automatic Storage Management (ASM).

  Default is FILE_SYSTEM.

- dest_db_database_area

  Standby database files Oracle-managed files (OMF) location. Can be a regular file system (if storage_type is FILE_SYSTEM) or an ASM diskgroup (if storage_type is ASM_STORAGE).

- dest_db_recovery_area

  Standby database fast recovery area.

- dest_listener_selection

  Standby database listener selection.

  – GRID_INFRA: Uses the Grid Infrastructure Home listener.

  – DEST_DB_HOME: Uses the listener from the standby database Oracle home.

  Default is GRID_INFRA.

- dest_listener_name

  Standby database listener name. This option is applicable only if dest_listener_selection is set to DEST_DB_HOME.

  If not specified, default value is the first existing TCP listener found in the standby database Oracle home. Note that if dest_listener_name is specified, then dest_listener_port must also be specified.

- dest_listener_port

  Standby database listener port. This option is applicable only if dest_listener_selection is set to DEST_DB_HOME. Note that if dest_listener_port is specified, then dest_listener_name must also be specified.

- stby_type

  Type of the standby database to be created.

  – PHYSICAL

  – LOGICAL

  Default is PHYSICAL.

- use_broker

  Uses Data Guard broker to manage the Data Guard configuration. Default is yes.

- use_sys_dba_monitoring_creds

  Uses SYSDBA credentials to monitor the standby database. Default is no.

- dest_staging_area

  Staging area used to store the backup files transferred from the primary host. This option is applicable only when use_duplicate is not set.

- configure_with_oracle_restart

  If the destination host has Oracle Restart configured, it configures the standby database with Oracle Restart. When required, Oracle Restart automatically starts the standby database. Default is no.

- src_ssh_tunnel_port

  SSH Tunnel port used by the standby database to connect to the primary database. This is the port created on the standby host to forward the connection request to the primary database listener port.

- dest_ssh_tunnel_port

  SSH Tunnel port used by the primary database to connect to the standby database. This is the port created on the primary host to forward the connection request to the standby database listener port.

- src_gateway_creds_name

  Hybrid Gateway Agent named credential for the primary database host.

- dest_gateway_creds_name

  Hybrid Gateway Agent named credential for the standby database host.

- dest_GI_host_creds_name

  Grid Infrastructure named credentials for an operating system user who can access the grid infrastructure Oracle home.

- tde_wallet_creds_name

  Transparent Data Encryption wallet credentials for the primary database. Use create_named_credential verb to create these credentials as shown below:

```
emcli create_named_credential
    -cred_name=WC1 -cred_type=GenericPassword
    -auth_target_type='<system>'
    -attributes="GENERIC_PASSWORD:<Primary Database TDE Wallet Password>"
```

Examples

Example 1

The following command creates a standby database with the database unique name "database1" generated from the backup of a primary single-instance database named "Database".

```
emcli create_standby_database
    -source_db_target_name="database"
    -source_db_target_type="oracle_database"
    -dest_oracle_sid="database1"
    -spname="database1"
```

Example 2

The following command creates a standby database with the database unique name "database1" generated from the backup of a cluster database named "primary". The standby database uses SYSDBA credentials for monitoring and uses Data Guard broker to manage the Data Guard configuration.

```
emcli create_standby_database
    -source_db_target_name="primary"
    -source_db_target_type="rac_database"
    -dest_oracle_sid="database1"
    -spname="database1"
    -use_broker="Yes"
    -use_sys_dba_monitoring_creds="Yes"
```

Example 3

The following command creates a standby database on the Cloud host 'cloudhost.oracle.com' reachable through a destination gateway for a primary database that is encrypted with TDE. Communication between the primary and standby databases will be established using the specified tunnel ports.

```
emcli create_standby_database
        -source_db_target_name="primary_database"
        -source_db_target_type="oracle_database"
        -dest_oracle_sid='opcst'
        -spname='opcst'
```

ORACLE®

```
                        -dest_target_name='opcst'
                        -use_sys_dba_monitoring_creds='YES'
                        -source_db_creds_name='SRC_DB_CRED'
                        -use_duplicate='YES'
                        -asm_inst_creds_name="ASM"
                        -dest_db_database_area="DATADG"
                        -dest_db_recovery_area="RECODG"
                        -dest_listener_selection='DEST_DB_HOME'
                        -source_host_creds_name='SRC_HOST_CREDS'
                        -dest_host_creds_name='OPC_SSH_NAMED_CREDS'
                        -dest_host_name="cloudhost.oracle.com"
                        -dest_oracle_home="/scratch/aime/db/product/dbhome_1"
                        -dest_gateway_creds_name='DEST_GATEWAY_CREDS'
                        -dest_GI_host_creds_name="DEST_GRID_CREDS"
                        -src_ssh_tunnel_port="4001"
                        -dest_ssh_tunnel_port="4001"
                        -configure_with_oracle_restart"
                        -tde_wallet_creds_name='WC1'
```

Example 4

The following command creates a standby database on the Cloud host
'cloudhost.oracle.com' reachable through a destination gateway for a primary
database. Communication between the primary and standby databases will be
established using the specified tunnel ports.

```
emcli create_standby_database
 -source_db_target_name="primary_database"
 -source_db_target_type="oracle_database"
 -dest_oracle_sid="database1"
 -spname="database1"
 -source_db_creds_name="sys_creds"
 -source_host_creds_name="oracle_creds"
 -dest_host_creds_name="oracle_creds"
 -dest_oracle_home="/scratch/oracle_base/product/12.1.0/dbhome_1"
 -dest_host_name="cloudhost.oracle.com"
 -storage_type="ASM_STORAGE"
 -asm_inst_creds_name="ASM"
 -use_sys_dba_monitoring_creds="YES"
 -use_duplicate="YES"
 -dest_db_database_area="DATADG"
 -dest_db_recovery_area="RECODG"
 -dest_GI_host_creds_name="grid_creds"
 -src_ssh_tunnel_port="4001"
 -dest_ssh_tunnel_port="4001"
 -dest_gateway_creds_name="stby_gateway_creds"
 -src_gateway_creds_name="primary_gateway_creds"
 -configure_with_oracle_restart" -use_broker="YES"
```

create_swlib_directive_entity

Creates an entity of the Directive type in the Software Library. On successful creation,
the entity revision appears in the specified folder on the Software Library Home page.

Format

```
emcli create_swlib_directive_entity
        -name="entity_name"
        -folder_id="folder_id"
        -arg="[<arg prefix>;]<arg prop name>[;<arg suffix>]"
```

```
      [-shell_type]="<shell type>"
      [-run_privileged]
      [-file="<abs/relative file path>[;<new file name>]"
      [-upload_storage="<storage location name>;<storage type>"] | [-refer_
storage="<storage location name>;<storage type>"]
      [-credential_set_name="setname"] | [-credential_name="name" -
credential_owner="owner"]
      [-desc="entity_desc"]
      [-attr="<attr name>:<attr value>"]
      [-note="note text"]
      [-show_entity_rev_guid]
      [-show_cmd_line_and_exit]
```

[ ] indicates that the parameter is optional.

Parameters

- name

  Name of the entity.

- folder_id

  Folder ID where the entity is created. The Software Library Home page exposes the ID for folders and entities as a custom column (Internal ID). However, this is hidden by default.

- arg

  Command line argument property name, specified optionally with a prefix and/or a suffix. To specify multiple arguments, repeat the option.

- shell_type

  Shell type can be one of the following:

  – SUB_Exec - Specified in the script

  – SUB_PSUB_Bash - Basherl - Perl

  – Defaults to SUB_Perl.

- run_privileged

  This is an option to specify whether the directive should be executed with privileged credentials or not. This is executed with normal credentials by default.

- file

  If -upload_storage is specified, it is the absolute path of the file that is uploaded. If -refer_storage is specified, it is the relative path of the file that is referred from the storage location specified. File name stored in the Software Library after the file is upload is defaulted to the name of the file being uploaded/referred. A different file name can be specified, optionally, separated by ';'. The first file specified in the command line will be defaulted as the main file and will be executed when the directive is run.

- host

  Target name of the host where the files are available. This should be used in conjunction with the -upload_storage option.

- credential_set_name

  The set name of the preferred credential stored in the Management Repository for the host target. This can be one of the following:

- HostCredsNormal - default unprivileged credential set

- HostCredsPriv - privileged credential set

- credential_name

  Named credential stored in the Management Repository. This option must be specified along with the `-credential_owner` option. This must be used in conjunction with the `-upload_storage` option.

- credential_owner

  Owner of a named credential stored in the Management Repository. This option must be specified along with the -credential_name option. This must be used in conjunction with the -upload_storage option.

- upload_storage

  Destination storage location and type for the upload, separated by ';'. The location specified must be in the 'active' status. Defaulted to storage type and location of the first upload location configured for Software Library. The storage type can be one of the following:

  - OmsShared (OMS Shared File System)

  - OmsAgent (OMS Agent File System)

- refer_storage

  Storage location and type for referring to files, separated by ';'. The location specified must be in the 'active' status. The storage type can be one of the following:

  - http

  - NFS

  - ExtAgent

  If specified, this option takes precedence over `-upload_storage` option.

- desc

  Description of the entity. The new description is visible to all existing revisions.

- attr

  Attribute and its value is separated by ':'. To specify multiple attributes, repeat the option.

- note

  A note about the entity. Repeat the option for multiple notes.

- show_entity_rev_guid

  Option to enable printing of the internal GUID of the new entity revision or not.

- show_cmd_line_and_exit

  Option to enable printing of the command line and exiting without creating a new entity revision or not.

Examples

Example 1

The following example creates a directive entity named 'myAcmeInstall' in the specified folder. The directive has one argument with a '-home' prefix and the argument value is of the 'oh_home' property. It has two script files associated with it, the first one is defaulted as the main file. The main file is executed when the directive is executed. Also, the mymodule.pm file content is saved by the name 'common.pm'. The files are accessible locally by the emcli process owner. The folder ID value can be found on the Software Library Home page. The Software Library Home page exposes the ID for folders and entities as a custom column (Internal ID). However, this is hidden by default.

```
emcli create_swlib_directive_entity
      -name="myAcmeInstall"
      -folder_
id="oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19AA560C"
-arg="-home :oh_home"
      -shell_type=SUB_Perl
      -file=/u01/scripts/myscript.pl
      -file=/u01/scripts/mymodule.pm;common.pm
```

Example 2

The following example creates a directive entity named 'myAcmeInstall' in the specified folder. The directive has one argument with a '-home' prefix and the argument value is of the 'oh_home' property. It has two script files associated with it, the first one is defaulted as the main file. The main file is executed when the directive is executed. Also, the mymodule.pm file content is saved by the name 'common.pm'. The files are retrieved from the host 'fs1.us.acme.com' using the credential identified as 'MyAcmeCreds' owned by 'ACME_USER'. The folder ID value can be found on the Software Library Home page. The Software Library Home page exposes the ID for folders and entities as a custom column (Internal ID). However, this is hidden by default.

```
emcli create_swlib_directive_entity
      -name="myAcmeInstall"
      -
folder_id="oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19AA560C
"id="oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19AA560C"
      -arg="-home :oh_home"
      -shell_type=SUB_Perl
      -file=/u01/scripts/myscript.pl
      -file=/u01/scripts/mymodule.pm;common.pm
      -host="fs1.us.acme.com"
      -credential_name="MyAcmeCreds"
      -credential_owner="ACME_USER
```

create_swlib_entity

Creates an entity in the software library. Upon successful creation, the entity revision appears under the specified folder on the software library home page.

Format

```
emcli create_swlib_entity
      -name="entity_name"
      -folder_id="folder_id"
      [-type]="type_internal_id"]
      [-subtype]="subtype_internal_id"]
      [-desc="entity_desc"]
      [-attr="<attr_name>:<attr value>"]
      [-prop="<prop_name>:<prop value>"]
      [-secret_prop="<secret_prop_name>:<secret_prop=_value>"]
```

```
     [-note="note_text"]
```

```
[ ]  indicates that the parameter is optionalis optional
```

Options

- **name**

  Name of the entity.

- **folder_id**

  Identifier of the folder where the entity is to be created. The software library home page exposes the identifier for folders and entities as a custom column (Internal ID), and is hidden by default.

- **type**

  Use the list_swlib_entity_types verb to identify the type.

- **subtype**

  Internal identifier of the entity subtype, which defaults to the 'Generic Component' subtype for the 'Component' type. Use the list_swlib_entity_types verb to identify the subtype.

- **desc**

  Description of the entity.

- **attr**

  An attribute and its value, separated by a colon (:). To specify values for multiple attributes, repeat this option.

- **prop**

  A configuration property and its value, separated by a colon (:). To specify values for multiple properties, repeat this option.

- **secret_prop**

  A configuration property and its secret value separated by a colon (:). It is recommended to not specify the secret value on the command line. If omitted from the command line, the value is prompted for. To specify values for multiple properties, repeat this option.

- **note**

  A note on the entity. For multiple notes, repeat this option.

Examples

Example 1

This example creates an entity named 'myAcmeInstall' under the specified folder. The entity is of type 'Component' and subtype 'Generic Component, by default. The folder identifier value can be found on the software library home page. The software library home page exposes the identifier for folders and entities as a custom column (Internal ID), and is hidden by default.

```
emcli create_swlib_entity
      -name="myAcmeInstall"
      -folder_id=
"oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19AA560C"
```

Example 2

This example creates an entity named 'myAcmeInstall' under the specified folder with the specified description. The entity is of type 'Component' and subtype 'Generic Component' by default. Values for the entity attributes, viz. PRODUCT, PRODUCT_VERSION and VENDOR, are specified. The value for the configuration property named DEFAULT_HOME is specified. A note on the entity is also specified. The identifier of the newly created entity revision is printed on the standard output.

```
emcli create_swlib_entity
      -name="myAcmeInstall"
      -folder_id=
"oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19AA560C"
       -desc="myAcmeInstall description"
       -attr="PRODUCT:Acme"
       -attr="PRODUCT_VERSION:3.0"
       -attr="VENDOR:Acme Corp"
       -prop="DEFAULT_HOME:/u01/acme3/"
       -note="myAcmeInstall for test servers"
```

create_swlib_folder

Creates a folder in the software library.

Format

```
emcli create_swlib_folder
      -name="folder_name"
      -parent_id="parent_folder_id"
      [-desc="folder_description"]
```

```
[ ]  indicates that the parameter is optionalis optional
```

Options

*   **name**

    Name of the folder.

*   **parent_id**

    Identifier of the parent folder under which the folder is to be created. To create a folder under the root folder, specify the parent folder identifier as 'ROOT.' The software library home page exposes the identifier for folders and entities as a custom column (Internal ID) and is hidden by default.

*   **desc**

    Description of the folder.

Example

This example creates a folder named 'myFolder' under the specified parent folder.

```
emcli create_swlib_folder
      -name="myFolder"
      -parent_id=
"oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19AA560C"
      -desc="myFolder description"
```

create_system

Defines a system: name and its members. After the system is created, you can edit the system from the Enterprise Manager Cloud Control console to configure charts to be displayed for system members.

A database system contains a primary database and related targets such as Listener and Automatic Storage Management. It also includes standby databases and their related targets if the database is in a Data Guard configuration. Database systems cannot be created for standby databases.

Format

```
emcli create_system
      -name="name"
      [-type=<system>]
      [-add_members="name1:type1:key_member/non_key_member;name2:type2;..."]...
      [-separator=add_members="sep_value"]
      [-subseparator=add_members="subsep_value"]
      -timezone_region="actual_timezone_region"
      [-owner="owner"]
      [-meta_ver="meta_version_of_system_type"]
      [-is_propagating="true|false"]
      [-availability_type="ALL|ANY"]
```

```
[ ]  indicates that the parameter is optionalis optional
```

Options

- **name**

  Name of the system.

- **type**

  System type: generic_system. Defaults to "generic_system".

- **add_members**

  Add existing targets to the system. Each target is specified as a name-value pair `target_name:target_type`. You can specify this option more than once. key_member specifies that this target is a part of the systems availability calculation.

- **separator**

  Name-value pair separator for the given argument.

- **subseparator**

  Separates the name from the value for the given argument.

- **timezone_region**

  Actual time zone region.

- **owner**

  Owner of the system.

- **meta_ver**

  Meta version of the system type. Defaults to "1.0".

- **is_propagating**

  Flag to indicate if the privilege on the system will be propagated to member targets or not. The default value is false.

- **availability_type**

  Availability calculation method of the system. Defining this is required if key_member is defined. ALL denotes that all key members must be up in order to mark the system as up. ANY denotes that at least one of the key members must be up in order to mark the system as up.

Output

Success / Error. If you attempt to create a standby database, you will receive the following message:

Operation not supported for given system type.

Examples

Example 1

This example creates a generic system named `db_system` and supports backward compatibility. This system consists of two Oracle databases: `emp_rec` and `payroll`. The owner of this system is `user1`. The meta version of the system type is `3.0`.

```
emcli create_system -name=db_system
    -add_members="emp_rec:oracle_database"
    -add_members="payroll:oracle_database"
    -timezone_region="PST8PDT"
    -owner="user1"
```

Example 2

This example creates a generic system named db_system1. This system consists of two Oracle databases: emp_rec and payroll. emp_rec is a key member for the system. The availability calculation method is if ANY of the key members is up, the system is up. The meta version of the system type is 3.0. This example shows the recommended method for creating a system.

```
emcli create_system -name=db_system1
    -add_members="emp_rec$oracle_database$key_member"
    -add_members="payroll$oracle_database"
    -subseparator=add_members="$"
    -timezone_region="PST8PDT"
    -availability_type="ANY"
```

create_tenant

Creates a new tenant in Enterprise Manager.

Format

Standard Mode

```
emcli create_tenant
    -name="name"
    -description="description"
    -owner_name="owner_name"
    [-owner_password="owner_password"]
```

```
           [-owner_type="owner_type"]
           [-namespace="namespace"]
```

```
[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
emcli create_tenant(
      name="name"
      ,description="description"
      ,owner_name="owner_name"
      [,owner_password="owner_password"]
      [,owner_type="owner_type"]
      [,namespace="namespace"]
)
```

```
[ ]  indicates that the parameter is optional
```

Options

*   name

    The name of the tenant.

*   description

    Description of the tenant.

*   owner_name

    The tenant owner name.

*   owner_type

    The type of owner. `EM_USER` is the default owner type. Other possible values are `EM_USER` and `EXTERNAL_USER`.

*   namespace

    The namespace for the tenant.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example creates a tenant by name `my_tenant` with `john_doe` as the tenant owner.

```
emcli create_tenant
      -name=my_tenant
      -description="This is the tenant description."
      -owner_name=john.doe
      -owner_password=pw
      -owner_type=EM_USER
```

create_udmmig_session

Creates a session to migrate user-defined metrics (UDMs) to metric extensions for targets.

Format

```
emcli create_udmmig_session
      -name=<session_name>
      -desc=<session_description>
      [-udm_choice=<specific_udm_to_convert>]*
      {-target=<type:name_of_target_to_migrate> }*
      | {-input_file=targetList:<complete_path_to_file>};
          {-template=<template_name_to_update> }*
      | {-input_file=templateList:<complete_path_to_file>}
      [-allUdms]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the migration session to be created.

- **desc**

  Description of the migration session to be created.

- **udm_choice**

  Specify if the session should migrate specific UDMs. Otherwise, all UDMs are migrated.

- **target**

  The type:name of the target to be updated. You can specify multiple values.

- **input_file=targetList**

  Specify a file name that contains a list of targets, one per line, in the following format:

  ```
  <targetType>:<targetName>
  ```

  For more information about the input_file parameter see -input_file Syntax Guidelines.

- **template**

  Name of the monitoring template to update. You can specify multiple values.

- **input_file=templateList**

  Specify a file name that contains a list of templates, one name per line.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **allUdms**

  Forces the session to contain all UDMs from targets and templates. (The default behavior just selects those not in a session.)

Examples

Example 1

This example creates a new session named hostsession that migrates the UDM hostudm on the target testhost.

```
emcli create_udmmig_session
      -name=hostsession -desc="Convert UDMs for Host Target"
      -udm_choice=hostudm -target=host:testhost
```

Example 2

This example creates a new session named hostsession that migrates all the unconverted UDMs on the target testhost that are not in a session.

```
emcli create_udmmig_session
      -name=hostsession -desc="Convert UDMs for Host Target"
      -target=host:testhost -allUdms
```

create_user

Creates a new Enterprise Manager administrator.

Standard Mode

```
emcli create_user
      -name="name"[-password="password"][-type="type of user"]
      [-roles="role1;role2;..."]
      {-email="email1;email2;..."]
      [-privilege="name[;secure-resource-details]]"]
      [-profile="profile_name"]
      [-desc="user_description"]
      [-expired="true/false"]
      [-prevent_change_password="true/false"]
      [-department="department_name"]
      [-cost_center="cost_center"]
      [-line_of_business="line_of_business"]
      [-contact="contact"]
      [-location="location"]
      [-external_user_id="external_user_id"]
      [-tenant="tenant"]
      [-like="like"]
      [-input_file="FILE:file_path"]
      [-separator="separator:attribute_name:character"]
      [-subseparator="subseparator:attribute_name:character"]

[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
create_user(
       name="name"
       [,password="password"]
       [,type="type of user"]
       [,roles="role1;role2;..."]
       [,email="email1;email2;..."]
       [,privilege="name[;secure-resource-details]]"]
       [,profile="profile_name"]
       [,desc="user_description"]
       [,expired="true/false"]
       [,prevent_change_password="true/false"]
       [,department="department_name"]
       [,cost_center="cost_center"]
       [,line_of_business="line_of_business"]
       [,contact="contact"]
       [,location="location"]
       [,external_user_id="external_user_id"]
       [,tenant="tenant"]
       [,like="like"]
       [,input_file="FILE:file_path"]
       [,separator="separator:attribute_name:character"]
```

```
        [,subseparator="subseparator:attribute_name:character"]        )
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Administrator name.

- **password**

  Administrator password.

- **type**

  Type of user. The default value of this option is EM_USER. Possible values for this option are:

  – EM_USER

  – EXTERNAL_USER

  – DB_EXTERNAL_USER

- **roles**

  List of roles to grant to this administrator. Currently, the built-in roles include PUBLIC.

- **email**

  List of e-mail addresses for this administrator.

- **privilege**

  Privilege to grant to this administrator. You can specify this option more than once. Specify <secure_resource_details> as:

  ```
  resource_guid|
  [resource_column_name1=resource_column_value1[:resource_column_name2=resource_colum
  n_value2]..]"
  ```

  To retrieve the list of SYSTEM privileges, which do not require resource information, execute the following emcli command:

  ```
  emcli get_supported_privileges -type=SYSTEM
  ```

  To retrieve the complete list of privileges and resource column names, execute the following command:

  ```
  emcli get_supported_privileges
  ```

  To retrieve the list of TARGET privileges, execute the following emcli command:

  ```
  emcli get_supported_privileges -type=TARGET
  ```

  To retrieve the list of job privileges, execute the following emcli command:

  ```
  emcli get_supported_privileges -type=JOB
  ```

- **profile**

  Database profile name. It uses DEFAULT as the default profile name.

- **desc**

  User description for the user being added.

- **expired**

  Use this option to expire the password immediately. The default is false.

- **prevent_change_password**

  Valid values are true or false. When set to true, you cannot change your own password. The default is false.

- **department**

  Name of the department of the administrator.

- **cost_center**

  Cost center of the administrator in the organization.

- **line_of_business**

  Line of business of the administrator.

- **contact**

  Contact information of the administrator.

- **location**

  Location of the administrator.

- **external_user_id**

  External user ID of the administrator..

- **tenant**

  Tenant name of the administrator.

- like

  Create like another user.

- **input_file**

  Reads the contents of a file and passes as property value.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **separator**

  By default, multi-value input attributes use a semicolon (;) as a separator. Specifying this option overrides the default separator value.

  Example: `separator="<`*`attribute_name=sep_char`*`>"` where *`attribute_name`* is name of the attribute for which you want to override the separator character, and *`sep_char`* is the new separator character.Example: `separator="att=#"`

- **subseparator**

  By default, multi-value input attributes use a colon (:) as a subseparator. Specifying this option overrides the default subseparator value.

  Example: `subseparator="<`*`attribute_name=sep_char`*`>"` where *`attribute_name`* is name of the attribute for which you want to override the separator character, and *`sep_char`* is the new subseparator character.Example: `separator="att=#"`

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

These examples create an Enterprise Manager administrator named `new_admin`. This administrator has two privileges: the ability to view the job with ID 923470234ABCDFE23018494753091111 and the ability to view the target `host1.example.com:host`. The administrator `new_admin` is granted the `PUBLIC` role.

Example 1 Command-Line

```
emcli create_user
      -name="new_admin"
      -password="oracle"
      -email="first.last@example.com;joe.shmoe@shmoeshop.com"
      -roles="public"
      -privilege="view_job;923470234ABCDFE23018494753091111"
      -privilege="view_target;host1.example.com:host"
```

Example 2 - Scripting and Interactive

```
create_user
      (name="new_admin"
      ,password="oracle"
      ,email="first.last@example.com;joe.shmoe@shmoeshop.com"
      ,roles="public"
      ,privilege="view_job;923470234ABCDFE23018494753091111"
      ,privilege="view_target;host1.example.com:host")
```

These examples make User1 an Enterprise Manager user, which is already created on an external user store like the SSO server. The contents of priv_file are view_target;host1.example.com:host . User1 will have view privileges on the host1.example.com:host target.

Example 3 - Command-Line

```
emcli create_user
      -name="User1"
      -type="EXTERNAL_USER"
      -input_file="privilege:/home/user1/priv_file"
```

Example 4 - Scripting and Interactive

```
create_user
      (name="User1"
      ,type="EXTERNAL_USER"
      ,input_file="privilege:/home/user1/priv_file")
```

create_user_profile

Creates a user profile.

Format

Standard Mode

```
emcli create_user_profile
      -name="profile name"
      [-description="profile desc"]
      [-users="users to be associated"]
```

```
[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
create_user_profile(
      name="profile name"
      [,description="profile desc"]
      [,users="users to be associated"]
      )
```

```
[ ]  indicates that the parameter is optional
```

Options

*   name

    The name of the user profile to be created.

*   description

    Description of the user profile to be created.

*   users

    The names of the users with whom the user profile is to be associated.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example creates a new user profile by name `profile1`.

```
emcli create_user_profile
      -name="profile1"
      -description=test profile
      -users=user1;user2
```

data_transfer

Transfers data from source to target.

Format

```
emcli data_transfer        -inputFile="File containing properties required for
transferring data"
```

Options

*   inputFile

    Location of file containing properties required for transferring data. The mandatory
    properties required for this job are:

    SOURCE_LOCATION = Location of the data at the source host.

    SRC_HOST = Source host containing the data.

    SRC_HOST_CREDS = Credentials for the host on which the data is located. If the
    source host is on OPC, this should be Host SSH Credentials.

    DEST_HOST = Destination host where the data should be copied to.

DEST_HOST_CREDS = Credentials for the host where the data will be copied to. If the destination host is on OPC, this should be Host SSH Credentials.

DEST_LOCATION = Location on the destination host where the data should be copied.

Example

The following example transfers data from the source to the target contained in the `data_transfer.props` file:

```
emcli data_transfer
      -input_file=data:/u01/files/data_transfer.props
```

db_clone_management

Verbs for database clone life cycle management. These are organized in three main sections:

- Creation of Clones
- Test Master Management
- Clone Lifecycle Management

Creation of Clones

EM CLI verbs used to create different types of clones.

- createFullClonePDB
- createFullClone
- createCloneDB
- createSnapshotClone

createFullClonePDB

Clone a non-CDB as a PDB.

```
emcli db_clone_management
-createFullClonePDB
-target_name="source database target name"
-target_type="<oracle_database | rac_database>"
-input_file=<input file path> -print_properties
```

Examples:

- Print the input properties required to clone a non-container database PROD to a pluggable database:

```
emcli db_clone_management
-createFullClonePDB
-target_name="PROD"
-target_type="oracle_database"
-print_properties
```

- Clone the non-container database PROD as a pluggable database:

```
emcli db_clone_management
-createFullClonePDB
-target_name="PROD"
-target_type="oracle_database"
-input_file=data:/u01/inputs/clone_noncdb_pdb.prop
```

createFullClone

Create a Full Clone Database.

```
emcli db_clone_management
-createFullClone
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
-clone_type="<LIVE | POINT_IN_TIME>"
-input_file=<input file path>
-print_properties
```

Examples:

- Print the input properties required to create a live clone from the database *PROD*:

```
emcli db_clone_management
-createFullClone
-target_name="PROD"
-target_type="oracle_database"
-clone_type="LIVE"
-print_properties
```

- Create a live clone from the database *PROD*:

```
emcli db_clone_management
-createFullClone
-target_name="PROD"
-target_type="oracle_database"
-clone_type="LIVE"
-input_file=data:/u01/inputs/full_live.prop
```

- Print the input properties required to create a prior point in time clone from the RAC database *MYRAC*:

```
emcli db_clone_management
-createFullClone -target_name="MYRAC"
-target_type="rac_database"
-clone_type="POINT_IN_TIME"
-print_properties
```

- Create a prior point in time clone from the RAC database *MYRAC*:

```
emcli db_clone_management
-createFullClone -target_name="MYRAC"
```

```
-target_type="rac_database"
-clone_type="POINT_IN_TIME"
-input_file=data:/u01/inputs/full_prior.prop
```

createCloneDB

Create a CloneDB Database.

```
emcli db_clone_management
-createCloneDB
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
-input_file=<input file path>
-print_properties
```

Examples:

• Print the input properties required to create a CloneDB from database *MYDB*:

```
emcli db_clone_management
-createCloneDB
-target_name="MYDB"
-target_type="oracle_database"
-print_properties
```

• Create CloneDB from the database"*MYDB*:

```
emcli db_clone_management
-createCloneDB
-target_name="MYDB"
-target_type="oracle_database"
-input_file=data:/u01/inputs/thin.prop
```

createSnapshotClone

Create an Exadata Sparse Clone Database.

```
emcli db_clone_management
-createSnapshotClone
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
-input_file=<input file path>
-print_properties
```

Examples:

• Print the input properties required to create an Exadata Sparse Clone from the Test
  Master *MYTEST*:

```
emcli db_clone_management
-createSnapshotClone
-target_name="MYTEST"
```

```
-target_type="oracle_database"
-print_properties
```

- Create an Exadata Sparse Clone from the Test Master *MYTEST*:

```
emcli db_clone_management
-createSnapshotClone
-target_name="MYTEST"
-target_type="oracle_database"
-input_file=data:/u01/inputs/sparse_input.prop
```

Test Master Management

EM CLI verbs for different Test Master database operations.

- createTestMaster
- enableTestMaster
- disableTestMaster
- listTestMasters

createTestMaster

Create a Test Master Database.

```
emcli db_clone_management
-createTestMaster
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
-clone_type="<LIVE | POINT_IN_TIME>"
-input_file=<input file path>
-print_properties
```

Examples:

- Print the input properties required for creating a Test Master from the RAC
  database *MYPROD*:

```
emcli db_clone_management
-createTestMaster
-target_name="MYPROD"
-target_type="rac_database"
-clone_type="LIVE"
-print_properties
```

- Create a Test Master from the RAC database *MYPROD*:

```
 emcli db_clone_management
-createTestMaster
-target_name="MYPROD"
-target_type="rac_database"
-clone_type="LIVE"
-input_file=data:/u01/inputs/tm_live.prop
```

- Print the input properties required for creating a prior point in time Test Master from the database *MYDB*:

```
emcli db_clone_management
-createTestMaster
-target_name="MYDB"
-target_type="oracle_database"
-clone_type="POINT_IN_TIME"
-print_properties
```

- Create a prior point in time Test Master from the database *MYDB*:

```
emcli db_clone_management
-createTestMaster
-target_name="MYDB"
-target_type="oracle_database"
-clone_type="POINT_IN_TIME"
-input_file=data:/u01/inputs/tm_prior.prop
```

enableTestMaster

Enable a Database as Test Master.

```
emcli db_clone_management
-enableTestMaster
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
-input_file=<input file path>
-print_properties
```

> **Note:**
>
> - `-enableTestMaster` can only be used for clones created using the Database Clone Management flow.
> - `-input_file` is a mandatory parameter if the Database is on Exadata.

Examples:

- Enable the database *MYDB* as a Test Master:

```
emcli db_clone_management
-enableTestMaster
-target_name="MYDB"
-target_type="oracle_database"
```

- Print the input properties required to enable *EXACLONE* as Test Master:

```
emcli db_clone_management
-enableTestMaster
-target_name="EXACLONE"
```

```
-target_type="oracle_database"
-print_properties
```

- Enable the database *EXACLONE* as Test Master:

```
emcli db_clone_management
-enableTestMaster
-target_name="EXACLONE"
-target_type="oracle_database"
-input_file=data:/u01/inputs/ena_tm.prop
```

disableTestMaster

Disable a Database as Test Master.

```
emcli db_clone_management
-disableTestMaster
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
-input_file=<input file path>
-print_properties
```

> **✎ Note:**
>
> - `-disableTestMaster` can only be used for clones created using the Database Clone Management flow.
> - `-input_file` is a mandatory parameter if the Database is on Exadata.

Examples:

- Disable the database *MYDB* as Test Master:

```
emcli db_clone_management
-disableTestMaster
-target_name="MYDB"
-target_type="oracle_database"
```

- Print the input properties required to disable *EXACLONE* as Test Master:

```
emcli db_clone_management
-disableTestMaster
-target_name="EXACLONE"
-target_type="oracle_database"
-print_properties
```

- Disable the database *EXACLONE* as Test Master:

```
emcli db_clone_management
-disableTestMaster
-target_name="EXACLONE"
```

```
-target_type="oracle_database"
-input_file=data:/u01/inputs/dis_tm.prop
```

listTestMasters

List Test Master Databases.

```
emcli db_clone_management -listTestMasters\
target_name="database target name"
-target_type="<oracle_database | rac_database>"
```

Examples:

- List all Test Master databases available in Enterprise Manager:

  ```
  emcli db_clone_management -listTestMasters
  ```

- List all Test Master databases created from the database instance *MYPROD*:

  ```
  emcli db_clone_management
  -listTestMasters
  -target_name="MYPROD"
  -target_type="oracle_database"
  ```

Clone Lifecycle Management

EM CLI verbs for Lifecycle Management operations for database clones.

refreshDatabase

Refresh a clone database.

```
emcli db_clone_management
-refreshDatabase
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
-clone_type="<LIVE | POINT_IN_TIME>"
-input_file=<input file path>
-print_properties
```

> **✎ Note:**
>
> - `refreshDatabase` can only be used for clones created using the Database Clone Management flow.
> - `refreshDatabase` is a mandatory parameter if the Database is on Exadata.

Examples:

- Print the input properties required to refresh a Test Master database *MYTEST* with latest data:

```
emcli db_clone_management
-refreshDatabase
-target_name="MYTEST"
-target_type="oracle_database"
-clone_type="LIVE"
-print_properties
```

- Refresh the Test Master *MYTEST* with latest data:

```
emcli db_clone_management
-refreshDatabase
-target_name="MYTEST"
-target_type="oracle_database"
-clone_type="LIVE"
-input_file=data:/u01/inputs/tm_refresh.prop
```

- Print the input properties required to perform a point in time refresh on a Snap Clone *SNAP*:

```
 emcli db_clone_management
-refreshDatabase
-target_name="SNAP"
-target_type="oracle_database"
-clone_type="POINT_IN_TIME"
-print_properties
```

- Perform a point in time refresh on the Snap Clone *SNAP*:

```
emcli db_clone_management
-refreshDatabase
-target_name="SNAP"
-target_type="oracle_database"
-clone_type="POINT_IN_TIME"
-input_file=data:/u01/inputs/snap_refresh.prop
```

listClones

List of database clones.

```
emcli db_clone_management
-listClones
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
```

Example:

- List all clones created from the database *MYDB*:

```
emcli db_clone_management
-listClones
```

```
-target_name="MYDB"
-target_type="oracle_database"
```

listDatabaseBackups

List of database backups.

```
emcli db_clone_management
-listDatabaseBackups
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
```

Example:

• List available RMAN backups and image profiles of the database *MYDB*:

```
emcli db_clone_management
-listDatabaseBackups
-target_name="MYDB"
-target_type="oracle_database"
```

listDatabaseSnapshot

List of database snapshots.

```
emcli db_clone_management
-listDatabaseSnapshots
-target_name="database target name"
-target_type="<oracle_database | rac_database>"
```

Example:

• List available snapshot profiles created from database *MYSRC*:

```
emcli db_clone_management
-listDatabaseSnapshots
-target_name="MYSRC"
-target_type="oracle_database"
```

deleteDatabase

Enter a short description of your topic here (optional).

```
emcli db_clone_management
-deleteDatabase-target_name="database target name"
-target_type="<oracle_database | rac_database | oracle_pdb>"
-input_file=<input file path>
-print_properties
```

Examples:

- Print the input properties required to delete database *MYCLONE*:

```
emcli db_clone_management
-deleteDatabase
-target_name="MYCLONE"
-target_type="oracle_database"
-print_properties
```

- Delete database *MYCLONE*:

```
emcli db_clone_management
-deleteDatabase -target_name="MYCLONE"
-target_type="oracle_database"
-input_file=data:/u01/inputs/deldb.prop
```

- Print the input properties required to delete pluggable database *MY_PDB_CLONE*:

```
emcli db_clone_management
-deleteDatabase
-target_name="MYCLONE"
-target_type="oracle_pdb"
-print_properties
```

- Delete pluggable database *MYCLONE*:

```
emcli db_clone_management
-deleteDatabase
-target_name="MY_PDB_CLONE"
-target_type="oracle_pdb"
-input_file=data:/u01/inputs/delpdb.prop
```

db_cloud_maintenance

Performs database Cloud maintenance tasks.

db_cloud_maintenance -activateSoftware

Activates the new software of the pool.

Format

```
emcli db_cloud_maintenance -activateSoftware
     -pool_name= "pool_name"
     -pool_type= "pool_type" l
     [-force= "force" ]

[ ]  indicates that the parameter is optional.
```

Options

- pool_name

  The name of the pool.

- pool_type

The type of the pool.

- force

  Forcibly activates new members.

Example

The following example forcibly activates new members and activates new software for the Oracle Cloud Zone pool with the name POOL.

```
emcli db_cloud_maintenance -activateSoftware
     -pool_name=POOL
     -pool_type=oracle_cloud_zone
     -force=true
```

db_cloud_maintenance -performOperation

Performs a named operation on a specified pool.

Format

```
emcli db_cloud_maintenance -performOperation
       -name= "name"
       -description= "description"
       -purpose= "purpose"
       -pool_name= "pool_name"
       -pool_type= "pool_type"
        [-start_schedule= "start_schedule"]
        [-end_schedule= "end_schedule" ]
        [-input_file= "data:input_file" ]
        [-target_type= "target_type" ]
        [-target_list= "target_list" ]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- name

  The name of the operation.

- description

  The description of the operation.

- purpose

  The purpose of the operation. The possible values are: UPDATE_DB, UPDATE_RACDB, UPDATE_GI, DEPLOY_DB_SOFTWARE, DEPLOY_GI_SOFTWARE, DEPLOY_RAC_SOFTWARE, ROLLBACK_DB, ROLLBACK_GI, ROLLBACK_RACDB, and CLEANUP_SOFTWARE.

- pool_name

  The name of the pool.

- pool_type

  The type of the pool.

- start_schedule

  The scheduled start time. The format for start_schedule is yyyy-MM-dd HH:mm:ss, for example start_schedule="2013-11-11 12:15:30". The default start time is immediate.

- end_schedule

  The scheduled end time. The format for end_schedule is yyyy-MM-dd HH:mm:ss, for example end_schedule="2014-11-11 22:30:00". The default end time is indefinite.

- input_file

  Input data for the maintenance action, for example input_file="data:~/input_files/data.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- target_type

  The default target type is identified based on the purpose. For example, if the purpose is DEPLOY_DB, then the default target type becomes oracle_home.

- target_list

  A comma separated list of targets. The target list is the list of entities based on the target type that is selected. For example, if target_type=rac_database target_list,then the target list is "rac1.example.com,rac2.example.com". The default target_list is based on the purpose. For example if the purpose is DEPLOY_DB, the default target list becomes the list of Oracle homes present in the pool.

Example

The following example performs the Update RAC Database operation for the Oracle Cloud Zone pool with the name POOL.

```
emcli -performOperation
    -name="Update RAC Database "
    -description="Update RAC database Instance"
    -purpose="UPDATE_RACDB"
    -start_schedule="start_time:2014/09/01 00:00"
    -end_schedule="start_time:2014/09/01 13:00"
    -pool_name="POOL NAME"
    -pool_type=oracle_cloud_zone
    -target_type=rac_database
    -target_list="rac1.example.com"
```

db_cloud_maintenance -relocateService

Relocates DB/PDB instance from one pool to another. You must use this verb with 'discoverOnly' option, to Onboard an EM managed DB/PDB on DBaaS Cloud.

Format

```
emcli db_cloud_maintenance –relocateService
    -input_file="data:<absolute_path_to_input_file>"
```

For more information on maintaining a database pool and customizing databases, see *Maintaining and Customizing Databases* in the *Enterprise Manager Cloud Administration Guide.*

Input File Properties

```
SRC_PDB_TARGET_NAME=<EM Target name of the PDB to be relocated>
DEST_CDB_TARGET_NAME=<EM Target name of the destination Target
Container DB, optional, required when SELECT_TARGET_OPTION=RUNTIME is
```

```
not used>
DEST_CDB_TARGET_TYPE=<oracle_database|rac_database>
STORAGE_LOCATION=<Optional>, Storage location for destination PDB
DEST_WORK_DIR=<Optional>, absolute path to temporary work directory at
destination
SELECT_TARGET_OPTION=RUNTIME<Optional, needed if destination CDB needs to be
selected automatically
DEST_CDB_POOL_NAME=<Optional>, Destination Pluggable Database Pool name
needed only when SELECT_TARGET_OPTION parameter is specified
USE_SAME_STORAGE_LOCATION=Y<Optional>, needed in case PDB relocation is to
be performed without moving PDB datafiles, target PDB datafile(s) storage
location must also be accessible to destination Container DB
WORKLOADS=<Optional>, to be specified if during relocate, the instance needs
to be associated with the different workload size. The given  workload  must
exist in the Service template.
PDBAAS_CUSTOM_PRE_SCRIPTS_URN=<Optional>, if not specified, pre script
configured in the Service Template associated with this instance will be
selected
PDBAAS_CUSTOM_POST_SCRIPTS_URN=<Optional>, if not specified, post script
configured in the Service Template associated with this instance will be
selected
```

emcli db_cloud_maintenance –resizeService

Helps in resizing an OEM DBaaS Pluggable Database instance.

Format

```
emcli db_cloud_maintenance -resizeService -pdbName="<pdb target name>"
    -workloadName="<workload name to be used for resizing>"
```

Options

- -pdbName

  Enter the pdb target name

- -workloadName

  Enter the workload name to be used for resizing

db_software_maintenance

Image and Subscriptions

createImage

Description

Creates a new image.

Format

```
emcli db_software_maintenance -createImage
  -image_name= "image_name"
  -description= "description"
  -type= "type"
```

```
-target_type= "target_type"
-version= "version"
-platform_id= "platform_id"
-status= "status"
```

Options

- image_name

  The name of the image.

- description

  The description of the image.

- type

  The type of the image, for example SWLIB if the image version will be in the Software Library.

- target_type

  The target type of the image. For example, if the image is being created to manage single instance Oracle Database then the target type is 'oracle_database.

- version

  The RDBMS version of the product, for example 11.2.0.4.0.

- platform_id

  The platform id, for example 226 for Linux x86_64.

- status

  The image status, for example PRODUCTION.

Examples

The following example creates a new image with the name GI_11204.

```
emcli db_software_maintenance -createImage
  -image_name="GI_11204"
  -description="GI_11204"
  -type="SWLIB"
  -target_type=cluster -version=11.2.0.4.0
  -platform_id=226 -status=PRODUCTION
```

getImages

Description

Returns the list of images which are created in the cloud flow. The gold image that is created as a part of profile creation will not be listed here. Images which failed or are inactive won't appear on the list.

Format

```
emcli db_software_maintenance -getImages
[-all]
```

Examples

The following example returns a list of images present in production.

```
emcli db_software_maintenance -getImages
```

To get a list of all created images, use the -all parameter:

```
emcli db_software_maintenance -getImages -all
```

deleteImage

Description

Deletes an image.

Format

```
emcli db_software_maintenance -deleteImage
  -image_id= "image_id"
  [-force= "force"]
```

[ ] indicates that the parameter is optional.

- image_id

  The ID of the image to be deleted.

- force

  Deletes forcibly even if the image has subscribed targets.

Examples

The following example deletes the image with the ID
01B5F14FD57D7B89E05313B2F00A739F.

```
emcli db_software_maintenance -deleteImage
  -image_id="01B5F14FD57D7B89E05313B2F00A739F
```

subscribeTarget

Description

Creates new target or modifies the target subscription.

Format

```
emcli db_software_maintenance -subscribeTarget
  -target_name= "target_name"
  -target_type= "target_type"
  -[parent_target_name= "parent_target_name"]
  -[parent_target_type= "parent_target_type"]
  -image_id= "image_id"
  -[version_id= "version_id'
```

[ ] indicates that the parameter is optional.

Options

- target_name

  The name of the target.

- target_type

  The target type.

- parent_target_name

  The parent target name.

- parent_target_type

  The parent target type.

- image_id

  The image id.

- version_id

  The version id.

Examples

The following example modifies the Oracle Cloud Zone target `POOL NAME`.

```
emcli db_software_maintenance -subscribeTarget
  -target_name="POOL NAME"
  -target_type=oracle_cloud_zone
  -image_id=FE55AD7AB28974EFE04313B2F00AD4A0
```

getTargetSubscriptions

Description

Returns a list of subscriptions for the specified target.

Format

```
emcli db_software_maintenance -getTargetSubscriptions
  -target_name= "target_name"
  -target_type= "target_type"
  [-image_type= "image_type"]
```

```
[ ]  indicates that the parameter is optional.
```

- target_name

  The name of the target.

- target_type

  The target type.

- image_type

  The image type.

Examples

The following example returns a list of subscriptions for the Oracle Cloud Zone target with the name `POOL NAME`.

```
emcli db_software_maintenance -getTargetSubscriptions
  -target_name="POOL NAME"
  -target_type="oracle_cloud_zone"
```

unsubscribeTarget

Description

Unsubscribes the specified target.

Format

```
emcli db_software_maintenance -subscribeTarget
  -target_name= "target_name"
  -target_type= "target_type"
  -image_id= "image_id"
  [-version_id= "version_id']
```

```
[ ]  indicates that the parameter is optional.
```

- target_name

  The name of the target.

- target_type

  The target type.

- image_id

  The image id.

- version_id

  The version id.

Examples

The following example unsubscribes the Oracle Cloud Zone target with the name `POOL NAME` and the image ID `FE55AD7AB28974EFE04313B2F00AD4A0`.

```
emcli db_software_maintenance -unsubscribeTarget
  -target_name="POOL NAME"
  -target_type=oracle_cloud_zone
  -image_id=FE55AD7AB28974EFE04313B2F00AD4A0
```

createSoftwareImage

Description

Creates a new software image for the specified the Oracle home. The createSoftwareImage verb either takes data from a text file or uses the getInputVariableList command.

Format

```
emcli db_software_maintenance -createSoftwareImage
[-input_file="data:<file_location>"]
[-getInputVariableList= "getInputVariableList"]
[-workDir= "Work Directory Location"]

[ ]  indicates that the parameter is optional.
```

Options

- input_file

  The path of the txt input file in the format: `data:input_file_path`

- getInputVariableList

  Provides the list of variables to be specified in the input file.

- workDir

  Specify the working directory to be used. This is an optional parameter.

Examples

Example 1

To get the list of all of the parameters to be passed inside the data file, run the
following command:

```
emcli db_software_maintenance -createSoftwareImage
  -getInputVariableList=true
```

Example 2

Create a new text file called `input_rac.txt` and paste the following:

```
IMAGE_NAME=DbGoldImage
IMAGE_DESCRIPTION=Gold Image for 11g db
REF_TARGET_NAME=ORACLE_HOME
IMAGE_SWLIB_LOC=Oracle Home Provisioning Profiles/11.2.0.3.0/linux_x64
REF_HOST_CREDENTIALS=ZONE_CREDS:TESTSUPERADMIN
WORKING_DIRECTORY=/tmp
STORAGE_TYPE_FOR_SWLIB=OmsShared
STORAGE_NAME_FOR_SWLIB=swlib
VERSION_NAME=Version1
```

Run the following example to create a new image and a version.

```
emcli db_software_maintenance -createSoftwareImage
  -input_file="data:/input_rac.txt"
```

searchImage

Description

Searches the image based on the filters provided. Use '%' for wildcards. If there is no
filter specified, it returns all the images present in the system.

Format

```
emcli db_software_maintenance -searchImage
  [-name_pattern= "name_pattern"]
  [-version_pattern= "version_pattern"]
  [-description_pattern= "description_pattern"]
  [-owner= "owner"]
  [-target_type= "target_type"]
  [-platform_id= "platform_id"]
```

[ ] indicates that the parameter is optional.

- name_pattern

  The name pattern.

- version_pattern

  The version pattern.

- description_pattern

  The description pattern.

- version_pattern

  The version pattern.

- target_type

  The target type.

- platform_id

  The platform id, for example 226 for Linux x86_64.

Examples

Example 1

The following example searches the database image for names that contain GI.

```
emcli db_software_maintenance -searchImage
  -name_pattern="%GI%"
```

Example 2

The following example searches the database image for platform id 226.

```
emcli db_software_maintenance -searchImage
  -platform_id="226"
```

createVersion

Description

Creates a new version in an existing image using an existing software library component.

Format

```
emcli db_software_maintenance -updateVersionStatus
  -version_name= "version_name"
  -image_id= "image_id"
```

```
-external_id= "external_id"
-status= "status"
```

Options

- version_name

  The name of the version.

- image_id

  The ID of the image.

- external_id

  The external ID of the version. For example, it will be the Uniform Resource Name (URN) of the Software Library gold image.

- status

  The status of the version, for example DRAFT, ACTIVE, CURRENT, RESTRICTED.

Examples

The following example creates a version Version1.

```
emcli db_software_maintenance -createVersion
  -version_name="Version1"
  -image_id="01B5F14FD57D7B89E05313B2F00A739F"
  -external_id="oracle:defaultService:em:provisioning:1:cmp:COMP
  Component:SUB_OracleDB:0191172464DD36B6E05313B2F00AB90A:0.1"
  -status=CURRENT
```

updateVersionStatus

Description

Updates the version status of the image.

Format

```
emcli db_software_maintenance -updateVersionStatus
  [-version_id= "version_id"]
  -status= "status"

[ ]  indicates that the parameter is optional.
```

Options

- version_id

  The version id.

- status

  The status of the version, for example DRAFT, ACTIVE, CURRENT, RESTRICTED.

Examples

The following example updates the version of the image with the version ID 02A635AOD8D904A4E05362F3E40ADFD8 to CURRENT.

```
emcli db_software_maintenance -updateVersionStatus
  -version_id=02A635AOD8D904A4E05362F3E40ADFD8
  -status=CURRENT
```

getVersions

Description

Returns a list of the versions for the specified image.

Format

```
emcli db_software_maintenance -getVersions
  -image_id= "image_id"
  [-version_status= "version_status"]
```

```
[ ]  indicates that the parameter is optional.
```

- image_id

  The name of the image.

- version_status

  The status filter for the version.

Examples

The following example returns a list of versions for the image with the ID `01B5F14FD57D7B89E05313B2F00A739F`.

```
emcli db_software_maintenance -getVersions
     -image_id="01B5F14FD57D7B89E05313B2F00A739F"
     -version_status=CURRENT
```

getImageSubscriptions

Description

Returns the list of subscribed targets.

Format

```
emcli db_software_maintenance -getImageSubscriptions
  -image_id= "image_id"
```

- image_id

  The ID of the image.

Examples

The following example returns a list of targets for the image with the ID `ID01B5F14FD57D7B89E05313B2F00A739F`.

```
emcli db_software_maintenance -getImageSubscriptions
  -image_id="01B5F14FD57D7B89E05313B2F00A739F"
```

ORACLE®

getSubscriptionsForContainer

Description

Returns the subscriptions for the container target, for example database pool.

Format

```
emcli db_software_maintenance -getSubscriptionsForContainer
  -image_id= "image_id"
  -target_name= "target_name"
  -target_type= "target_type"
```

- image_id

    The image id.

- target_name

    The name of the target in Oracle Enterprise Manager.

- target_type

    The target type.

Examples

The following example returns the subscriptions for the Oracle Cloud Zone target with the name `POOL NAME` and the image ID `FE55AD7AB28974EFE04313B2F00AD4A0`.

```
emcli db_software_maintenance -getSubscriptionsForContainer
  -target_name="POOL NAME"
  -target_type=oracle_cloud_zone
  -image_id=FE55AD7AB28974EFE04313B2F00AD4A0
```

Fleet Management Software

performOperation -Deploy

Description

The deploy command automatically uses the current version of the subscribed image while creating the new Oracle Home.

Format

```
emcli db_software_maintenance -performOperation
  -name=""
  -purpose=""
  [-description=""]
  [-reportOnly="<true|false>"]
  [-workDir=""]
  [-start_schedule=""]
  [-inputfile="data:<file_location>"]
  [-target_type=""]
  [-target_list=""]
  [-normal_credential="<cred_name:cred_owner>"]
  [-privilege_credential="<cred_name:cred_owner>"]
  [-sql_sec="<sql_sec>"]
  [-rolling="true|false"]
```

```
[-node_list=""]
[-startupDBAAfterSwitch="<true|false>"]
[-skipSwitchDatabase="<true|false>"]
[-startupDatabase="<true|false>"]
[-ignoreStandbyPrereq="<true|false>"]
[-standbyAutoDeploy="<true|false>"]
[-blackoutguid="<true|false>"]
```

[ ] indicates that the parameter is optional.

- name

  This is the unique name of the operation.

- purpose

  There are standard purposes that can be performed by fleet operations which can be:

  – DEPLOY_DB_SOFTWARE

  – DEPLOY_RAC_SOFTWARE

  – DEPLOY_SIHA_SOFTWARE

  – DEPLOY_GI_SOFTWARE

  – DEPLOY_CBD_SOFTWARE

- description

  Description of the operation.

- reportOnly

  This is used in case of CLEANUP_SOFTWARE where the user wishes to clean Oracle Homes that are candidates for deletion, but does not want to perform the actual cleanup operation.

- workDir

  Work directory location.

- target_type

  The type of target on which this operation is being performed.

- target_list

  This is a comma separated list of targets that need to be patched.

- normal_credential

  This must be entered in the format <Named Credential: Credential Owner> where:

  – <Named Credential>: Named credential for the host where new Oracle home will be deployed.

  – <Credential Owner>: The Enterprise Manager user who owns this Named Credential.

- privilege_credential

  This must be entered in the format <Named Credential: Credential Owner> where:

  – <Named Credential>: Named credential for the host where new Oracle home will be deployed.

  – <Credential Owner>: The Enterprise Manager user who owns this Named Credential.

    These credentials are used to run scripts as root.

- start_schedule

  The date on which the stage and deploy is to be started if it is to be started in the future.

  Format: "start_time:yyyy/mm/dd HH:mm"

  > **Note:**
  >
  > This is an optional parameter. If no date is provided, the fleet operation will start immediately.

- Input_file

  This file contains the following entries:

  – NEW_ORACLE_HOME_LIST= <path of new Oracle home>

    This is the location where the new Oracle Home will be installed on all the hosts. The Credential Owner must have read / write access to this location.

  – workingDir=<Name of temp directory>

  – dispatchLoc=<Dispatch location>

    This is the location where all scripts will be staged on the host. These scripts will be executed as a "root" user.

  – SKIP_PREREQUISITE_CHECKS=<true|false>- The default value is false.

  – SKIP_CVU_CHECK=<true|false>- The default value is false.

  – PREREQS_ONLY=<true|false>- The default value is false. This can be used to detect any errors during pre-requisites checks. The actual deployment will not happen when the value is set to "true".

  – STANDBY_START_OPTION=<`OPEN | MOUNT | NOMOUNT | READ ONLY | READ ONLY WITH APPLY`> - If this option is provided while patching the standby database, it will be started in the specified mode after patching. If the value is `READ ONLY WITH APPLY`, the MRP process will be started automatically. If this option is not provided:

    * Rolling applicable patch: The existing `open_mode` for the standby database will be archived before patching and it will be stored for the standby database after patching. If the `open_mode` for the standby database can not be retrieved, the standby database will be started in the default mode and a warning message will be shown. The process will not be stopped for any failure.

    * Non-rolling applicable patch: The standby database will be started in default mode and MRP process will not be started up.

    > **Note:**
    >
    > This verb submits a procedure which must be completed before you can proceed with the next steps.

- sql_sec

  Sequence of CPU/PSU/EXA

- rolling

  Flag to indicate if the patches can be applied in rolling mode.

- node_list

  Comma separated list of hosts where the patching needs to be (e.g. rac_node1,rac_node2).

- startupDBAfterSwitch

  Flag to indicate if databases need to be started after the update operation only if the purpose is UPDATE_GI.

- skipSwitchDatabase

  Flag to indicate if databases need to be started after the update operation with UPDATE_RACDB or UPDATE_DB purposes.

- startupDatabase

  Flag to indicate if databases need to be started in this update operation with UPDATE_RACDB or UPDATE_DB purposes for standby databases in conjunction with 'skipSwitchDatabase' flag.

- ignoreStandbyPrereq

  Updating a primary database requires that all standby attached to it sa flag to indicate if this prereqignored.

- standbyAutoDeploy

  Flag to indicate if deploy function needs to be run for standby

- blackoutguid

  Blackout guid to be passed to perform operation of UPDATE_RACDB purposparameter node_list is passed.When passed, the update procedure will not create a new blackout on the database instance and wi blackoutguid.

Examples

The following example goes over the deployment of Oracle Home.

Attach Oracle Home:

To merge Oracle Homes of two databases, provide the flag `attach_home=true` and type the same location in the `input_file` where the same version of the image is already deployed.

For more information, see Attach Oracle Homes in *Enterprise Manager Lifecycle Management Administrator's Guide*.

```
emcli db_software_maintenance -performOperation -name="Deploy Home"
 -purpose="DEPLOY_DB_SOFTWARE"
 -target_type=oracle_database
 -target_list=db1221
 -normal_credential="NORMAL:SYSMAN"
 -privilege_credential="ROOT:SYSMAN"
 -input_file="data:/deploy_OH.prop"
 -attach_home=true
```

Deploy GI Software:

```
emcli db_software_maintenance -performOperation -name="Deploy
 -1120407 GI Home"
```

```
-purpose=DEPLOY_GI_SOFTWARE
-target_type=input_file
-target_list="CLUSTER1"
-normal_credential="NC_HOST_CREDS:TESTSUPERADMIN"
-privilege_credential="HOST_PRIV:TESTSUPERADMIN"
-inputfile="data:/usr/oracle/deploy.txt"
```

Deploy Container Databases:

```
emcli db_software_maintenance -performOperation
-purpose="DEPLOY_CDB"
-target_name="<>CDB Name that is being patched"
-target_type="<target type>
-name="Operation Name"
-description="Operation description"
-db_prefix | db_name ="<DB Name prefix or DB name>"
-normal_credential="<credential name>"
-privilege_credential="<credential name>"
-database_credential="SYSDBA credential name>"
```

If you are using existing container:

```
emcli db_software_maintenance -performOperation
-purpose="ATTACH_CDB"
-target_name="<CDB Name that is being patched>"
-target_type="<target type>"
-name="Operation Name"
-description="Operation description"
-destinationCDB ="<Container database to which PDBs will be migrated>"
-normal_credential="<credential name>"
-privilege_credential="<credential name>"
-database_credential="<SYSDBA credential name>"
```

performOperation -Update

Description

The Update command is used for migrating listeners and updating Oracle Homes with the least amount of downtime:

Format

```
emcli db_software_maintenance -performOperation
  -name=""
  -purpose=""
  [-description=""]
  [-reportOnly="<true|false>"]
  [-workDir=""]
  [-start_schedule=""]
  [-inputfile="data:<file_location>"]
  [-target_type=""]
  [-target_list=""]
  [-normal_credential="<cred_name:cred_owner>"]
  [-privilege_credential="<cred_name:cred_owner>"]
  [-sql_sec="<sql_sec>"]
  [-rolling="true|false"]
  [-node_list=""]
  [-startupDBAAfterSwitch="<true|false>"]
  [-skipSwitchDatabase="<true|false>"]
  [-startupDatabase="<true|false>"]
  [-ignoreStandbyPrereq="<true|false>"]
```

```
[-standbyAutoDeploy="<true|false>"]
[-blackoutguid="<true|false>"]
```

`[ ]  indicates that the parameter is optional.`

- name

    This is the unique name of the operation.

- purpose

    There are standard purposes that can be performed by fleet operations which can be:

    – MIGRATE_LISTENER

    – UPDATE_DB for single instance and SIHA

    – UPDATE_RACDB for RAC database and RAC one-node

    – UPDATE_GI for cluster

    – UPDATE_SIHA

- target_type

    The type of target on which this operation is being performed. This can be "rac_database" for RAC and "oracle_database" for single instance databases.

- target_list

    This is a comma separated list of targets which are to be migrated.

- normal_credential

    This must be entered in the format <Named Credential: Credential Owner> where:

    – <Named Credential>: Named credential for the host where new Oracle home will be deployed.

    – <Credential Owner>: The Enterprise Manager user who owns this Named Credential.

- privilege_credential

    This must be entered in the format <Named Credential: Credential Owner> where:

    – <Named Credential>: Named credential for the host where new Oracle home will be deployed.

    – <Credential Owner>: The Enterprise Manager user who owns this Named Credential.

        These credentials are used to run scripts as root.

- rolling

    This is an optional flag with the default value as true. The update procedure works in "Rolling Patch" mode by default but you can override this if necessary.

- start_schedule

    The date on which the stage and deploy is to be started if it is to be started in the future.

    Format: "start_time:yyyy/mm/dd HH:mm"

    > ✎ **Note:**
    >
    > This is an optional parameter. If no date is provided, the fleet operation will start immediately.

- Input_file

  This file contains the following entries:

  – NEW_ORACLE_HOME_LIST= <path of new Oracle home>

    This is the location where the new Oracle Home will be installed on all the hosts. The Credential Owner must have read / write access to this location.

  – workingDir=<Name of temp directory>

  – dispatchLoc=<Dispatch location>

    This is the location where all scripts will be staged on the host. These scripts will be executed as a "root" user.

  – SKIP_PREREQUISITE_CHECKS=<true|false>- The default value is false.

  – SKIP_CVU_CHECK=<true|false>- The default value is false.

  – PREREQS_ONLY=<true|false>- The default value is false. This can be used to detect any errors during pre-requisites checks. The actual deployment will not happen when the value is set to "true".

  – STANDBY_START_OPTION=<`OPEN | MOUNT | NOMOUNT | READ ONLY | READ ONLY WITH APPLY`> - If this option is provided while patching the standby database, it will be started in the specified mode after patching. If the value is `READ ONLY WITH APPLY`, the MRP process will be started automatically. If this option is not provided:

    * Rolling applicable patch: The existing `open_mode` for the standby database will be archived before patching and it will be stored for the standby database after patching. If the `open_mode` for the standby database can not be retrieved, the standby database will be started in the default mode and a warning message will be shown. The process will not be stopped for any failure.

    * Non-rolling applicable patch: The standby database will be started in default mode and MRP process will not be started up.

> **Note:**
>
> This verb submits a procedure which must be completed before you can proceed with the next steps.

Examples

Migrating the Listeners

If there are listeners running from the database home, use the following verb to migrate them to the parallel Oracle Home, you can use this command to migrate the listeners running from Oracle database homes. Grid Infrastructure homes are automatically migrated during the GI update process.

```
emcli db_software_maintenance -performOperation
 -name="Update Listener"
 -purpose=migrate_listener
 -target_type=oracle_database
 -target_list="DB1"
 -normal_credential="NC_HOST_CREDS:SYSMAN"
```

```
 -privilege_credential="HOST PRIV:SYSMAN"
 start_schedule
```

Updating the Database / Cluster

After the stage and deploy operations are completed, the grid infrastructure instances should be migrated to the newly deployed Grid Infrastructure Homes.

```
emcli db_software_maintenance -performOperation
 -name="Update Cluster"
 -purpose=UPDATE_GI
 -target_type=cluster
 -target_list=CLUSTER1
 -normal_credential="NC_HOST_CREDS:SYSMAN"
 -privilege_credential="HOST PRIV:SYSMAN"
 [-rolling=<true/false]
```

To update an Oracle Database, database credentials can be provided using the flag – `database_credential="DB_SYS_CREDS:SYSMAN"`. If the credentials are not provided with this flag, the preferred set credentials will be used instead.

The TDE wallet password must be provided in the input file – `input_file="data:<INPUT.PROPERTIES>"`. If the TDE configuration fails or database credentials are not provided, the database update will continue without TDE configuration.

For UPDATE_PDB:

- TDE must already be configured in the destination CDB.

- Source and destination passwords must be included in the input file.
  Example:

  ```
  srcWalletPassword=<SOURCE_PASSWORD>
  ```

  ```
  destWalletPassword=<DESTINATION_PASSWORD>
  ```

Node Wise RAC Database / Cluster Update

A cluster update always requires that all the RAC database instances running on that node be shut down during the switch process of the cluster instance. Hence, a cluster update followed by a RAC database updates results in an instance getting restarted twice.

There may be several situations where the administrator may require more control over the switch process. For example, the administrator may choose to perform a node-wise update of the cluster and RAC databases in order to avoid multiple restarts of the database instances. The administrator may also need to perform node specific pre-post steps.

```
emcli db_software_maintenance -performOperation -name="Update Cluster"
 -purpose=UPDATE_GI
 -target_type=cluster
 -target_list=CLUSTER1
 -normal_credential="NC_HOST_CREDS:SYSMAN"
 -privilege_credential="HOST_PRIV:SYSMAN"
 -rolling=true
 -node_list="host1.example.com"
 -startupDatabase=false
```

This option provides this control by enabling the user to perform the following tasks for each node:

- Switch the cluster instance.

- Leave the RAC database instances shutdown.

For example, consider RAC databases RACDB_112 and RACDB_121 are running on this cluster. The instances RACDB_112_1 and RACDB_121_1 running on this specific node will continue to remain shut down after the cluster instance is switched.

- Switch all the RAC database instances on the same node.

- This step will switch the instances RACDB_112_1 and RACDB_121_1 to the new home and will restart the same.

  For example, consider RAC databases RACDB_112 and RACDB_121 are running on this cluster. The instances RACDB_112_1 and RACDB_121_1 running on this specific node will continue to remain shut down after the cluster instance is switched.

```
emcli db_software_maintenance -performOperation
 -name="Update RAC DB"
 -purpose=UPDATE_RACDB
 -target_type=rac_database
 -target_list=RACDB
 -normal_credential="NC_HOST_CREDS:SYSMAN"
 -privilege_credential="HOST_PRIV:SYSMAN"
 -rolling=true
 -node_list="host1.example.com"
```

performOperation -Rollback

Description

This command is used to switch the database back to the previous Oracle home after the Update operation has been completed.

Format

```
emcli db_software_maintenance -performOperation
  -name=""
  -purpose=""
  [-description=""]
  [-reportOnly="<true|false>"]
  [-workDir=""]
  [-start_schedule=""]
  [-inputfile="data:<file_location>"]
  [-target_type=""]
  [-target_list=""]
  [-normal_credential="<cred_name:cred_owner>"]
  [-privilege_credential="<cred_name:cred_owner>"]
  [-sql_sec="<sql_sec>"]
  [-rolling="true|false"]
  [-node_list=""]
  [-startupDBAAfterSwitch="<true|false>"]
  [-skipSwitchDatabase="<true|false>"]
  [-startupDatabase="<true|false>"]
  [-ignoreStandbyPrereq="<true|false>"]
  [-standbyAutoDeploy="<true|false>"]
  [-blackoutguid="<true|false>"]

[ ]  indicates that the parameter is optional.
```

- name

  This is the unique name of the operation.

- purpose

  There are standard purposes that can be performed by Fleet Operations which cane be:

  – ROLLBACK_DB

  – ROLLBACK_RACDB

  – ROLLBACK_GI

  – ROLLBACK_SIHA

- target_type

  The type of target being provided in this operation which can "rac_database" or "oracle_database".

- target_list

  This is a comma separated list of targets which need to be patched.

  – Targets of homogenous types are supported in a single fleet operation.

  – A unique list of hosts based on this target list is displayed and start stage of Oracle home software on those hosts.

  – If targets running from the same Oracle home are provided in this list, the stage and deploy operation will be started only once and not for all targets.

- normal_credential

  This must be entered in the format <Named Credential: Credential Owner> where:

  – <Named Credential>: Named credential for the host where new Oracle home will be deployed.

  – <Credential Owner>: The Enterprise Manager user who owns this Named Credential.

- privilege_credential
  This must be entered in the format <Named Credential: Credential Owner> where:

  – <Named Credential>: Named credential for the host where new Oracle home will be deployed.

  – <Credential Owner>: The Enterprise Manager user who owns this Named Credential.

    These credentials are used to run scripts as root.

- non_rolling

  By default, rollback is performed on all nodes. If the non_rolling back flag is enabled, you can select the list of nodes (using the node_list command) that are to be rolled back.

- rolling

  By default rollback is performed in rolling fashion. This flag is used when the current Oracle home has patches that were applied in non-rolling mode (OJVM) and need to be rolled back.

- node_list

  This is a comma separated list of hosts on which the instances need to be updated.

  For example: If RACDB is running on a 4 node cluster host1, host2, host3, and host4 and you choose to update the instances in only 2 hosts at a time, the value of this parameter needs to be specified as node_list="host1, host2".

- Input_file

  This file contains the following entries:

– NEW_ORACLE_HOME_LIST= <path of new Oracle home>

This is the location where the new Oracle Home will be installed on all the hosts. The Credential Owner must have read / write access to this location.

– workingDir=<Name of temp directory>

– dispatchLoc=<Dispatch location>

This is the location where all scripts will be staged on the host. These scripts will be executed as a "root" user.

– SKIP_PREREQUISITE_CHECKS=<true|false>- The default value is false.

– SKIP_CVU_CHECK=<true|false>- The default value is false.

– PREREQS_ONLY=<true|false>- The default value is false. This can be used to detect any errors during pre-requisites checks. The actual deployment will not happen when the value is set to "true".

– STANDBY_START_OPTION=<`OPEN | MOUNT | NOMOUNT | READ ONLY | READ ONLY WITH APPLY`> - If this option is provided while patching the standby database, it will be started in the specified mode after patching. If the value is `READ ONLY WITH APPLY`, the MRP process will be started automatically. If this option is not provided:

* Rolling applicable patch: The existing `open_mode` for the standby database will be archived before patching and it will be stored for the standby database after patching. If the `open_mode` for the standby database can not be retrieved, the standby database will be started in the default mode and a warning message will be shown. The process will not be stopped for any failure.

* Non-rolling applicable patch: The standby database will be started in default mode and MRP process will not be started up.

> **Note:**
>
> This verb submits a procedure which must be completed before you can proceed with the next steps.

Examples

Rollback RAC DB

```
emcli db_software_maintenance -performOperation
 -name="Rollback RAC DB"
 -purpose=ROLLBACK_RACDB
 -target_type=rac_database
 -target_list=RACDB
 -normal_credential="NC_HOST_CREDS:SYSMAN"
 -privilege_credential="HOST_PRIV:SYSMAN"
 [-rolling=true/false]
 [-node_list="host1.example.com"]
```

performOperation -Cleanup

Description

User can cleanup a Oracle Home by providing the target name in `-target_list` and `-target_type=oracle_home`.

Format

```
emcli db_software_maintenance -performOperation
-name="cleanup"
-purpose="CLEANUP_SOFTWARE"
  [-description=""]
  [-reportOnly="<true|false>"]
  [-workDir=""]
  [-start_schedule=""]
  [-target_type=""]
  [-target_list=""]
  [-normal_credential="<cred_name:cred_owner>"]
  [-privilege_credential="<cred_name:cred_owner>"]
  [-force=true]
```

> **Note:**
>
> The user can perform a cleanup in two ways:
>
> - The user can pass the database in `-target_list` and pass `-target_type=<oracle_database/rac_database>` and the script will perform the cleanup to the attached Oracle Home.
>
> - The user can pass the Oracle Home as a target in `-target_list` and the script will validate and perform the cleanup.
>
> If any target is associated with the given home, Cleanup will be not done for the given home. User has to provide homes on all the nodes for a RAC before Cleanup can be done.

- description

  Description of the operation.

- reportOnly

  This is used in case of CLEANUP_SOFTWARE where the user wishes to clean Oracle Homes that are candidates for deletion, but does not want to perform the actual cleanup operation.

- workDir

  Work directory location.

- target_type

  The type of target on which this operation is being performed.

- target_list

  This is a comma separated list of targets that need to be patched.

- normal_credential

  This must be entered in the format <Named Credential: Credential Owner> where:

  – <Named Credential>: Named credential for the host where new Oracle home will be deployed.

  – <Credential Owner>: The Enterprise Manager user who owns this Named Credential.

- privilege_credential

  This must be entered in the format <Named Credential: Credential Owner> where:

  – <Named Credential>: Named credential for the host where new Oracle home will be deployed.

  – <Credential Owner>: The Enterprise Manager user who owns this Named Credential.

    These credentials are used to run scripts as root.

- start_schedule

  The date on which the stage and deploy is to be started if it is to be started in the future.

  Format: "start_time:yyyy/mm/dd HH:mm"

> **Note:**
>
> This is an optional parameter. If no date is provided, the fleet operation will start immediately.

- force
  This is an optional flag. If the cleanup of oracle home affects the database, the user can provide <-force=true> flag to skip this verification. Default value for this flag is false.

Example

```
emcli db_software_maintenance -performOperation
-name="cleanup"
-purpose="CLEANUP_SOFTWARE"
-target_list=OraDB12Home_host.com:1111
-target_type=oracle_home
-normal_credential=NORMAL:SYSMAN
-privilege_credential=ROOT:SYSMAN
-workDir=/tmp
```

dbimport

Imports data from export dumps to the database target specified.

Format

```
emcli dbimport        -inputFile="File containing properties for importing data
to a database"
```

Options

- inputFile

  Location of the file containing properties required for importing data to the database.

Example

The following example imports data from export dumps to the database target specified in the `dbimport.props` file:

```
emcli dbimport
      -inputFile=/u01/files/dbimport.props
```

deactivate_mda_finding_types

Deactivates the specified MDA finding types. A finding type can have status 'N' (new), 'A' (active), or 'I' (inactive). Only those finding types that are currently active, for example, status 'A', will be deactivated. For others the previous status is retained. Once deactivated, no new targets will be enabled for the finding type, nor will analysis runs be scheduled for the inactive finding type.

Format

```
emcli deactivate_mda_finding_types      [-finding_types="<list of finding
types>"]       [-separator=finding_types="separator_for_finding_types_values"]

[ ]  indicates that the parameter is optional.
```

Options

- finding_types

  List of finding types. The default separator to be used is ';'.

- separator=finding_types

  Indicates the custom separator used for the list of finding types. This option is mandatory if any other character apart from ';' is used as a separator in the finding types list.

Examples

Example 1

The following example deactivates a single finding type:

```
emcli deactivate_mda_finding_types
      -finding_types="oracle.sysman.emas.wls_gc_overhead"
```

Example 2

The following example deactivates multiple finding types:

```
emcli deactivate_mda_finding_types
      -
finding_types="oracle.sysman.emas.wls_gc_overhead;oracle.sysman.emas.wls_heap_config"
```

define_diagcheck_exclude

Defines a diagnostic check exclusion with regard to groups and checks to exclude.

Format

```
emcli define_diagcheck_exclude
        -target_type="type"
        -exclude_name="name"
        { [-excl_group="diag_group" ]*
          [-excl_check="diag_check" ]* |
         -input_file=excl_def:<complete_path_to_file> }
```

[ ] indicates that the parameter is optionalis optional

Options

- **target_type**

  Type of target.

- **exclude_name**

  Name to use for the exclusion.

- **excl_group**

  Group of diagchecks to exclude.

- **excl_check**

  Name of diagcheck to exclude.

- **input_file**

  For information about the input_file parameter, see -input_file Syntax Guidelines.

delete_assoc

Deletes target association instances.

Format

Standard Mode

```
emcli delete_assoc
      -assoc_type="association type"
      -source="target_name:target_type"
      -dest="target_name1:target_type1[;target_name2:target_type2..]"
      [-separator="separator:attribute_name:character"]
      [-subseparator="subseparator:attribute_name:character"]
```

Interactive (Script) Mode

```
delete_assoc(
     assoc_type="association type"
     ,source="target_name:target_type"
     ,dest="target_name1:target_type1[;target_name2:target_type2..]"
     [,separator="separator:attribute_name:character"
     [,subseparator="subseparator:attribute_name:character"]        )
```

[ ] indicates that the parameter is optional.

Options

- **assoc_type**

Association type.

- **source**

  Target name and target type of the source target.

- **dest**

  Target name and target type of the destination targets.

- **separator**

  By default, multi-value input attributes use a semicolon (;) as a separator. Specifying this option overrides the default separator value.

  Example: `separator="<attribute_name=sep_char>"` where `attribute_name` is name of the attribute for which you want to override the separator character, and `sep_char` is the new separator character.Example: `separator="att=#"`

- **subseparator**

  By default, multi-value input attributes use a colon (:) as a subseparator. Specifying this option overrides the default subseparator value.

  Example: `subseparator="<attribute_name=sep_char>"` where `attribute_name` is name of the attribute for which you want to override the separator character, and `sep_char` is the new subseparator character.Example: `separator="att=#"`

Exit Codes

0 indicates that the verb processing was successful.

Non-zero values indicate that the verb processing was not successful.

Example

This example deletes associations of type cluster_contains from target "abc_cluster:cluster" to targets "def.oracle.com:host" and "ghi.oracle.com:host":

```
emcli delete_assoc
     -assoc_type="cluster_contains"
     -source="abc_cluster:cluster"
     -dest="def.oracle.com:host;ghi.oracle.com:host"
```

delete_bda_cluster

Deletes the specified Hadoop cluster target and all its children. If this is the last cluster in the BDA target, also deletes the BDA target and all its children.

If Hadoop clusters are spread across multiple racks, performs deletions across the BDA rack. If other clusters exist within the BDA rack, relocates any shared targets before deleting the Hadoop cluster target.

Format

```
emcli delete_bda_cluster
        -cluster="cluster_name"
```

Options

- **cluster**

  Name of the cluster to be deleted.

Examples

The following example deletes the acme cluster target and all of its children. If acme is the last cluster in the BDA rack, deletes the rack and all of its children. If there are other clusters in the rack, relocates shared targets before deleting the cluster.

```
emcli delete_bda_cluster
        -cluster="acme"
```

delete_blackout

Deletes a blackout that has already ended or has been fully stopped. You cannot delete a blackout that is either in progress or currently scheduled. You must first run stop_blackout.

Format

```
emcli delete_blackout
      -name="name"
      [-createdby="blackout_creator"]
      [-emd_url="emd_url"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- name

  Name of the blackout to delete.

- createdby

  Enterprise Manager user who created the blackout. The default is the current user. The SUPER_USER privilege is required to delete a blackout created by another user. For blackouts created via emctl, -createdby="<SYSTEM>" should be specified.

- emd_url

  The emd_url of the agent through which emctl blackout was created. This is a mandatory parameter for blackouts created through emctl.

Examples

Example 1

This example deletes blackout backup_monthly created by the current user.

```
emcli delete_blackout
      -name=backup_monthly
```

Example 2

This example deletes the blackout black_2015-04-14 04:45:10 which was created via emctl for the given unique name.

```
emcli delete_blackout
      -name="black_2015-04-14 04:45:10"
      -createdby="<SYSTEM>"
      -emd_url="https://myhost.company.com:1234/emd/main/"
```

delete_charge_item

Deletes the custom charge item from Chargeback.

Format

```
emcli delete_charge_item
        -target_type="target_type"
        -item_name="item_name"
```

Options

- **target_type**

  Target type associated with the custom charge item.

- **item_name**

  Name of the custom charge item to be deleted.

Examples

Example 1

This example deletes a custom charge item named total_proc associated with the host target type:

```
emcli delete_charge_item
        -target_type="host"
        -item_name="total_proc"
```

Example 2

This example deletes a custom charge item named custom_config associated with the Oracle Database target type:

```
emcli delete_charge_item
        -target_type="oracle_database"
        -item_name="custom_config"
```

delete_chef_cookbook

Deletes a software library component and directives for a chef cookbook. Use this verb once for each cookbook.

Format

```
emcli delete_chef_cookbook
      -name="component_name"
      -folder_name="swlib_folder_name"
```

Options

- name

  The software library component name

- folder_name

  The software library folder where the component and directives were saved.

Example

The following example deletes a custom software library component and directives for a chef cookbook.

```
emcli delete_chef_cookbook
    -name=component
    -folder_name="MyComponents"
```

delete_cloud_service_instances

Deletes the cloud service instances based on the specified filter.

Format

```
emcli delete_cloud_service_instances
    -user="username"
    [-family="family"]
    [-type="service type"]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **user**

Identifies the name of the user to be used for filtering the service instances that are to be deleted.

• **family**

Identifies the service family name to use to filter cloud requests.

• **type**

Identifies the Service Type to be used for filtering the service instances that are to be deleted.

Examples

Example 1

This example deletes all cloud instances that are owned by a specified user (user1) and belong to a specified service family (family1):

```
emcli delete_cloud_service_instances -user="user1" -family="family1"
```

Example 2

This example deletes all cloud instances that are owned by a specified user (user1), belong to a specified service family (family1), and belong to a specified service type (type1):

```
emcli delete_cloud_service_instances -user="user1" -family="family1" -
type="type1"
```

delete_cloud_user_objects

Deletes cloud user objects including cloud service instances and requests.

Format

```
emcli delete_cloud_user_objects
      -user="username"
      [-purge]
      [-force]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **user**

  Identifies the name of the user to be used for filtering user objects.

- **purge**

  Sets a flag to purge the completed cloud service requests. Default is **false** unless this option is used.

- **force**

  Sets a flag to attempt to cancel In Progress requests. Depending on the job state, there may be some manual cleanup required.

  USE WITH CAUTION. There is no way to undo the operation once started.

Examples

Example 1

Delete all cloud objects owned by a specified user (`user1`) and cancel all scheduled requests:

```
emcli delete_cloud_user_objects -user="user1"
```

Example 2

Delete all cloud objects owned by a specified user (`user1`), cancel all scheduled requests, and purge all completed requests:

```
emcli delete_cloud_user_objects -user="user1" -purge
```

delete_compare_check

Deletes a comparison check for the specified target type. The latest comparison check is deleted.

Format

```
emcli delete_compare_check
      -name="<check_name>"
      -target_type="<target_type>"
```

Options

- name

  Name of the comparison check being deleted.

- target_type

  Target type in which the comparison check is being deleted.

Example

The following example deletes the check_deployment_my_target comparison check from the host target.

```
emcli delete_compare_check
    -name="check_deployment_my_target"
    -target_type="host"
```

delete_config_compare_template

Deletes the specified configuration comparison template from the repository.

Format

```
emcli delete_config_compare_template        -template_name="<template_name>"
```

Options

* template_name

    Name of the configuration comparison template.

Example

The following example deletes the configuration comparison template with template_name "Host Template"

```
emcli delete_config_compare_template
    -template_name="Host Template"
```

delete_config_onetimecompare

Deletes the saved one-time comparison from the repository.

Format

```
emcli delete_config_onetimecompare
    -name="<comparison_name>"
```

Options

* name

    Name of the saved configuration comparison.

Example

The following example deletes the saved one-time comparison from the repository with the name "cmp_host1".

```
emcli delete_config_onetimecompare
    -name="cmp_host1"
```

delete_config_search

Deletes the saved configuration search from the repository.

Format

```
emcli delete_config_search
      -name="saved_search_name"
      [-search_type="Latest/History"]
```

`[ ]  indicates that the parameter is optional.`

Options

* name

  The name of saved search.

* search_type

  The saved search type, either Latest or History. This is an optional parameter with Latest as default value.

Example

The following example deletes the saved search 'search_07_15-2015':

```
emcli delete_config_search       -name="search_07_15_2015"
      -search_type="Latest"
```

delete_credential_set

Deletes a credential set. Only Enterprise Manager Super Administrators can delete credential sets. Out-of-box credential sets cannot be deleted.

Format

```
emcli delete_credential_set        -set_name="set_name"        -target_type="ttype"
```

Options

* **set_name**

  Credential set name to be deleted.

* **target_type**

  Target type of the credential set.

Examples

This example deletes a credential set named Old_Credential_Set.

```
emcli delete_credential_set
      -set_name=Old_Credential_Set
      -target_type=host
```

delete_custom_plugin_update

Deletes the custom plug-in update for a plug-in. All subsequent plug-in deployments will use the latest applicable version or revision available with Enterprise Manager Self Update.

Does not automatically redeploy to Management Agents on which this custom plug-in update was previously deployed. Applies only to subsequent plug-in deployments.

Format

```
emcli delete_custom_plugin_update
     -plugin="<plugin_id>:<plugin_version>:<plugin_revision>"
```

Options

- **plugin**

  ID, version, and revision of the plug-in. To view the version and revision of a plug-in, run 'emcli list_custom_plugin_updates'.

Example

The following example deletes the custom plug-in update of the 12.1.0.2.0 version of the `oracle.sysman.db2` plug-in.

```
emcli delete_custom_plugin_update        -plugin="oracle.sysman.db2:12.1.0.2.0"
```

delete_database

Deletes a database and target from Oracle Enterprise Manager.

Format

```
emcli delete_database
     -inputFile="File containing properties required for deleting a database"
```

Options

- inputFile

  The location and name of the file containing the properties required for deleting the database.

Example

The following example deletes a database using the parameters contained in the `/u01/files/delete_database.props` file:

```
emcli delete_database        -inputFile=/u01/files/delete_database.props
```

delete_database_size

Deletes the database size created with the create_database_size verb.

Format

```
emcli delete_database_size
     -name="<Existing size name>"
```

Options

- name

  The name of the existing database size.

Example

The following example deletes the database size names Small.

```
emcli delete_database_size
      -name=Small
```

delete_dbaas_quota

Deletes the database quota for an SSA user role.

Format

```
emcli delete_dbaas_quota
        -role_name="<SSA user role  name>"
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   **role_name**

    Name of the SSA user role for which the quota is to be deleted.

Example

This example deletes the quota for My Role:

```
emcli delete_dbaas_quota
        -role_name="My Role"
```

delete_dbprofile

Deletes an existing database profile component.

Format

```
emcli delete_dbprofile     -comp_loc="Database Profile component location and name in
software library"    [-version="Database Profile component version name"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   comp_loc

    Combination of database profile component location and name.

*   version

    Database profile component version name.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example deletes Database profile component with the profile name "RMAN_Profile", version "RMAN_Backup_10_04_14_12_40_PM" and location "Database Provisioning Profiles/11.2.0.4.0/linux_x64".

```
emcli delete_dbprofile -comp_loc="Database Provisioning Profiles/11.2.0.4.0/
linux_x64/RMAN_Profile" -version="RMAN_Backup_10_04_14_12_40_PM"
```

delete_diag_snapshot

Deletes a specified diagnostic snapshot.

Format

```
emcli delete_diag_snapshot
        -name="<diag_snapshot_name>"
        [-debug]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **name**

Name of the diagnostic snapshot to be deleted. Ensure that the diagnostic snapshot exists for the specified name.

• **debug**

Runs the verb in verbose mode for debugging purposes.

Examples

This example deletes a diagnostic snapshot with the name of Snapshot1 from Cloud Control.

```
emcli delete_diag_snapshot
     -name="Snapshot1"
```

delete_fmw_profile

Deletes a Fusion Middleware provisioning profile from software library.

Format

```
emcli delete_fmw_profile
     -location="Profile Location"
      -source="source"
      -dest="association type"
      [-separator="separator:attribute_name:character"]
      [-subseparator="subseparator:attribute_name:character"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

• location

The complete software library path to the profile. Use the list_fmw_profiles verb to identify the complete path.

> **Note:**
>
> The name and owner parameters must be used together.

Example

The following example deletes the Fusion Middleware profile "MyProfile" from software library.

```
emcli delete_fmw_profile        -location="Fusion Middleware Provisioning/Profiles/
MyProfile"
```

delete_from_target_properties_master_list

Deletes a property from the target properties master list.

Format

Standard Mode

```
emcli delete_from_target_properties_master_list
      -property_name="null"
      -property_value="null"
```

Interactive or Script Mode

```
delete_from_target_properties_master_list(
property_name="null"
,property_value="null"
)
```

```
[ ] indicates that the parameter is optional. Delete if the verb has no optional
parameters
```

Options

- property_name

  The name of the property to delete.

- property_value

  The value of the property to delete.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

delete_gold_agent_image

Deletes a Management Agent gold image version.

Format

```
emcli delete_gold_agent_image
     -version_name="gold_image_version_name_to_delete"
```

Options

• version_name

  Management Agent gold image version that you want to delete.

Example

The following example deletes the Management Agent gold image
`OPC_AGI_DB_JUL_13`.

```
emcli delete_gold_agent_image        -version_name=OPC_AGI_DB_JUL_13
```

delete_group

Deletes a group. Deleting a non-existent group generates the error "Group X does not exist."

Format

```
emcli delete_group
     -name="name"
     [-type=<group>]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **name**

  Name of the group to delete.

• **type**

  Group type: group. Defaults to "group".

Examples

Example 1

This example removes the group `payroll_group` that consists of database target types.

```
emcli delete_group -name=payroll_group
```

Example 2

This example removes the group `my_hosts` that consists of host target types.

```
emcli delete_group -name=my_hosts
```

delete_incident_record

Deletes one or more open incidents based on the provided IDs, up to a maximum of 20 incidents. This removes any *association* with the underlying events and annotates

them accordingly. Incident deletion **does not** remove the actual underlying events: These events will remain open.

Privilege Requirements: Only users with Manage Incident privilege can delete the incident.By default, incidents that have workflow attributes (such as Escalation, Priority, Resolution Status, Acknowledgement, Owner Assignment, or Suppression) set to non-default values will not be deleted unless the `-force` option is used.Closed incidents, diagnostic (ADR) incidents, and incidents with tickets created cannot be deleted.The status of each incident deletion is displayed upon command execution.

Format

```
emcli delete_incident_record
-incident_number_list="Comma-separated list of incident numbers"
[-force]
[-preview]

[ ]  indicates that the parameter is optional
```

Options

• **incident_number_list**

  Comma-separated list of incident numbers (up to 20) to be deleted.

• **force**

  Deletes incidents without checking for their non-default workflow values.

• **preview**

  Displays whether or not specified incidents (by incident number) can be deleted.

Examples

Example 1

This example displays whether or not incidents 173, 1886, 32, 5, and 853 can be deleted.

The command output is shown below.

```
emcli delete_incident_record -incident_number_list="173,1886,32,5,853" -preview

=========
RESULTS
=========
=> Incident 173 can be deleted.

=> Incident 1886 can only be deleted using the -force option, as one or more incident
workflow attributes have been been used.

=> Incident 32 cannot be deleted because there is ticket attached with the incident.

=> Incident 5 cannot be deleted because user AdminX does not have at least a manage
incident privilege.

=> Incident 853 can be deleted.
```

Example 2

This example deletes incidents 178, 1886, and 853 without checking for non-default incident workflow values. The command output is shown below.

```
emcli delete_incident_record -incident_number_list="173,1886,853" -force

=========
RESULTS
=========

=> Incident 173 has been successfully deleted.

=> Incident 1886 has been successfully deleted.

=> Incident 853 has been successfully deleted.
```

delete_instance

Deletes a stopped or completed deployment instance. An instance can only be deleted when its status is stopped, completed, or completed with an error.

Format

```
emcli delete_instance
      [-instance=<instance_guid>]
      [-exec=<execution_guid>]
      [-name=<execution_name>]
      [-owner=<execution_owner>]

[ ]  indicates that the parameter is optional
```

Options

- **instance**

  Instance GUID.

- **exec**

  Execution GUID.

- **name**

  Execution name.

- **owner**

  Execution owner.

Examples

Example 1

```
emcli delete_instance -instance=16B15CB29C3F9E6CE040578C96093F61
```

Example 2

```
emcli delete_instance -exec=2B15CB29C3F9E6CE040578C96093F16
```

delete_job

Deletes a job or a set of jobs matching the filter criteria. A job cannot be deleted if any of its executions are active. All executions must be in one of the following states:

ABORTED, FAILED, COMPLETED, STOPPED, SKIPPED

Use the `get_jobs` verb to obtain a list of existing jobs along with their job IDs and statuses.

Format

```
emcli delete_job
      [-job_id="ID1;ID2;..."]
      [-name="job name pattern"]
      [-owner="job owner"]
      [-type="job type"]
      [-targets="target name:target type"]
      [-input_file=property_file:"filename"]
      [-preview]

[ ]  indicates that the parameter is optional
```

Options

- **job_id**

  Semi-colon (;) separated list of job(s) to delete.

  NOTE: This filter cannot be used with other filters.

- **name**

  Name or pattern of the job to delete. To uniquely identify the job, the current user is used.

- **owner**

  Owner of the job(s).

- **type**

  Job type of the job(s).

- **targets**

  Target name and target type of the job(s) to be deleted.

- **input_file**

  The properties for filtering jobs can be specified in "filename". Any jobs matching all the specified filter criteria are deleted. You must specify at least one filter, and the logged in administrator must have the necessary privileges on the matching jobs.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **preview**

  Lists only the jobs to be deleted. In the list of options you specify, if -preview is not one of the options, jobs are deleted, and then these jobs will be listed. If -preview is one of the options, the identical list is shown, but no jobs are deleted.

Examples

Example 1

This example deletes an existing job with the job ID 123456789012345678901234356789012.

```
emcli delete_job -job_id=123456789012345678901234356789012
```

Example 2

This example stops and deletes a job named MY_JOB owned by the logged in administrator. You can use the stop and delete pattern to delete active jobs.

```
emcli stop_job -name=my_job
emcli delete_job -name=my_job
```

delete_library_job

Deletes a library job you created using the create_library_jobs command.

Format

```
emcli delete_library_job
     -name=<"library_job_name">
     [-owner=<"library_job_owner">]
```

[ ] indicates that the parameter is optional

Options

- **name**

  Name of the library job.

- **owner**

  Owner of the library job if different from the current logged-in EM CLI
  administrator.

Examples

Example 1

This example deletes the library job "libjob1" owned by the current logged-in
Enterprise Manager administrator.

```
emcli delete_library_job -name=libjob1
```

Example 2

This example deletes the library job "libjob2" owned by the Enterprise Manager
administrator "emadmin1."

```
emcli delete_library_job -name=libjob2 -owner=emadmin1
```

delete_metric_promotion

Deletes a promoted metric.

Format

```
emcli delete_metric_promotion
     -name=<service_target_name>
     -type=<service_target_type>
     [-category=<usage/performance/business>]
     [-promotedMetricName=<promoted_metric]
     [-promotedMetricColumn=<promoted_metric_column>]
     -promotedMetricKey=<key_value_of_promoted_metric>
```

[ ] indicates that the parameter is optional

Options

- **name**

Name of the service target.

- **type**

  Name of the service type.

- **category**

  Defines whether the promoted metric is a usage or a performance metric of a service. This determines the promoted metric name and metric column. If you do not specify this, you must specify the `promotedMetricName` and `promotedMetricColumn`.

- **promotedMetricName**

  Promoted metric name. This is optional if you specify the category .

- **promotedMetricColumn**

  Promoted metric column. This is optional if you specify the category .

- **promotedMetricKey**

  Determines the key value of the promoted metric. It is equivalent to the displayed name of the promoted metric in the user interface.

Examples

This example deletes the promoted performance metric with the key value `mymetric1` on the service `MyTarget`.

```
emcli delete_metric_promotion -name='MyTarget' -type='generic_service'
     -category=Performance -promotedMetricKey=mymetric1
```

delete_mw_profile

Deletes a non-Oracle Middleware Provisioning Profile from software library.

Examples of non-Oracle middleware include Apache Tomcat, JBoss, etc.

For Oracle Middleware Provisioning Profile, refer to other verbs such as delete_fmw_domain_profile or delete_fmw_home_profile.

Format

```
emcli delete_mw_profile
     -location="Profile Location"
```

Options

location

Complete software library path to the Profile. Use list_mw_profiles to know the complete path.

Examples

The following example deletes the profile named 'MyProfile':

```
 emcli delete_mw_profile
     -location="Middleware Provisioning/Generic Profiles/MyProfile"
```

delete_named_credential

Deletes an existing named credential.

Format

```
emcli delete_named_credential
      -cred_owner=<owner>
      -cred_name=<name>
```

Options

- **cred_owner**

  Credential owner.

- **cred_name**

  Required credential name. This does not support wild cards.

delete_operation_plan

Deletes the specified operation plan from a Site Guard configuration.

Format

```
emcli delete_operation_plan
      -name=<plan_name>
```

Options

- **name**

  Name of the operation plan you want to delete.

Example

```
emcli delete_operation_plan
      -name="BISystem1-switchover"
```

delete_oracle_database

De-configures and deletes Oracle Database.

Format

```
emcli delete_oracle_database
      -input_file=data:"absolute file path"
      [-instances|-software|-all]
      [-schedule=
          [frequency:interval|weekly|monthly|yearly];
          start_time:yy-MM-dd HH:mm;
          end_time:yy-MM-dd HH:mm;
          [repeat:#m];
          [months:#,#,#,...];
          [days:#,#,#,...];
          [tz:{timezone ID}];
          [grace_period:xxx]
      ]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **software**

  Deletes the empty SIDB home.

- **listeners**

  Deletes or de-configures listeners running from the oracle home.

- **all**

  Deletes the instances, listeners, and the software home.

- **input_file**

  Property file that contains necessary parameters to perform de-configure and delete Oracle Database. Use "emcli describe_input -oracle_database" command to get the list of properties for performing cleanup operation.

- **schedule**

  – frequency: Frequency type with which the Oracle Restart is removed. It can be interval (in minutes), weekly, monthly, or yearly

  – start_time: Denotes the starting time of Cleanup Oracle Restart in the format yy-MM-dd HH:mm

  – end_time: Denotes the end time of Cleanup Oracle Restart in the format yy-Mm-dd HH:mm

  – repeat: Repetition rate at Cleanup Oracle Restart. If the frequency is interval, then repeat will be in minutes

  – months: Number of months after which repetition of Cleanup Oracle Restart will occur

  – days: Number of days after which repetition of Cleanup Oracle Restart will occur

  – tz: Time Zone ID

Examples

Example 1

This example cleans the Oracle Database Instances based on property file "deletesiha.txt" with the specified schedule.

```
emcli delete_oracle_database -instances
        -input_file="data:/tmp/deletesidb.txt"
        -schedule="frequency:interval;start_time:15-10-27 05:30;end_time:14-10-12
05:23;repeat:30;grace_period:60;tz:America/New_York"
```

Example 2

This example cleans the Oracle Database based on property file "deletesiha.txt".

```
emcli delete_oracle_database -all
        -input_file="data:/tmp/deletesiha.txt"
```

delete_oracle_restart

De-configures and deletes Oracle Restart (SIHA).

Format

```
emcli delete_oracle_restart
        -input_file=data:"absolute file path"
        [-instances|-software|-all]
        [-schedule=
            [frequency:interval|weekly|monthly|yearly];
            start_time:yy-MM-dd HH:mm;
            end_time:yy-MM-dd HH:mm;
            [repeat:#m];
            [months:#,#,#,...];
            [days:#,#,#,...];
            [tz:{timezone ID}];
            [grace_period:xxx]
        ]
```

`[ ]  indicates that the parameter is optional.`

Options

• **instances**

  De-configures the Oracle Restart (SIHA) instance only.

• **software**

  Deletes the empty Oracle Restart (SIHA) home.

• **all**

  De-configures and Delete Oracle Restart (SIHA) home.

• **input_file**

  Property file that contains necessary parameters to perform de-configure and delete Oracle Restart. Use "emcli describe_input -oracle_restart" command to get the list of properties for performing cleanup operation.

• **schedule**

  – frequency: Frequency type with which the Oracle Restart is removed. It can be interval (in minutes), weekly, monthly, or yearly

  – start_time: Denotes the starting time of Cleanup Oracle Restart in the format yy-MM-dd HH:mm

  – end_time: Denotes the end time of Cleanup Oracle Restart in the format yy-Mm-dd HH:mm

  – repeat: Repetition rate at Cleanup Oracle Restart. If the frequency is interval, then repeat will be in minutes

  – months: Number of months after which repetition of Cleanup Oracle Restart will occur

  – days: Number of days after which repetition of Cleanup Oracle Restart will occur

  – tz: Time Zone ID

Examples

Example 1

This example cleans the Oracle Restart Instances based on property file "deletesiha.txt" with the specified schedule.

```
emcli delete_oracle_restart -instances
        -input_file="data:/tmp/deletesiha.txt"
        -schedule="frequency:interval;start_time:15-10-27 05:30;end_time:14-10-12
05:23;repeat:30;grace_period:60;tz:America/New_York"
```

Example 2

This example cleans the Oracle Restart based on property file "deletesiha.txt".

```
emcli delete_oracle_restart -all
        -input_file="data:/tmp/deletesiha.txt"
```

delete_paas_zone

Deletes a PaaS Infrastructure Zone. A PaaS Infrastructure Zone cannot be deleted if an existing software pool is associated with it.

Format

```
emcli delete_paas_zone
        -name="<name of PaaS Zone>
```

[ ] indicates that the parameter is optional.

Options

• **name**

  Name of the existing PaaS Infrastructure Zone.

Example

This example deletes the PaaS Infrastructure Zone with the name My PaaS Zone:

```
emcli delete_paas_zone
        -name="My PaaS Zone"
```

delete_patch_plans

Deletes patch plans.

Format

```
emcli delete_patch_plans
        -name="plan_names"
```

[ ] indicates that the parameter is optional

Options

name

Specifies the names of the patch plans that you want to delete. Use a comma as a separator if you want to specify multiple patch plans for this option.

Examples

The following example deletes the patch plans plan_1, plan_2, and plan_3:

```
emcli delete_patch_plans -name="plan_1,plan_2,plan_3"
```

delete_patches

Deletes patches from the software library.

Format

```
emcli delete_patches
      -patch_name=<patch_name>
      -release=<release_id>
      -platform=<platform_id>
```

Options

- **patch_name**

  Patch number.

- **release**

  Patch release ID.

- **platform**

  Patch platform ID.

Example

```
emcli delete_patches -patch_name=13741363 -release=80112310 -platform=226
```

See Also

create_patch_plan
get_connection_mode
get_patch_plan_data
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data
submit_patch_plan
upload_patches

Also see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*:

delete_pluggable_database

Deletes pluggable databases (PDBs).

Format

```
emcli delete_pluggable_database
     -cdbTargetName="CDB_of_target_PDBs"
     -cdbTargetType="CDB_target_type"
     -cdbHostCreds="CDB_host_credentials"
     -cdbTargetCreds="CDB_target_credentials"
     -pdbName="PDB_names"
     [-cdbHostPrivCreds="CDB_host_privileged_credentials"]
     [-ignoreStorageWarnings]
```

[ ] indicates that the parameter is optional.

Options

• **cdbTargetName**

Target container database (CDB) that contains the PDBs that you want to delete. Ensure that the target CDB you specify is a valid target in Enterprise Manager.

• **cdbTargetType**

Database type of the target CDB, which can be `oracle_database, rac_database,` and so on.

• **cdbHostCreds**

Credentials for the host on which the target CDB is located.

• **cdbTargetCreds**

Credentials for the target CDB.

• **pdbName**

Names of the PDBs that you want to delete. Ensure that you separate the names using a comma.

• **cdbHostPrivCreds**

Privileged credentials for the host on which the Snap Clone storage mount points are located. Note that this option is required only if you are deleting PDBs that were created using Snap Clone.

• **ignoreStorageWarnings**

Ignore any storage warnings that may be generated while deleting PDBs that were created using Snap Clone.

Examples

This example deletes the `test_pdb` PDB, which is a part of `test_CDB,` an Oracle single-instance CDB, using `HOST_CREDS` as the CDB host credentials and `DB_CREDS` as the CDB target credentials:

```
emcli delete_pluggable_database -cdbTargetName=test_database -
cdbTargetType=oracle_database -pdbName=test_pdb -cdbHostCreds=HOST_CREDS -
cdbTargetCreds=DB_CREDS
```

delete_pool verb

Deletes a software pool. A software Pool cannot be deleted if there is an existing service template associated with it.

Format

```
emcli delete_pool
        -name="<software pool name>"
        -target_type="<software pool target type>"
```

[ ] indicates that the parameter is optional.

Options

*   **name**

    The name of the existing software pool.

*   **target_type**

    The target type of the existing software pool.

Example

The following example deletes the software pool My Pool:

```
emcli delete_pool
        -name="My Pool"
        -target_type="mwaas_zone"
```

delete_privilege_delegation_settings

Deletes a privilege delegation setting template.

Format

```
emcli delete_privilege_delegation_settings
        -setting_names="setting_name1;setting_name2;setting_name3;"
```

Options

*   **setting_names**

    Name of the settings you want to delete.

Example

This example deletes the privilege settings for the names setting_name1, setting_name2, and setting_name3.

```
emcli delete_privilege_delegation_settings
        -setting_names="sudo_setting1;sudo_setting2;pbSetting1
```

delete_procedure

Deletes a deployment procedure. If the purge option is used, the procedure is deleted permanently. Otherwise, it is moved to the recycle bin.

Format

```
emcli delete_procedure
      [-procedure={procedure guid}]
      [-name={procedure name}]
      [-owner={procedure owner}]
      [-purge]
```

Parameters

- procedure

  The procedure GUID.

- name

  The procedure name.

- owner

  The procedure owner.

> **Note:**
>
> The name and owner parameters must be used together.

- purge

  Deletes the procedure permanently.

Example

The following example permanently deletes the procedure
16B15CB29C3F9E6CE040578C96093F61.

```
emcli delete_procedure        -procedure=16B15CB29C3F9E6CE040578C96093F61
      -purge
```

delete_proxy

Deletes a HTTP(S) proxy identified by the specified name.

Syntax

```
emcli delete_proxy
      -name="<name>"
```

Options

- -name

  Name identifying the proxy.

Example

The following command deletes a proxy with the "us-proxy-1" name.

```
emcli delete_proxy
      -name="us-proxy-1"
```

delete_resolution_state

Deletes an existing resolution state. You typically use this command for resolution states that are no longer used. You need to also specify an alternative resolution state in case there are any references to the state. In this case, the references are changed to this alternative state. This action might require some time.

Only a super administrator can execute this command. A success message is reported if the command is successful. An error message is reported if the deletion fails.

> **✎ Note:**
>
> No notifications are sent for any incidents or problems updated in this process.

Format

```
emcli delete_resolution_state
      -label="label of the state to be deleted"
      -alt_res_state_label="alternative resolution state"
```

Options

*   **label**

    Label of the state to be deleted.

*   **alt_res_state_label**

    Alternative state to be used.

Examples

This example deletes the resolution state "Waiting for SR" and replaces any references to this state with the state "Work in Progress".

```
emcli delete_resolution_state  -label="Waiting for SR" -
alt_res_state_label="Work in Progress"
```

delete_role

Deletes an existing Enterprise Manager administrator role.

Format

```
emcli delete_role
      -name="role_name"
```

Options

*   **name**

    Role name.

Examples

This example deletes the role name `existing_role`.

```
emcli delete_role -name="existing_role"
```

delete_saved_config

Deletes a saved target configuration.

Format

```
emcli delete_saved_config
      -name="saved config name"
      [-target_type="host"]
      [-target_name="test_host"]
```

```
[ ] indicates that the parameter is optional.
```

Parameters

- name

  The name of the saved configuration to delete.

- target_type

  The target type of the configuration to delete. The value should be the internal name. This is an optional parameter.

- target_name

  The name of the target of the configuration to delete. This is an optional parameter.

Examples

Example 1

The following command delete the saved configuration with the name "saved config name":

```
emcli delete_saved_config
      -name="saved config name"
```

Example 2

The following command deletes the saved configuration for host target type with the name "test_host":

```
emcli delete_saved_config
      -target_type="host"
      -target_name="test_host"
```

delete_saved_onetimecomparison

Deletes a saved target configuration.

Format

```
emcli delete_saved_onetimecomparison
      -name="saved config name"
      [-target_type="host"]
```

```
         [-target_name="test_host"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- name

  The name of the saved configuration to delete.

- target_type

  The target type of the configuration to delete. The value should be the internal name. This option is not mandatory.

- target_name

  The name of the target of the configuration to delete. This option is not mandatory.

Examples

Example 1

The following command delete the saved configuration with the name "saved config name":

```
emcli delete_saved_onetimecomparison
     -name="saved config name"
```

Example 2

The following command deletes the saved configuration for host target type with the name "test_host":

```
emcli delete_saved_onetimecomparison
     -target_type="host"
     -target_name="test_host"
```

delete_service_template

Deletes a service template.

Format

```
emcli delete_service_template
-name="<service template name>"]
-service_family="<service family name>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **name**

  Name of the existing service template.

- **service_family**

  Service family to which the service template belongs; for example, DBAAS for database and MWAAS for middleware.

Example

This example deletes the service template with name template2 and service family MWAAS:

```
emcli delete_service_template
      -name="Middleware service Template August"
      -service_family="MWAAS"
```

delete_siebel

Deletes one or more Siebel Enterprise instances and their associated targets, such as Siebel servers, component groups, components, work flows, and so on.

Format

```
emcli delete_siebel
      -enterprise=<Siebel_enterprise_1>,<Siebel_enterprise_2>
      [-out_file='<output_file>']
      [<-debug>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **enterprise**

  Target name of the Siebel enterprise as seen in the Enterprise Manager console. If multiple enterprises need to be deleted at the same time, provide a comma-separated (,) value.

- **out_file**

  Fully-qualified path of the output file. The output of the command is redirected to this file.

  If you include this option, the list of deleted targets are printed in the file. If you do not include this option, the list is printed on the console directly.

- **debug**

  Executes in verbose mode and generates debug log messages in the output.

Examples

This example deletes the Siebel Enterprise instances from Cloud Control. The output of the command is redirected to the deletion_output.txt file.

```
emcli delete_siebel
      -enterprise=SBA80_ent1.example.com,SBA78_ent2.us.example.com
      -out_file='c:\emcli\deletion_output.txt'
```

delete_siteguard_aux_host

Deletes an auxiliary host associated with the system.

Format

```
emcli delete_siteguard_aux_host
      -system_name="name_of_the_system"
      [-host_name="name_of_the_auxiliary_host"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **system_name**

  Name of the system whose auxiliary host you want to delete.

- **host_name**

  Name of the auxiliary host that you want to delete. If not specified, all auxiliary hosts associated with the system will be deleted.

Examples

Example 1

This example deletes the auxiliary host `host1.domain.com` associated with `austin-system`:

```
emcli delete_siteguard_aux_host
      -system_name="austin-system"
      -host_name="host1.domain.com"
```

Example 2

This example deletes the auxiliary host `host2.domain.com` associated with `austin-system`:

```
emcli delete_siteguard_aux_host
      -system_name="austin-system"
      -host_name="host2.domain.com"
```

delete_siteguard_configuration

Deletes the Site Guard configuration. The entire configuration (scripts, credential associations, site associations, operation plans) pertaining to the specified system and all the associated standby systems are deleted.

Format

```
emcli delete_siteguard_configuration
      -primary_system_name=<name> | -standby_system_name=<name>
```

Options

- **primary_system_name**

  Name of the primary system. Specify either primary_system_name or standby_system_name.

- **standby_system_name**

  Name of the standby system.

Examples

Example 1

```
emcli delete_siteguard_configuartion
      -primary_system_name="BISystem1"
```

Example 2

```
emcli delete_siteguard_configuration
      -standby_system_name="BISystem2"
```

See Also

delete_siteguard_credential_association

Deletes the credential association from the Site Guard configuration.

Format

```
emcli delete_siteguard_credential_association
      -system_name=<name>
      [-target_name=<name>]
      -credential_type=<type>
```

```
{ }  indicates that the parameter is optional
```

Options

* **system_name**

  Name of the system.

* **target_name**

  Name of the target.

* **credential_type**

  Type of the credential, which can be HostNormal, HostPrivileged, WLSAdmin, or DatabaseSysdba.

Examples

Example 1

```
emcli create_siteguard_credential_association
      -system_name="BISystem1"
      -credential_type="HostNormal"
      -credential_name="HOST-SGCRED"
      -credential_owner="sysman"
```

Example 2

```
emcli create_siteguard_credential_association
      -system_name="BIsystem1"
      -target_name="database-instance"
      -credential_type="HostNormal"
      -credential_name="HOST-DBCRED"
      -credential_owner="sysman"
```

See Also

ORACLE®

delete_siteguard_lag

Updates the limit for Apply lag and Transport lag for all or selected databases of the system.

Format

```
emcli delete_siteguard_lag
        [-system_name="name_of_the_system"]
        [-target_name="name_of_the_database"]
        [-property_name="lag_type"]

[ ]  indicates that the parameter is optional
```

Options

- **system_name**

  Name of the system whose lag limit property you want to update.

- **target_name**

  Name of the target database whose lag limit property you want to update.

- **property_name**

  Name of the lag property. Valid values for this option are `ApplyLag` and `TransportLag`.

Examples

Example 1

This example deletes the `ApplyLag` property on all of the databases configured on `austin-system`:

```
emcli delete_siteguard_lag
        -system_name="austin-system"
        -property_name="ApplyLag"
```

Example 2

The following example deletes the `TransportLag` property on the database `OID-db` configured on `austin-system`:

```
emcli delete_siteguard_lag
        -system_name="austin-system"
        -target_name="OID_db"
        -property_name="TransportLag"
```

delete_siteguard_script

Deletes the specified script from the Site Guard configuration.

Format

```
emcli delete_siteguard_script
      -script_id=<script_id>
```

Options

- **script_id**

  ID associated with the script.

Examples

```
emcli delete_siteguard_script
      -script_id="10"
```

See Also

get_siteguard_script_credential_params
create_siteguard_script
get_siteguard_script_hosts

delete_siteguard_script_hosts

Deletes the host or hosts associated with a given script.

Format

```
emcli delete_siteguard_script_hosts
      -script_id=<script_id>
      -host_name=<name1;name2;...>
```

Options

- **script_id**

  ID associated with the script.

- **host_name**

  Name of the host where this script will be run. You can specify this option more than once.

Examples

```
emcli delete_siteguard_script_hosts
      -script_id="10"
      -host_name="BIHOST1"
```

Output Columns

Step Number, Operation Name, Target Name, Target Host, and Error Mode

See Also

get_siteguard_script_credential_params
create_siteguard_script
get_siteguard_script_hosts

delete_sla

Deletes one or more SLAs for a target.

Format

```
emcli delete_sla
    -targetName=<target_name>
    -targetType=<target_type>
    -slaName=<SLA_name>
```

Options

- **targetName**

  Name of the target.

- **targetType**

  Type of target.

- **slaName**

  Name of the SLA.

Example

This example deletes the SLA with the name 'gold_sla' from the target.

```
emcli delete_sla
    -targetName='my_service' -targetType='generic_service'
    -slaName='gold_sla'
```

delete_swlib_entity

Deletes an entity from the Software Library

Format

```
emcli delete_swlib_entity -entity_rev_id="entity_rev_id"
```

Options

- **entity_rev_id**

  The identifier of the entity revision. The Software Library home page exposes the identifier for folders and entities as a custom column (Internal ID) and is hidden by default. This id can be obtained from the EMCLI verb list_swlib_entities with the -show_entity_rev_id flag for entities.

Examples

This example deletes the entity with the revision id
"`oracle:defaultService:em:provisioning:1:cmp:COMP_Component:SUB_Generic:B1B1880C6A8C62AAE040548C4D14:0.1`".

```
emcli delete_swlib_entity
    -
entity_rev_id="oracle:defaultService:em:provisioning:1:cmp:COMP_Component:SUB_Generic:B1B1880C6A8C62AAE040548C4D14:0.1"
```

delete_system

Deletes a system.

Format

```
emcli delete_system
      -name="name"
      [-type=<generic_system>]
```

`[ ]  indicates that the parameter is optional`

Options

- **name**

  Name of the system to delete.

- **type**

  System type: generic_system. Defaults to "generic_system".

Examples

This example deletes the system `my_system`.

```
emcli delete_system -name=my_system
```

delete_target

Deletes a specified target from the Enterprise Manager Cloud Control monitoring framework. Deleting a target removes it from the Management Repository and does not physically remove the target itself.

You can use the `get_targets` verb to obtain a list of available targets and their respective types.

Format

```
emcli delete_target
      -name=<name>
      -type=<type>
      [-delete_monitored_targets]
      [-async]
      [-delete_members]
```

`[ ]  indicates that the parameter is optional`

Options

- **name**

  Target name.

- **type**

  Target type.

- **delete_monitored_targets**

Deletes the targets monitored by the specified Management Agent. This is only applicable with the oracle_emd target type.

• **async**

Deletes the target asynchronously.

• **delete_members**

Deletes all the members of the target as well.

Examples

Example 1

This example deletes the Agent named test.example.com:1836 and all of its monitored targets. The Agent must be marked UNREACHABLE in Enterprise Manger Cloud Control to perform this operation.

```
emcli delete_target
     -name="test.example.com:1836"
     -type="oracle_emd"
     -delete_monitored_targets
     -async
```

Example 2

This example deletes the example_ias_farm target with the name "farm01_base_domain" and all of its members, such as domain, clusters, servers, application deployments, and so forth.

```
emcli delete_target
     -name="farm01_base_domain"
     -type="example_ias_farm"
     -delete_members
```

delete_tenant

Deletes an Enterprise Manager tenant.

Format

Standard Mode

```
emcli delete_tenant
     -name="tenant name"
```

Interactive Mode

```
emcli delete_tenant (name="tenant_name")
```

Options

• name

Identifies the name of the tenant to be deleted.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example deletes the `my_name` tenant.

```
emcli delete_tenant        -name="my_tenant"
```

delete_test

Deletes a Services test along with its constituent steps and step groups.

Format

```
emcli delete_test
      -name=<target_name>
      -type=<target_type>
      -testname=<test_name>
      -testtype=<test_type>

[ ]  indicates that the parameter is optional
```

Options

*   **name**

    Service target name.

*   **type**

    Service target type.

*   **testname**

    Name of the test.

*   **testtype**

    Type of test.

Example

This example deletes an HTTP test name MyTest for the generic_service target name MyTarget.

```
emcli delete_test -name='MyTarget' -type='generic_service'
                  -testname='MyTest' -testtype='HTTP'
```

delete_test_threshold

Deletes a test threshold.

Format

```
emcli delete_test_threshold
      -name=<target_name>
      -type=<target_type>
      -testname=<test_name>
      -testtype=<test_type>
      -metricName=<metric_name>
      -metricColumn=<metric_column>
      [-beaconName=<beacon_name>]
      [-stepName=<step_name>]
```

```
          [-stepGroupName=<stepgroup_name>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Service target name.

- **type**

  Service target type.

- **testname**

  Name of the test.

- **testtype**

  Type of test.

- **metricName**

  Name of the metric.

- **metricColumn**

  Name of the column.

- **beaconName**

  Name of the beacon.

- **stepName**

  Name of the step.

- **stepGroupName**

  Name of the step group.

Example

```
emcli delete_test_threshold
     -name="Service Name"
     -type="generic_service"
     -testname="Test Name"
     -testtype="HTTP"
     -metricName="http_response"
     -metricColumn="timing"
```

delete_PDB_Profiles

Self Service Application the administrator can use to delete an existing data profile
using emcli commands.

Format

```
emcli delete_dataprofile -profile_name="PDB_Profile"
        host_cred=credname:credowner
```

Options

- **profile_name**

The name of the profile to be deleted.

- **host_cred**

  Key value pair credentials for the host.

- **host_name**

  Host name where the profile backup exists.

Example

Format

```
emcli delete_dataprofile -profile_name=PDB_Profile
     -host_cred=CUSER:SYSMAN

emcli delete_dataprofile -profile_name="PDB_Profile"
      -host_target_name=den01nrr.us.oracle.com  -host_cred= CUSER:SYSMAN
```

delete_user

Deletes an existing Enterprise Manager administrator.

When a user is deleted, all jobs the user creates are stopped and deleted. Also, any blackouts the user creates are deleted. However, a user cannot be deleted if any blackouts the user creates are active at the time the call to delete the user is issued. This situation is considered an invalid state from which to delete a user. First, all of these active blackouts must be stopped, and a thwarted delete user call must be reissued.

Format

```
emcli delete_user
     -name=<user_name>
     [-new_object_owner=<user_name>]
     [-force]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Administrator name.

- **new_object_owner**

  Name of the administrator to assign the secure objects owned by the current administrator being deleted. If you do not specify this option, the secure objects are deleted that are owned by the administrator being deleted.

- **force**

  Deletes the administrator even if the administrator is currently logged in.

Examples

Example 1

This example deletes the Enterprise Manager administrator named `sysman3`.

```
emcli delete_user -name=sysman3
```

Example 2

This example deletes the Enterprise Manager administrator named user1, and assigns all the secure objects owned by user1 to user5.

delete_user_profile

Deletes a user profile.

Format

Standard Mode

```
emcli delete_user_profile
      -name="profile name"
```

Interactive or Script Mode

```
create_user_profile(
        name="profile name"
        )
```

Options

*   name

    The name of the user profile to be deleted.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example deletes the profile (`profile1`) of the current tenant.

```
emcli delete_user_profile
      -name="profile1"
```

deploy_bipublisher_reports

This verb deploys all of the Enterprise Manager Oracle-provided reports, or optionally, specific Enterprise Manager Plug-in reports to the BI Publisher catalog

You can also use this verb to upload a reports jar file (located on the OMS(s)'s file system. The operation does not overwrite existing BI Publisher Reports in the Enterprise Manager reports folder unless you specify the -force option.

> ✏️ **Note:**
>
> This verb requires Enterprise Manager Super Administrator privileges.

Format

```
emcli deploy_bipublisher_reports
    [-force]
```

```
        [-all | -reportsjarfile=<reports_jar_file> | (-pluginid=<plugin_id>
         [-pluginversion=<plugin_version>]) )
[ ]  indicates that the parameter is optional
```

Options

> **Note:**
>
> Using -force applies to the entire operation. The absence of all parameters assumes -all.
>
> You can specify the -all option, or -reportsjarfile option, or -pluginid option, but not all three at the same time. If you use the -pluginid option, you can also also include the -pluginversion option.

- **force**

  Overwrites reports. If you use this option, all reports on the BI Publisher server are overwritten with the new copies.

- **all**

  Overwrites reports. If you use this option, all reports on the BI Publisher server are overwritten with the new copies.

- **reportsjarfile**

  Deploys a single Enterprise Manager reports jar file that contains one or more BI Publisher Reports. This jar file is located relative to the OMS's $ORACLE_HOME.

- **pluginid**

  In addition to Enterprise Manager system reports, also deploys any subsequently loaded plug-in-based BI Publisher Reports.

- **pluginversion**

  Limits the plug-ins to a specific version.

Examples

Example 1

This example deploys all platform and plug-in Enterprise Manager Oracle-provided reports, but does not overwrite any existing reports

```
emcli deploy_bipublisher_reports -all
```

Example 2

This example deploys only the Chargeback and Trending reports, and overwrites any existing reports.

```
emcli deploy_bipublisher_reports -force -pluginid=oracle.sysman.emct -
pluginversion=12.1.0.3.0
```

deploy_bipublisher_selfupdates

Deploys Self Update Enterprise Manager reports to the BI Publisher catalog. To deploy all reports in a folder, specify the `-folder` option. To deploy a single report, use the `-folder` and `-report` options. To deploy all Self Update reports, use the `-all` option.

The deploy operation will not overwrite existing BI Publisher reports in the BI Publisher catalog unless the `-force` option is given.

Format

```
emcli deploy_bipublisher_selfupdates
    [-force]
    -all | (-folder=<folder> [-report=<reportname>])
```

`[ ]` indicates that the parameter is optional.

Options

- **force**

  Overwrites any existing reports in the BI Publisher catalog with the copy being deployed.

- **all**

  Deploys all Self Update reports.

- **folder**

  Limits the reports deployed to the specified folder. By default, all reports in the folder are deployed unless the `-report` option is also specified.

- **report**

  Deploys a single, specified report. The `-folder` parameter must be specified when using the -report parameter.

Examples

Example 1

The following example deploys all Self Update reports to the Compliance Reports folder:

```
emcli deploy_bipublisher_selfupdates -folder="Compliance Reports"
```

Example 2

The following example deploys only the Self Update report named Consolidation Report located in the Compliance Reports folder and overwrites any existing report.

```
 emcli deploy_bipublisher_selfupdates -force -folder="Compliance Reports" -
report="Consolidation Report"
```

deploy_plugin_on_agent

Deploys a plug-in on Management Agents. Agent names must be provided for plug-in deployment.

> **✏️ Note:**
>
> A plug-in can only be deployed on any Management Agent after it has been successfully deployed on the management server.

Format

```
emcli deploy_plugin_on_agent
        -agent_names=<agent1;agent2>
        -plugin=<plug-in_id[:version>]
        [-discovery_only]
```

```
[ ]  indicates that the parameter is optional
```

Options

* **agent_names**

  Management Agents (host:port) on which the plug-in needs to be deployed.

* **plugin**

  Plug-in ID and version that needs to be deployed. Version is optional, and it defaults to the latest applicable version deployed on the management server. If a later version is available but not certified on the Agent OS platform, the latest version is not picked up.

* **discovery_only**

  To be used when only discovery content needs to be deployed.

Examples

Example 1

This example deploys the latest version of oracle.sysman.db2 on Management Agent myhost1.example.com.

```
emcli deploy_plugin_on_agent -plugin="oracle.sysman.db2"
-agent_names="myhost1.example.com:1838"
```

Example 2

This example deploys version 12.1.0.1.0 of plug-in oracle.sysman.db2 on management agent myhost1.us.example.com.

```
emcli deploy_plugin_on_agent
      -plugin="oracle.sysman.db2:12.1.0.1.0"
      -agent_names="myhost1.us.example.com:1838"
```

deploy_plugin_on_server

Deploys a plug-in on the Management Servers. The deployment process for some plug-ins might restart the Management Servers. If the plug-in is already deployed on one of the servers, this server is skipped. If a lower version of the plug-in is already deployed, the plug-in is upgraded. If a lower revision of the plug-in is already deployed, the new revision is applied.

Format

```
emcli deploy_plugin_on_server
        -plugin=<plug-in_id>[:<version>]
        [-sys_password=<sys_password>]
        [-prereq_check]
        [-use_last_prereq_result]

[ ]  indicates that the parameter is optional
```

Options

*   **plugin**

    ID or ID:Version of the plug-in to be deployed on the Management Servers of the form -plugin=<oracle.sysman.db:12.1.0.1.0>, where the plug-in ID (like oracle.sysman.db) is a required parameter, and the version is optional. If do not specify a version, the highest version of the plug-in that has been downloaded is considered for deployment. If multiple revisions of this plug-in version are downloaded, the highest revision is considered for deployment.

*   **sys_password**

    Password of the repository DBA SYS. If you do not provide this, you are prompted for the password. This is not required if you use the prereq_check .

*   **prereq_check**

    If you provide this option, instead of deploying the plug-in, the verb displays only a check for all the unfulfilled prerequisites for this plug-in deployment to be successful. If you do not provide this option, plug-in deployment follows a prerequisities check.

*   **use_last_prereq_result**

    If prerequisites checks have been performed previously for a given set of plug-ins using the -prereq_check option and no other deployment activity occurred for these plug-ins, you can use this option to skip prerequisite checks and start the deployment immediately.

Examples

Example 1

This example deploys the latest downloaded version of Oracle Database plug-in (plug-in ID: oracle.sysman.db) on the management server.

```
emcli deploy_plugin_on_server
      -plugin=oracle.sysman.db
      -sys_password=<welcome>
```

Example 2

The folllowing example deploys the Oracle Database plug-in (with version 12.1.0.2..0) and Oracle Fusion Middleware plug-in (12.1.0.2.0) on the management server. Since sys password has not been passed on the command line, you are prompted for it. If a lower version of both plug-ins have already been deployed, they are upgraded to 12.1.0.2.0. If a lower version of only one of the plug-ins is deployed, this generates an error, and you will have to deploy them separately.

```
emcli deploy_plugin_on_server
      -plugin="oracle.sysman.db:12.1.0.2.0;oracle.sysman.emas:12.1.0.2.0"
```

deregister_forwarder_agents

Takes a list of agents and deregisters each agent as a forwarding agent.

Format

```
emcli deregister_forwarder_agents
      -agent_list="agent_list"
[ ]  indicates that the parameter is optional.
```

Options

• agent_list

List of agents that need to be deregistered as forwarders. The agents must be separated by space.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example deregisters agent1 and agent2 as forwarding agents.

```
emcli deregister_forwarder_agents
      -agent_list="agent1 agent2..."
```

describe_dbprofile_input

Lists and describes all database profile creation input variables.

Format

```
emcli describe_dbprofile_input      [-data_mode={EXPORT/DBCA_TEMPLATE/RMAN/
STORAGE_SNAPSHOT}]

[ ]  indicates that the parameter is optional.
```

Options

• data_mode

Data mode for which the database profile needs to be submitted.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example lists all input variables required for creating a snapshot database profile.

```
emcli describe_dbprofile_input -data_mode=STORAGE_SNAPSHOT
```

describe_engr_sys_targets

Lists all the components of an engineered system target discovered in Enterprise Manager.

Format

```
emcli describe_engr_sys_targets
    -system_target_name="system_target_name"
    -system_target_type="system_target_type"
    [-component_type="component_type"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- component_type

  Specifies the engineered system component target type.

Examples

Example 1

The following example displays all the components of the engineered system "DB Machine slcm12.example.com":"oracle_dbmachine":

```
emcli describe_engr_sys_targets
    -system_target_name="DB Machine slcm12.example.com"
    -system_target_type="oracle_dbmachine"
```

Example 2

The following example displays the Oracle Infiniband Switches targets of the engineered system "DB Machine slcm12.example.com":"oracle_dbmachine":

```
emcli describe_engr_sys_targets
    -system_target_name="DB Machine slcm12.example.com"
    -system_target_type="oracle_dbmachine"
    -component_type="Oracle Infiniband Switch"
```

describe_fmw_profile

Provides a description of the Fusion Middleware provisioning profile from the software library.

Format

```
emcli describe_fmw_profile
    -location="Profile Location"
```

Options

- location

  The complete software library path to the profile. Use the list_fmw_profiles verb to identify the complete path.

> **Note:**
>
> The name and owner parameters must be used together.

Example

The following example displays a description of the Fusion Middleware profile "MyProfile" from software library.

```
emcli describe_fmw_profile        -location="Fusion Middleware Provisioning/Profiles/
MyProfile"
```

describe_job

Describes a job and gets its properties for a job you have submitted from the user interface or using the create_job verb. The output can be redirected into a file and used as a template.

This verb support multi-task jobs.

Format

```
emcli describe_job
     -name=<"job_name">
     [-owner=<"job_owner">]
     [-verbose]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the job to describe.

- **owner**

  Enterprise Manager administrator who owns this job. If not provided, the current EM CLI logged-in administrator is assumed as the owner. The logged-in Enterprise Manager administrator must have at least the view privilege to describe a job.

- **verbose**

  Outputs a help template along with the properties.

Examples

Example1

This example describes the library job "yourJob" owned by the Enterprise Manager administrator "admin1". The logged-in Enterprise Manager administrator has view privilege on this job.

```
emcli describe_job -name=yourJob -owner=admin1
```

Example 2

```
emcli describe_job -name=EMCLI_JOB_2 -verbose

# Job Name : EMCLI_JOB_2

# Current status of the job is ACTIVE.

# Job Type: OSCommand.
# This job type supports the following target types only :
host,j2ee_application,metadata_repository,oracle_apache,oracle_apm,oracle_beacon,
oracle_csa_collector,oracle_database,oracle_emd,oracle_emrep,oracle_home,oracle_i
as_farm,oracle_oms,oracle_oms_console,oracle_oms_pbs,weblogic_domain,weblogic_j2e
eserver.


# Target List.
# In a target list, each member is specified using the target name and target
type # in the fashion:
#     target_name:target_type
# To specify an element of the target list, the following notation is used:
#     job_target_list.1=target_name:target_type
# The suffix "1" after the key word "job_target_list" signifies that the entry
is # for the first element.
# The target target_name:target_type should exists in EM.
# Permissible target types are:
host,j2ee_application,metadata_repository,oracle_apache,oracle_apm,oracle_beacon,
oracle_csa_collector,oracle_database,oracle_emd,oracle_emrep,oracle_home,oracle_i
as_farm,oracle_oms,oracle_oms_console,oracle_oms_pbs,weblogic_domain,weblogic_j2e
eserver.
# A sample target list could be:
# job_target_list.1=<target_name>:host
# job_target_list.2=<target_name>:host
# The target list can only contain targets of the same target type. A cluster,
# group, domain or system
# target must not be intermixed with targets of the other target types.


# Variable List.
# In a variable list, each member is specified in the following way:
# Scalar variable: A variable whose value can be represented as a single string.
#     variable.variable_name=variable_value
# Here "variable" is a keyword. Variable name is the name of the variable whose
# value is being specified.
# Value is specified on the right hand side after the equal to sign.
# Vector variable: A variable whose value is represented as an array or list of
# string values.
#     variable.variable_name.1=value1
#     variable.variable_name.2=value2
# Here the numbers suffixing the variable name signify the entry number in the
# list.
# Large variable: A variable whose value is exceptionally large. Syntax is
similar # to a scalar variable.
#     variable.large_variable_name=a_very_very_big_value
```

```
# Credential List.
# This is the list of credential usages declared by the job type.
# Each entry takes the form:
#    cred.credusage_name.target_details=cred_type:cred_details
# Here the prefix "cred" is a keyword signifying that this line represents a
# credential entry.
# "credusage_name" would be substituted with the name of the credential usage
# declared in the job type.
# This is followed by the target details, which take the following form:
#    target_name:target_type
# The value for this credential usage entry is specified using the type of the
# credential and its details.
# "cred_type" can take either "SET" or "NAMED" as its value, depending on whether #
the credential is a credential set or a named credential.
# "cred_details" can specify either the name of a credential set or the name of a #
named credential based on the "cred_type"
# A sample entry for a target target1:host for credential usage defaultHostCred
# for a credential set could look like:
#    cred.defaultHostCred.target1:host=SET:HostCredsNormal
# A sample entry for a target target1:host for credential usage defaultHostCred
# for a named credential could look like:
#    cred.defaultHostCred.target1:host=NAMED:MyNamedCredential
# A sample entry for a target target1:host for credential usage defaultHostCred
# for a named credential shared by EM Admin "admin1" could look like:
#    cred.defaultHostCred.target1:host=NAMED:admin1:MyNamedCredential

# Schedule.
# Specify a schedule for the job. Detailed instructions as per below:
# Frequency: Specifies the frequency of repeatedly submitting instances of this
# job.
#    scheule.frequency=Frequency_Type
# Frequency type could be either of IMMEDIATE, ONCE, WEEKLY, MONTHLY, YEARLY,
# REPEAT_BY_MINUTES, REPEAT_BY_HOURS, REPEAT_BY_DAYS, REPEAT_BY_WEEKS.
# If frequency is IMMEDIATE, then other schedule fields do not matter.
# Start Time: Start time for the schedule.
#    scheule.startTime=MM-DD-YYYY
# End Time: End time for the schedule.
#    scheule.endTime=MM-DD-YYYY
# Grace Period: Grace period in minutes for the schedule.
#    scheule.graceperiod=
# Months : Months for repetition. January is denoted by 0 and December by 11
#    schedule.months=0,1,2
# Days: Days of the week for repetition. Sunday is denoted by 0 and Saturday by 6.
#    schedule.days=0,1,2
# Timezone: Timezone information is further detailed into type, target index, zone #
offset and region.
#    schedule.timezone.type: either of TIMEZONE_TARGET, TIMEZONE_SPECIFIED,
# TIMEZONE_REGION_SPECIFIED.
#    schedule.timezone.targetIndex : specify the index of the target whose
# timezone is to be used.
#    schedule.timezone.zoneOffset : timezone offset.
#    schedule.timezone.region : timezone region
# Following is a complete schedule section, remove #  and populate the values for #
submission:
# scheule.frequency=ONCE
# schedule.startTime=12-21-2012
# schedule.endTime=12-21-2012
# schedule.gracePeriod=10
# schedule.months=
# schedule.days=
```

```
# schedule.timezone.type=TIMEZONE_TARGET
# schedule.timezone.targetIndex=1
# schedule.timezone.zoneOffset=
# schedule.timezone.region=

job_target_list.1=myhost.us.example.com:host

# Variable: args
# Description: Options of the command to run on the target
variable.args=hello

# Variable: command
# Description: Command to run on the target
variable.command=echo

# Credential Usage: defaultHostCred
# Description:
cred.defaultHostCred.myhost.us.example.com:host=NAMED:SYSMAN:CRED1

schedule.frequency=REPEAT_BY_MINUTES
schedule.startTime=2012-02-01 01:01:01.0
schedule.endTime=2051-02-01 01:01:01.0
schedule.gracePeriod=-1
schedule.months=
schedule.days=
schedule.interval=1
schedule.timezone.type=TIMEZONE_TARGET
schedule.timezone.targetIndex=1
schedule.timezone.zoneOffset=0
schedule.timezone.region=
```

describe_job_type

Describes the job type and gets its properties. The output can be redirected into a file.

This verb dumps out a properties file for a job type that supports the Job System Generic EM CLI. This file contains some documentation, a list of all required credential usages, and a list of all variables required to create a (library) job instance of the job type.

This verb support multi-task jobs.

Format

```
emcli describe_job_type
     -job_type=<"job_type_internal_name">
     [-verbose]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **job_type**

Specify the name of the job type to describe. You can use the get_job_types verb to obtain the names of all job types for which a job or library jobs can be created using EM CLI.

• **verbose**

Outputs a help template along with the properties.

Examples

Example 1

This example produces a property file on the console, which can be redirected to a file and used multiple times.

```
emcli describe_job_type -job_type=OSCommand

# Job Type: OSCommand.
# This job type supports the following target types only :
host,j2ee_application,metadata_repository,oracle_apache,oracle_apm,oracle_beacon,oracle
_csa_collector,oracle_database,oracle_emd,oracle_emrep,oracle_home,oracle_ias_farm,orac
le_oms,oracle_oms_console,oracle_oms_pbs,weblogic_domain,weblogic_j2eeserver.


# Variable: args
# Description: Options of the command to run on the target
variable.args=

# Variable: command
# Description: Command to run on the target
variable.command=

# Credential Usage: defaultHostCred
# Description:
cred.defaultHostCred.<target_name>:<target_type>=
```

Example 2

This example with the verbose option generates a property dump with help on how to specify each individual property for the job.

```
emcli describe_job_type -job_type=OSCommand -verbose

# Job Type: OSCommand.
# This job type supports the following target types only :
host,j2ee_application,metadata_repository,oracle_apache,oracle_apm,oracle_beacon,oracle
_csa_collector,oracle_database,oracle_emd,oracle_emrep,oracle_home,oracle_ias_farm,orac
le_oms,oracle_oms_console,oracle_oms_pbs,weblogic_domain,weblogic_j2eeserver.


# Target List.
# In a target list, each member is specified using the target name and target type #
in the fashion:
#    target_name:target_type
# To specify an element of the target list, the following notation is used:
#    job_target_list.1=target_name:target_type
# The suffix "1" after the key word "job_target_list" signifies that the entry is #
for the first element.
# The target target_name:target_type should exists in EM.
# Permissible target types are:
host,j2ee_application,metadata_repository,oracle_apache,oracle_apm,oracle_beacon,oracle
_csa_collector,oracle_database,oracle_emd,oracle_emrep,oracle_home,oracle_ias_farm,orac
le_oms,oracle_oms_console,oracle_oms_pbs,weblogic_domain,weblogic_j2eeserver.
# A sample target list could be:
# job_target_list.1=<target_name>:host
# job_target_list.2=<target_name>:host
# The target list can only contain targets of the same target type. A cluster,
# group, domain or system
# target must not be intermixed with targets of the other target types.
```

```
# Variable List.
# In a variable list, each member is specified in the following way:
# Scalar variable: A variable whose value can be represented as a single string.
#    variable.variable_name=variable_value
# Here "variable" is a keyword. Variable name is the name of the variable whose
# value is being specified.
# Value is specified on the right hand side after the equal to sign.
# Vector variable: A variable whose value is represented as an array or list of
# string values.
#    variable.variable_name.1=value1
#    variable.variable_name.2=value2
# Here the numbers suffixing the variable name signify the entry number in the
# list.
# Large variable: A variable whose value is exceptionally large. Syntax is
similar # to a scalar variable.
#    variable.large_variable_name=a_very_very_big_value

# Credential List.
# This is the list of credential usages declared by the job type.
# Each entry takes the form:
#    cred.credusage_name.target_details=cred_type:cred_details
# Here the prefix "cred" is a keyword signifying that this line represents a
# credential entry.
# "credusage_name" would be substituted with the name of the credential usage
# declared in the job type.
# This is followed by the target details, which take the following form:
#    target_name:target_type
# The value for this credential usage entry is specified using the type of the
# credential and its details.
# "cred_type" can take either "SET" or "NAMED" as its value, depending on
whether # the credential is a credential set or a named credential.
# "cred_details" can specify either the name of a credential set or the name of
a # named credential based on the "cred_type"
# A sample entry for a target target1:host for credential usage defaultHostCred
# for a credential set could look like:
#    cred.defaultHostCred.target1:host=SET:HostCredsNormal
# A sample entry for a target target1:host for credential usage defaultHostCred
# for a named credential could look like:
#    cred.defaultHostCred.target1:host=NAMED:MyNamedCredential
# A sample entry for a target target1:host for credential usage defaultHostCred
# for a named credential shared by EM Admin "admin1" could look like:
#    cred.defaultHostCred.target1:host=NAMED:admin1:MyNamedCredential

# Schedule.
# Specify a schedule for the job. Detailed instructions as per below:
# Frequency: Specifies the frequency of repeatedly submitting instances of this
# job.
#    scheule.frequency=Frequency_Type
# Frequency type could be either of IMMEDIATE, ONCE, WEEKLY, MONTHLY, YEARLY,
# REPEAT_BY_MINUTES, REPEAT_BY_HOURS, REPEAT_BY_DAYS, REPEAT_BY_WEEKS.
# If frequency is IMMEDIATE, then other schedule fields do not matter.
# Start Time: Start time for the schedule.
#    scheule.startTime=MM-DD-YYYY
# End Time: End time for the schedule.
#    scheule.endTime=MM-DD-YYYY
# Grace Period: Grace period in minutes for the schedule.
#    scheule.graceperiod=
# Months : Months for repetition. January is denoted by 0 and December by 11
#    schedule.months=0,1,2
# Days: Days of the week for repetition. Sunday is denoted by 0 and Saturday by
```

```
6.
#     schedule.days=0,1,2
# Timezone: Timezone information is further detailed into type, target index, zone #
offset and region.
#     schedule.timezone.type: either of TIMEZONE_TARGET, TIMEZONE_SPECIFIED,
# TIMEZONE_REGION_SPECIFIED.
#     schedule.timezone.targetIndex : specify the index of the target whose
# timezone is to be used.
#     schedule.timezone.zoneOffset : timezone offset.
#     schedule.timezone.region : timezone region
# Following is a complete schedule section, remove #  and populate the values for #
submission:
# scheule.frequency=ONCE
# schedule.startTime=12-21-2012
# schedule.endTime=12-21-2012
# schedule.gracePeriod=10
# schedule.months=
# schedule.days=
# schedule.timezone.type=TIMEZONE_TARGET
# schedule.timezone.targetIndex=1
# schedule.timezone.zoneOffset=
# schedule.timezone.region=


# Variable: args
# Description: Options of the command to run on the target
variable.args=

# Variable: command
# Description: Command to run on the target
variable.command=

# Credential Usage: defaultHostCred
# Description:
cred.defaultHostCred.<target_name>:<target_type>=
```

describe_library_job

Describes a library job and gets its properties. The output can be redirected into a file.

Format

```
emcli describe_library_job
    -name=<"job_name">
    [-owner=<"job_owner">]
    [-verbose]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the library job to describe.

- **owner**

  Enterprise Manager administrator who owns this library job. If not provided, the current EM CLI logged-in administrator is assumed as the owner. The logged-in Enterprise Manager administrator must have at least the view privilege to describe a job.

- **verbose**

  Outputs a help template along with the properties.

Examples

Example 1

This example describes the library job "yourLibJob" owned by the Enterprise Manager administrator "admin1". The logged-in Enterprise Manager administrator has view privilege on this library job.

```
emcli describe_library_job -name=yourLibJob -owner=admin1
```

Example 2

```
emcli describe_library_job -name=MYJOB1

# Job Name : MYJOB1

# Current status of the job is ACTIVE.

# Job Type: OSCommand.
# This job type supports the following target types only :
host,j2ee_application,metadata_repository,oracle_apache,oracle_apm,oracle_beacon,
oracle_csa_collector,oracle_database,oracle_emd,oracle_emrep,oracle_home,oracle_i
as_farm,oracle_oms,oracle_oms_console,oracle_oms_pbs,weblogic_domain,weblogic_j2e
eserver.


job_target_list.1=myhost.us.example.com:host

# Variable: args
# Description: Options of the command to run on the target
variable.args=hello

# Variable: command
# Description: Command to run on the target
variable.command=echo

# Credential Usage: defaultHostCred
# Description:
cred.defaultHostCred.myhost.us.example.com:host=NAMED:SYSMAN:CRED1

schedule.frequency=REPEAT_BY_MINUTES
schedule.startTime=2012-02-01 01:01:01.0
schedule.endTime=2051-02-01 01:01:01.0
schedule.gracePeriod=-1
schedule.months=
schedule.days=
schedule.interval=1
schedule.timezone.type=TIMEZONE_TARGET
schedule.timezone.targetIndex=1
schedule.timezone.zoneOffset=0
schedule.timezone.region=
```

describe_patch_plan_input

Describes the input data of a patch plan.

Format

```
emcli describe_patch_plan_input
       -name=<name>
```

Options

*   **name**

    Name of a given patch plan.

Example

```
emcli describe_patch_plan_input -name="plan_name
```

See Also

create_patch_plan
delete_patches
get_connection_mode
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data
show_patch_plan
submit_patch_plan
upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

describe_procedure_input

Describes the input data of a deployment procedure or a procedure configuration.

Format

```
emcli describe_procedure_input
      [-procedure=<procedure_GUID>]
      [-name=<procedure_name_or_procedure_conf>]
      [-owner=<procedure_owner_or_procedure_config>]
      [-parent_proc=<procedure_of_procedure_config>]

[ ]  indicates that the parameter is optional
```

Options

*   **procedure**

    GUID of the procedure to execute.

- **name**

  Name of the procedure or procedure configuration.

- **owner**

  Owner of the procedure or procedure configuration.

- **parent_proc**

  Procedure of the procedure configuration. This applies to describe a procedure configuration when both a procedure and a procedure configuration have the same name.

Examples

```
emcli describe_procedure_input -procedure=16B15CB29C3F9E6CE040578C96093F61 >
describeDP.properties
```

dg_change_protection_mode

Changes the protection mode of a Data Guard configuration associated with the specified primary database.

Format

```
emcli dg_change_protection_mode
      -new_protection_mode="maximum_protection|maximum_availability|
maximum_performance"
      -primary_target_name="<primary target name>"
      -primary_target_type="oracle_database|rac_database"
      -standby_target_name="<standby target name>"
      -standby_target_type="oracle_database|rac_database"
      [-primary_db_creds_name="<primary database credential name>"]
      [-primary_host_creds_name="<primary database host credential name>"]
      [-standby_db_creds_name="<standby database credential name>"]
      [-standby_host_creds_name="<standby database host credential name>"]
      [-tde_wallet_creds_name="<transparent data encryption wallet credentials
of the primary database>"]
```

Options

- new_protection_mode

  Type of the new protection mode. Allowed values:

  – maximum_protection: Maximum Protection

  – maximum_availability: Maximum Availability

  – maximum_performance: Maximum Performance

- primary_target_name

  Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- primary_target_type

  Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- standby_target_name

Standby database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- standby_target_type

  Standby database target type. Specify oracle_database for single instance, or rac_database for cluster.

- primary_db_cred_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- primary_host_cred_name

  Primary database host named credential for an operating system user who can access the primary database Oracle home. Default is to use preferred credential.

- standby_db_cred_name

  Named credential of a standby database for a user with a SYSDBA role or a SYSDG role.

  Default value: Preferred credentials will be used.

- standby_host_cred_name

  Standby database host named credential for an operating system user who can access the standby database Oracle home. Default is to use preferred credential.

- tde_wallet_creds_name

  Transparent Data Encryption wallet credentials for the primary database. Use create_named_credential verb to create these credentials as shown below:

```
emcli create_named_credential
    -cred_name=WC1 -cred_type=GenericPassword
    -auth_target_type='<system>'
    -attributes="GENERIC_PASSWORD:<Primary Database TDE Wallet Password"
```

Example

The following example changes the Data Guard protection mode between the primary single-instance database named 'database' and the standby single-instance database named 'database1' to Maximum Performance mode.

```
emcli dg_change_protection_mode
 -new_protection_mode="max_performance"
 -primary_target_name="database"
 -primary_target_type="oracle_database"
 -standby_target_name="database1"
 -standby_target_type="oracle_database"
```

dg_configure_fsfo

Configures Data Guard fast-start failover between the specified primary and standby databases.

> **📝 Note:**
>
> The Enterprise Manager alternate observer feature is not supported for database version 12.2 and higher. Instead, the Data Guard broker multiple observer capability should be used. If an alternate observer was configured for these versions, it should be removed using the following EMCLI command:
>
> ```
> "emcli dg_configure_observers -delete_alternate_observer "
> ```

Format

```
emcli dg_configure_fsfo
      -operation= "enable | disable | edit"
      -primary_target_name=""
      -primary_target_type="oracle_database|rac_database"
      -standby_target_name=""
      -standby_target_type="oracle_database|rac_database"
      [-observer_host_name=""]
      [-observer_oracle_home=""]
      [-observer_host_cred_name=""]
      [-alternate_observer_host_name=""]
      [-alternate_observer_oracle_home=""]
      [-alternate_observer_host_cred_name=""]
      [-failover_threshold=""]
      [-lag_limit=""]
      [-failover_conditions=""]
      [-auto_reinstate_primary="Yes|No"]
      [-auto_shutdown_primary="Yes|No"]
      [-primary_db_creds_name=""]
      [-primary_host_creds_name=""]
      [-standby_db_creds_name=""]
      [-standby_host_creds_name=""]
      [-primary_gi_host_creds_name=""]
      [-standby_gi_host_creds_name=""]
      (
          [-enable_flashback_database="Yes|No"]
          [-primary_fast_recovery_area="<primary database fast recovery area>"]
          [-standby_fast_recovery_area="<standby database fast recovery area>"]
      )
[ ]  indicates that the parameter is optional.
```

Options

- `operation`: Operation to be performed on the fast-start failover, as follows:

  – `enable`: enable fast-start failover.

  – `disable`: disable fast-start failover and stop the observer.

  – `edit`: Make changes to the fast-start failover configuration currently in place for the specified primary database. You can change the fast-start failover designated standby database, adjust the thresholds or configure a new observer by specifying a new observer host. If you change the observer host and/or oracle home location, the current observer will be stopped and a new one started with the new location. However, if the observer host or Oracle home location is not changed, there will be no change to the state of the already configured observer.

- `primary_target_name`: Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- `primary_target_type`: Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- `standby_target_name`: Standby database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- `standby_target_type`: Standby database target type. Specify oracle_database for single instance, or rac_database for cluster.

- `primary_db_cred_name`: Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- `primary_host_cred_name`: Primary database host named credential for an operating system user who can access the primary database Oracle home. Default is to use preferred credential.

- `standby_db_cred_name`: Standby database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- `standby_host_cred_name`: Standby database host named credential for an operating system user who can access the standby database Oracle home. Default is to use preferred credential.

- `observer_host_name`: Discovered host where Enterprise Manager will start the observer. Default is none.

- `observer_oracle_home`: Oracle home on the observer host, if specified Default is none.

- `observer_host_cred_name`: Observer host named credential for an operating system user who can access the observer Oracle home.

- `alternate_observer_host_name`: Discovered host where Enterprise Manager will start the alternate observer if the main observer becomes inaccessible and cannot be restarted. Default is none.

- `alternate_observer_oracle_home`: Oracle home on the alternate observer host, if specified, where Enterprise Manager will start the alternate observer if necessary. Default is none.

- `alternate_observer_host_cred_name`: Alternate observer host named credential for an operating system user who can access the alternate observer Oracle home.

- `failover_threshold`: Amount of time in seconds the primary database must be out of contact with the observer and the standby database before a fast-start failover is initiated. Default is 30 seconds.

- `lag_limit`: Amount of time in seconds the standby database is allowed to fall behind the primary database, beyond which a fast-start failover will not be allowed. Default is 30 seconds.

- `failover_conditions`: A comma separated list of failover conditions, that if detected on the primary database, trigger a fast-start failover.

  - 1: Corrupted Controlfile
  - 2: Corrupted Dictionary
  - 3: Inaccessible Logfile
  - 4: Stuck Archiver

–   5: Datafile Write Errors

> **Note:**
>
> Default is 1, 2, 5

- `auto_reinstate_primary`: Controls whether the observer will automatically reinstate the former primary database once contact is re-established after the former primary database is restarted. Does not control reinstate behavior for failovers caused by an error condition. Default is yes.

- `auto_shutdown_primary`: Controls whether the primary database will shut itself down if it independently discerns that a fast-start failover may have occurred, but cannot verify it due to network isolation from the observer and the standby database. Does not control shutdown behavior for failovers caused by an error condition. Default is yes.

- `primary_gi_host_creds_name`: Grid Infrastructure named credentials for an operating system user who can access the grid infrastructure Oracle home of the Primary Database.

- `standby_gi_host_creds_name`: Grid Infrastructure named credentials for an operating system user who can access the grid infrastructure Oracle home of the Standby Database.

- `enable_flashback_database`: Flashback logging will be enabled on the primary and standby databases. Flashback database is required to reinstate the old primary database after a failover.
  Enabling flashback for database versions 12.2 and above while configuring fast-start failover is optional.

  This option is supported only for "enable"|"edit" operation.

  Default Value: "Yes" for database versions lower than 12.2 and "No" for database versions 12.2 and greater.

  Allowed Value: Yes or No

- `primary_database_fast_recovery_area`: Specifies the default storage area where archived redo log files (and other recovery-related files) reside on the primary database.
  This input will be considered only when the fast recovery area is not already set and when flashback must be enabled.

  This option is supported only for "enable"|"edit" operation.

  Default Value: <oracle_base>/fast_recovery_area

  Default Value of fast recovery area size will be set to twice the database size.

- `standby_database_fast_recovery_area`: Specifies the default storage area where archived redo log files (and other recovery-related files) reside on the standby database.
  This input will be considered only when the fast recovery area is not already set and when flashback must be enabled.

  This option is supported only for "enable"|"edit" operation.

  Default Value: <oracle_base>/fast_recovery_area

  Default Value of fast recovery area size will be set to twice the database size.

Examples

Example 1

The following example enables a fast-start failover where the primary target is a single-instance database named "dbprimary" and the standby target is a single-instance database named "dbstandby", with no automatic reinstatement of the former primary database when contact is re-established.

```
emcli dg_configure_fsfo
     -operation="enable"
     -primary_target_name="dbprimary"
     -primary_target_type="oracle_database"
     -standby_target_name="dbstandby"
     -standby_target_type="oracle_database"
     -auto_reinstate_primary="no"
```

Example 2

The following example disables a fast-start failover for the primary single-instance database named "database" and the standby single-instance database named "database1".

```
emcli dg_configure_fsfo
     -operation="disable"
     -primary_target_name="database"
     -primary_target_type="oracle_database"
     -standby_target_name="database1"
     -standby_target_type="oracle_database"
```

Example 3

The following example edits a fast-start failover for the primary target single-instance database named "dbprimary", changing the standby target to a single-instance database named "dbalternate". It also changes the lag limit to 60 seconds and the failover threshold to 40 seconds.

```
emcli dg_configure_fsfo
     -operation="edit"
     -primary_target_name="dbprimary"
     -primary_target_type="oracle_database"
     -standby_target_name="dbalternate"
     -standby_target_type="oracle_database"
     -lag_limit="60"
     -failover_threshold="40"
```

Example 4

The following example enables the fast-start-failover for the primary single-instance database named "database" and the standby single-instance database named "database1". It also creates the observer on the specified host and Oracle home. It also specifies that the primary should not be automatically reinstated and shutdown."

```
emcli dg_configure_fsfo
     -operation="enable"
     -primary_target_name="database"
     -primary_target_type="oracle_database"
     -standby_target_name="database1"
     -standby_target_type="oracle_database"
     -observer_host_name="observer_host_name"
     -observer_oracle_home="observer_oracle_home"
```

ORACLE®

```
                    -observer_host_cred_name="observer_host_cred_name"
                    -auto_reinstate_primary="No"
                    -auto_shutdown_primary="No"
                    -failover_conditions="1,2,5"
                    -failover_threshold="90"
```

dg_configure_observers

This verb is applicable to database version 12.2 and higher. There are multiple subcommands available for this verb:

dg_configure_observers -start

Starts one or more fast-start failover observers on the specified hosts and Oracle homes for the Data Guard configuration associated with the specified primary database.

Format

```
emcli dg_configure_observers -start
        -primary_target_name="<primary target name>"
        -primary_target_type="oracle_database|rac_database"
        -observer_input_file="<full pathname of input file>"
        [-primary_db_creds_name="<primary database credential name>"]
        [-primary_host_creds_name="<primary database host credential name>"]

[ ]  indicates that the parameter is optional.
```

Options

*   primary_target_name

    Primary database Enterprise Manager target name. This can be either a single-instance database or a cluster database.

*   primary_target_type

    Primary database target type. The allowed values are:

    –   oracle_database: Single instance database

    –   rac_database: Cluster database

*   observer_input_file

    The name of a file containing the information to start the respective observers. The format of this file as follows:

```
*observer.<i>.observer_name=<#<i>observer name>
*observer.<i>.observer_host=<Host name where #<i> observer will be started.>
*observer.<i>.observer_orale_home=<Oracle home location on the host where
#<i> observer will be started.>
*observer.<i>.observer_host_cred_name=<host named credential for user who
can access the #<i> observer Oracle home>
*observer.<i>.observer_host_wallet_cred_name=<named credential of the Oracle
wallet configured in the #<i> observer Oracle home>
observer.<i>.is_Master=<Y||N>
observer.<i>.observer_runtime_data_file=<Runtime data file path and name for
the #<i> observer. If not specified, the file fsfo.dat will be created under
the oracle base.>
observer.<i>.observer_log_file=<The full path of the #<i> observer log file.>
```

```
Where
<i>:(0,1,2)
(*)- Mandatory parameters.
```

- primary_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Preferred credentials are used as the default values.

- primary_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle home. Preferred credentials are used as the default values.

Example 1

The following example will start the observers 'oemcli1' and 'oemcli2' for the Data Guard configuration associated with primary database 'database'. This will also delete the Enterprise Manager alternate observer and its associated corrective action job for the Data Guard configuration associated with primary database 'database'.

```
emcli dg_configure_observers
    -start -delete_alternate_observer
    -primary_target_name=database
    -primary_target_type=oracle_database
    -primary_db_creds_name=DB_CRED
    -primary_host_creds_name=HOST_CRED
    -observer_input_file=/scratch/startObs.props
```

Example 2

The following example will start the observers provided in observer_input_file '/scratch/startObs.props' for the Data Guard configuration associated with primary database 'database'.

```
emcli dg_configure_observers
    -start -primary_target_name=database
    -primary_target_type=oracle_database
    -primary_db_creds_name=DB_CRED
    -primary_host_creds_name=HOST_CRED
    -observer_input_file=/scratch/startObs.props

Following is the content of /scratch/startObs.props:
observer.0.observer_name=oemcli1
observer.0.observer_host=host1.us.oracle.com
observer.0.observer_oracle_home=/scratch/oracle_base1/product/12.2.0/dbhome_1
observer.0.observer_host_cred_name=HOST1_CRED
observer.0.observer_host_wallet_cred_name=WC2
observer.0.observer_runtime_data_file=/scratch/oracle_base1/oemcli1/
obs_omemcli1_122.dat
observer.0.observer_log_file=/scratch/oracle_base1/oemcli1/oemcli1_122.log
observer.0.is_Master=Y
observer.1.observer_name=oemcli2
observer.1.observer_host=host2.us.oracle.com
observer.1.observer_oracle_home=/scratch/oracle_base1/product/12.2.0/dbhome_1
observer.1.observer_host_cred_name=HOST2_CRED
observer.1.observer_host_wallet_cred_name=WC2
```

Use the create_named_credential verb to create observer_host_wallet_cred_name credential as shown below:

```
emcli create_named_credential
    -cred_name=WC2 -cred_type=GenericPassword
```

```
-auth_target_type='<system>'
-attributes="GENERIC_PASSWORD:<Oracle Wallet Password>"
```

dg_configure_observers -stop

Stops the respective observer/observers for the Data Guard configuration associated with the specified primary database.

Format

```
emcli dg_configure_observers -stop
    -primary_target_name="<primary target name>"
    -primary_target_type="oracle_database|rac_database"
    [-observer_input_file="<full pathname of input file>"]
    [-stop_all]
    [-primary_db_creds_name="<primary database credential name>"]
    [-primary_host_creds_name="<primary database host credential name>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   primary_target_name

    Primary database Enterprise Manager target name. This can be either a single-instance database or a cluster database.

*   primary_target_type

    Primary database target type. The allowed values are:

    – oracle_database: Single instance database

    – rac_database: Cluster database

*   observer_input_file

    The name of a file containing the information to stop the respective observers. The format of this file is as follows:

    ```
    observer.0.observer_name=<#1 observer name>
    observer.1.observer_name=<#2 observer name>
    observer.2.observer_name=<#3 observer name>
    ```

*   stop_all

    Stops all the observers. This option cannot be used with -observer_input_file.

*   primary_db_creds_name

    Primary database named credential for a user with SYSDBA or SYSDG role. Preferred credentials are used as the default values.

*   primary_host_creds_name

    Primary database host named credential for an operating system user who can access the primary database Oracle home. Preferred credentials are used as the default values.

Example 1

The following example will stop the observers 'oemcli1', 'oemcli2' and 'oemcli3' for the Data Guard configuration associated with primary database 'database'.

```
emcli dg_configure_observers
    -stop -primary_target_name=database
```

```
    -primary_target_type=oracle_database
    -primary_db_creds_name=DB_CRED
    -primary_host_creds_name=HOST_CRED
    -observer_input_file=/scratch/stopObs.props

The following is the content of /scratch/stopObs.props:
observer.0.observer_name=oemcli1
observer.1.observer_name=oemcli2
observer.2.observer_name=oemcli3
```

Example 2

The following example will stop all the observers for the Data Guard configuration associated with primary single-instance database 'database'.

```
emcli dg_configure_observers
    -stop -primary_target_name=database
    -primary_target_type=oracle_database
    -primary_db_creds_name=DB_CRED
    primary_host_creds_name=HOST_CRED -stop_all
```

dg_configure_observers -setMaster

Configures the master observer for a Data Guard configuration associated with the specified primary database.

Format

```
emcli dg_configure_observers -setMaster
    -primary_target_name="<primary target name>"
    -primary_target_type="oracle_database|rac_database"
    -master_observer_name=<name of the master observer>
    [-primary_db_creds_name="<primary database credential name>"]
    [-primary_host_creds_name="<primary database host credential name>"]
```

Options

*   primary_target_name

    Primary database Enterprise Manager target name. This can be either a single-instance database or a cluster database.

*   primary_target_type

    Primary database target type. The allowed values are:

    –   oracle_database: Single instance database

    –   rac_database: Cluster database

*   master_observer_name

    Name of the observer that is currently running and has to be configured as the master observer.

*   primary_db_creds_name

    Primary database named credential for a user with SYSDBA or SYSDG role. Preferred credentials are used as the default values.

*   primary_host_creds_name

    Primary database host named credential for an operating system user who can access the primary database Oracle home. Preferred credentials are used as the default values.

Example

The following example will set observer 'oemcli2' as the master observer for the Data Guard configuration associated with primary single-instance database 'database'.

```
emcli dg_configure_observers
    -setMaster -primary_target_name=database
    -primary_target_type=oracle_database
    -primary_db_creds_name=DB_CRED
    -primary_host_creds_name=HOST_CRED
    -master_observer_name=oemcli2
```

dg_configure_observers -show

Displays the list of observers configured for a Data Guard configuration associated with the specified primary database.

Format

```
emcli dg_configure_observers -show
    -primary_target_name="<primary target name>"
    -primary_target_type="oracle_database|rac_database"
    [-primary_db_creds_name="<primary database credential name>"]
    [-primary_host_creds_name="<primary database host credential name>"]
```

Options

• primary_target_name

  Primary database Enterprise Manager target name. This can be either a single-instance database or a cluster database.

• primary_target_type

  Primary database target type. The allowed values are:

  – oracle_database: Single instance database

  – rac_database: Cluster database

• primary_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Preferred credentials are used as the default values.

• primary_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle home. Preferred credentials are used as the default values.

Example

The following example will list all the observers for the Data Guard configuration associated with primary single-instance database 'database'.

```
emcli dg_configure_observers -show
    -primary_target_name=database
    -primary_target_type=oracle_database
    -primary_db_creds_name=DB_CRED
    -primary_host_creds_name=HOST_CRED
```

dg_configure_observers -delete_alternate_observer

Deletes the Enterprise Manager alternate observer and its associated corrective action job. The Enterprise Manager alternate observer feature is not supported for database version 12.2 and higher.

Format

```
emcli dg_configure_observers -delete_alternate_observer
    -primary_target_name="<primary target name>"
    -primary_target_type="oracle_database|rac_database"
    [-primary_db_creds_name="<primary database credential name>"]
    [-primary_host_creds_name="<primary database host credential name>"]
```

Options

*   primary_target_name

    Primary database Enterprise Manager target name. This can be either a single-instance database or a cluster database.

*   primary_target_type

    Primary database target type. The allowed values are:

    –   oracle_database: Single instance database

    –   rac_database: Cluster database

*   primary_db_creds_name

    Primary database named credential for a user with SYSDBA or SYSDG role. Preferred credentials are used as the default values.

*   primary_host_creds_name

    Primary database host named credential for an operating system user who can access the primary database Oracle home. Preferred credentials are used as the default values.

Example

The following example will delete the Enterprise Manager alternate observer and its associated corrective action job for the Data Guard configuration associated with the primary single-instance database 'database'.

```
emcli dg_configure_observers
    -delete_alternate_observer
    -primary_target_name=database
    -primary_target_type=oracle_database
    -primary_db_creds_name=DB_CRED
    -primary_host_creds_name=HOST_CRED
```

dg_convert_standby

It converts a physical standby database to snapshot standby database. Conversely, converts a snapshot standby database to a physical standby database.

Format

```
emcli dg_convert_standby
    -standby_target_name=""
    -standby_target_type="oracle_database|rac_database"
    [-standby_db_creds_name=""]
```

```
        [-standby_host_creds_name=""]
```

`[ ]` indicates that the parameter is optional.

Options

- standby_target_name

  Standby database Enterprise Manager target name. It can be either a single-instance database or a cluster database.

- standby_target_type

  Standby database target type. The following values are allowed:

  – oracle_database: Single instance database

  – rac_database: Cluster database

- standby_db_creds_name

  Standby database named credential for a user with SYSDBA or SYSDG role. Default Value: Preferred credentials will be used.

- standby_host_creds_name

  Standby database host named credential for an operating system user who can access the standby database Oracle home. Default Value: Preferred credentials will be used.

Example

The following example converts the physical single-instance standby database named 'database' to snapshot standby database.

```
emcli dg_convert_standby
      -standby_target_name="database"
      -standby_target_type="oracle_database"
```

dg_failover

Performs a Data Guard failover from a primary database to a standby database.

Format

```
emcli dg_failover
      -primary_target_name="<primary target name>"
      -primary_target_type="oracle_database|rac_database"
      -standby_target_name="<standby target name>"
      -standby_target_type="oracle_database|rac_database"
      [-primary_db_creds_name="<primary database credential name>"]
      [-primary_host_creds_name="<primary database host credential name>"]
      [-standby_db_creds_name="<standby database credential name>"]
      [-standby_host_creds_name="<standby database host credential name>"]
      [-swap_jobs]
      [-swap_thresholds]
      [-immediate]
```

Options

- primary_target_name

Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- primary_target_type

Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- standby_target_name

Standby database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- standby_target_type

Standby database target type. Specify oracle_database for single instance, or rac_database for cluster.

- primary_db_creds_name

Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- primary_host_creds_name

Primary database host named credential for an operating system user who can access the primary database Oracle home. Default is to use preferred credential.

- standby_db_creds_name

Standby database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- standby_host_creds_name

Standby database host named credential for an operating system user who can access the standby database Oracle home. Default is to use preferred credential.

- swap_jobs

Specifies that scheduled jobs on either the primary or standby database that are of a transferable type (Backup, RMAN Script, and SQL Script) will be moved to the other database after Enterprise Manager detects a fast-start failover. Default is not enabled.

- swap_thresholds

Specifies that monitoring settings will be swapped between the primary and standby databases after Enterprise Manager detects a fast-start failover. Default is not enabled.

- immediate

Specifies an immediately failover without waiting for all the available redo data to be applied on the standby database. This is the fastest way to failover. However, data may be lost. Without this option, a complete failover will be performed minimizing the data loss. Oracle recommends a complete failover. Default is not enabled.

Examples

Example 1

The following command performs a complete failover from a primary single-instance database named "database" to a standby single-instance database named "database1".

```
emcli dg_failover
     -primary_target_name="database"
     -primary_target_type="oracle_database"
```

```
-standby_target_name="database1"
-standby_target_type="oracle_database"
```

Example 2

The following command performs an immediate from a cluster primary database named "database" to a cluster standby database named "database1".

```
emcli dg_failover
     -primary_target_name="database"
     -primary_target_type="rac_database"
     -standby_target_name="database1"
     -standby_target_type="rac_database"
     -immediate
```

dg_remove_configuration

Removes the Data Guard configuration determined from the member database that is a part of the Data Guard configuration.

Format

```
emcli dg_remove_configuration
    -member_target_name="<target name>"
       -member_target_type="oracle_database|rac_database"
       [-member_db_creds_name="<member database credential name>"]
       [-member_host_creds_name="<mamber database host credential name>"]
       [-preserve_redo_dests]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   member_target_name

    Enterprise Manager target name of the member database that is a part of the Data Guard configuration. This can be either a single-instance database or a cluster database.

*   member_target_type

    Database target type of the member database that is a part of the Data Guard configuration. The following values are allowed:

    –   oracle_database: Single instance database

    –   rac_database: Cluster database

*   member_db_creds_name

    Database named credential for a user with SYSDBA or SYSDG role of the member database that is a part of the Data Guard configuration. Default Value: Preferred credentials will be used.

*   member_host_creds_name

    Database host named credential for an operating system user who can access the member database Oracle home. Default Value: Preferred credentials will be used.

*   preserve_redo_dests

Preserve all archived log destinations configured on the primary database after the Data Guard broker configuration is removed. Redo continues to be shipped to the standby databases. Default Value: Not enabled

Examples

Example 1

The following example removes the Data Guard configuration determined from member database named "database".

```
emcli dg_remove_configuration
    -member_target_name="database"
    -member_target_type="oracle_database"
```

Example 2

The following example removes the Data Guard configuration determined from member database named "database", and the redo continues to be shipped to the standby database.

```
emcli dg_remove_configuration
    -member_target_name="database"
    -member_target_type="rac_database"
    -preserve_redo_dests
```

dg_remove_standby_database

Removes a standby database from a Data Guard configuration.

Format

```
emcli dg_remove_standby_database
    -primary_target_name="<primary target name>"
    -primary_target_type="oracle_database|rac_database"
    -standby_target_name="<standby target name>"
    -standby_target_type="oracle_database|rac_database"
    [-primary_db_creds_name="<primary database credential name>"]
    [-primary_host_creds_name="<primary database host credential name>"]
    [-standby_db_creds_name="<standby database credential name>"]
    [-standby_host_creds_name="<standby database host credential name>"]
    [-preserve_redo_dests]
```

```
[ ]  indicates that the parameter is optional.
```

Options

* primary_target_name

  Primary database Enterprise Manager target name. This can be either a single-instance database or a cluster database.

* primary_target_type

  Primary database target type. The following values are allowed:

  – oracle_database: Single instance database

  – rac_database: Cluster database

* standby_target_name

Standby database Enterprise Manager target name. This can be either a single-instance database or a cluster database.

- standby_target_type

  Standby database target type. The following values are allowed:

  – oracle_database: Single instance database

  – rac_database: Cluster database

- primary_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Default Value: Preferred credentials will be used.

- primary_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle home. Default Value: Preferred credentials will be used.

- standby_db_creds_name

  Standby database named credential for a user with SYSDBA or SYSDG role. Default Value: Preferred credentials will be used.

- standby_host_creds_name

  Standby database host named credential for an operating system user who can access the standby database Oracle home. Default Value: Preferred credentials will be used.

- preserve_redo_dests

  Preserve the archived log destination configured on the primary database for this standby database after it is removed from the Data Guard broker configuration. Redo will continue to be shipped to this standby database. Default Value: Not enabled.

Examples

Example 1

The following example removes the standby single-instance database named "database1" from the Data Guard configuration associated with the primary single-instance database named "database".

```
emcli dg_remove_standby_database
    -primary_target_name="database"
    -primary_target_type="oracle_database"
    -standby_target_name="database1"
    -standby_target_type="oracle_database"
```

Example 2

The following example removes the standby cluster database named "database1" from the Data Guard configuration associated with the primary cluster database named "database" and the redo continues to be shipped to this standby database.

```
 emcli dg_remove_standby_database
    -primary_target_name="database"
    -primary_target_type="rac_database"
    -standby_target_name="database1"
```

```
            -standby_target_type="rac_database"
            -preserve_redo_dests
```

dg_switchover

Performs a Data Guard switchover between a primary database and a standby database.

Format

```
emcli dg_switchover
      -primary_target_name="<primary target name>"
      -primary_target_type="oracle_database|rac_database"
      -standby_target_name="<standby target name>"
      -standby_target_type="oracle_database|rac_database"
      [-primary_db_creds_name="<primary database credential name>"]
      [-primary_host_creds_name="<primary database host credential name>"]
      [-standby_db_creds_name="<standby database credential name>"]
      [-standby_host_creds_name="<standby database host credential name>"]
      [-swap_jobs]
      [-swap_thresholds]
      [-tde_wallet_creds_name="<transparent data encryption wallet credentials of the
primary database>"]
```

Options

- primary_target_name

  Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- primary_target_type

  Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- standby_target_name

  Standby database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- standby_target_type

  Standby database target type. Specify oracle_database for single instance, or rac_database for cluster.

- primary_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- primary_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle home. Default is to use preferred credential.

- standby_db_creds_name

  Standby database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- standby_host_creds_name

  Standby database host named credential for an operating system user who can access the standby database Oracle home. Default is to use preferred credential.

- swap_jobs

  Specifies that scheduled jobs on either the primary or standby database that are of a transferable type (Backup, RMAN Script, and SQL Script) will be moved to the other database after Enterprise Manager detects a fast-start failover. Default is not enabled.

- swap_thresholds

  Specifies that monitoring settings will be swapped between the primary and standby databases after Enterprise Manager detects a fast-start failover. Default is not enabled.

- **tde_wallet_creds_name**

  Transparent Data Encryption wallet credentials for the primary database. Use create_named_credential verb to create these credentials as shown below:

```
emcli create_named_credential
    -cred_name=WC1
    -cred_type=GenericPassword
    -auth_target_type='<system>'
    -attributes="GENERIC_PASSWORD:<Primary Database TDE Wallet Password>"
```

Examples

Example 1

The following command performs a switchover between a primary single-instance database named "database" and a standby single-instance database named "database1".

```
emcli dg_switchover
    -primary_target_name="database"
    -primary_target_type="oracle_database"
    -standby_target_name="database1"
    -standby_target_type="oracle_database"
```

Example 2

The following command performs a switchover between a primary single-instance database named "database" and a standby single-instance database named "database1". The scheduled jobs and the monitoring settings on "database" will be moved to "database1" after the switchover is complete.

```
emcli dg_switchover
    -primary_target_name="database"
    -primary_target_type="rac_database"
    -standby_target_name="database1"
    -standby_target_type="rac_database"
    -swap_jobs
    -swap_thresholds
```

dg_verify_config

Verifies the health of the Data Guard configuration of a primary database and performs the following checks:

- Obtains detailed status information from Data Guard broker for each database.

- Verifies that the redo transport is functioning between the primary and standby databases.

- Verifies the proper configuration of the standby redo log files.

- Verifies that the Data Guard broker properties are consistent with underlying database properties.

- Verifies the status of the Agents for all databases.

- Verifies that the preferred credentials are set for all databases.

When the verification process ends, the following fixes are performed automatically:

- Resolves inconsistencies between broker and database properties.

- Creates standby redo log files for any databases when needed.

Format

```
emcli dg_verify_config
      -primary_target_name="<primary target name>"
      -primary_target_type="oracle_database|rac_database"
      [-primary_db_creds_name="<primary database credential name>"]
      [-primary_host_creds_name="<primary database host credential name>"]
      [-reset_inconsistent_props="broker|database"]
      [-create_srls]
      [-verify_only]
```

Options

- primary_target_name

  Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- primary_target_typ

  Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- primary_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- primary_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle home. Default is to use preferred credential.

- reset_inconsistent_props

  Resets the inconsistent properties to the broker or the database values. Allowed values:

  – broker

  – database

  Default is broker.

- create_srls

  Creates standby redo log files for any database that either have none or do not have the proper number and/or sizes. The files will be created as Oracle-managed files (OMF) for any databases that are configured with OMF.

- verify_only

Runs a verification. Neither resolves automatically inconsistent properties nor creates standby redo log files.

Examples

Example 1

The following command verifies the health of the Data Guard configuration associated with the primary single-instance database named "database" and resolves the inconsistencies between the database and the broker properties if any.

```
emcli dg_verify_config
      -primary_target_name="database"
      -primary_target_type="oracle_database"
```

Example 2

The following command only verifies the health of the primary single-instance database named "database".

```
emcli dg_verify_config
      -primary_target_name="database"
      -primary_target_type="rac_database"
      -verify_only
```

diagchecks_deploy_status

Gets the status of diagnostic checks deployments against different target types.

Format

```
emcli diagchecks_deploy_status
        [-target_type=<type>]*

[ ]  indicates that the parameter is optional
```

Options

• **target_type**

  Type of target. You can specify multiple values.

diagchecks_deploy_tglist

Gets the target list for a particular deployment type for a target type.

Format

```
emcli diagchecks_deploy_tgtlist
        -target_type=<type>
        -deploy_type=<CURRENT|OLDER|MISSING|ALL>
        [-show_excludes]

[ ]  indicates that the parameter is optional
```

Options

• **target_type**

  Type of target. You can specify multiple values.

- **deploy_type**

  Deployment type of either CURRENT, OLDER, MISSING, or ALL.

- **show_excludes**

  For targets where excludes have been set, print them.

diagnose_awr

Run diagnostics for the AWR Warehouse database specified by the target name and target type parameters. If the specified database is an AWR Warehouse, then warehouse diagnostics will be run, else source side diagnostics will be run.

Format

```
emcli diagnose_awr
        -target_name=<target_database_name>
        -target_type=<target_database_type>
```

Options

- target_name

  Name of the target database (AWR Warehouse or source database).

- target_type

  Type of target. The possible values for target type are `oracle_database`, `oracle_pdb`, and `rac_database`.

Output

Success/Error

Example

The following example runs diagnostics for the target AWR Warehouse database, `sample_database`:

```
emcli diagnose_awr
        -target_name=sample_database
        -target_type=oracle_database
```

disable_audit

Disables auditing for all user operations.

Format

```
emcli disable_audit
```

Example

This example disables auditing for all operations.

```
emcli disable_audit
```

disable_config_history

Disables configuration history computation for a target type.

Format

```
emcli disable_config_history
      -target_type="{target type|'*'}"
```

Options

- target_type

  Target type for which the configuration history is being disabled. The value should be the internal name or "*" to indicate all target types.

Examples

Example 1

This example disables configuration history computation for the host target type.

```
emcli disable_config_history -target_type="host"
```

Example 2

This example disables configuration history computation for all target types.

```
emcli disable_config_history -target_type="*"
```

disable_mda_finding_types_for_targets

Disables the specified MDA finding types for the specified targets. The finding types are disabled for a target only if the finding is applicable for the specified target type and the EM CLI user has permissions on the target.

Format

```
emcli disable_mda_finding_types_for_targets      [-finding_types="<list of
finding types>"]
      [-targets="<list of targets and their target types>"]

[ ]  indicates that the parameter is optional.
```

Options

- finding_types

  List of finding types. The default separator to be used is ';'.

- targets

  List of targets with their target types. The default separator to be used is ';'.

Examples

Example 1

The following example disables a single finding type on a single target:

```
emcli disable_mda_finding_types_for_targets
      -finding_types="oracle.sysman.emas.wls_gc_overhead"
      -targets="/EMGC_EMGC_DOMAIN/EMGC_DOMAIN/EMGC_OMS1:weblogic_j2eeserver"
```

Example 2

The following example disables multiple finding types on multiple targets:

```
emcli disable_mda_finding_types_for_targets
      -
finding_types="oracle.sysman.emas.wls_gc_overhead;oracle.sysman.emas.wls_heap_config"
      -targets="/EMGC_EMGC_DOMAIN/EMGC_DOMAIN/EMGC_OMS1:weblogic_j2eeserver;/
test_base_domain/base_domain/MS1:weblogic_j2eeserver"
```

disable_sla

Disables an SLA for a target.

Format

```
emcli disable_sla
    -targetName=<target_name>
    -targetType=<target_type>
    -slaName=<SLA_name>
```

Options

*   **targetName**

    Name of the target.

*   **targetType**

    Type of target.

*   **slaName**

    Name of the SLA.

Examples

This example disables an SLA named 'gold_sla' for target my_service (generic_service).

```
emcli disable_sla
      -targetName='my_service' -targetType='generic_service'
      -slaName='gold_sla'1
```

disable_snapclone

Disables the Snap Clone feature for a database.

Format

Standard Mode

```
emcli disable_snapclone
      -db_name="<database name>"
```

Interactive or Script Mode

```
disable_snapclone(
      db_name="<database name>"
```

```
            )
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   db_name

    Name of the database.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example disables the Snap Clone feature for the database `testmstr`:

```
emcli disable_snapclone
              -db_name="testmstr"
```

disable_target

Disables the target on both the Management Repository and Management Agent side.

Syntax

```
 emcli disable_target
        -type="target_type1"
        -name="target_name1"
        -agent="agent_name1"
        [-ignore_invalid_target]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   -type=target_type1

    Target type of the target being disabled.

*   -name=target_name1

    Name of the target. You can use the percentage character (%) as a wild character to disable all targets of a specified type on a specified Management Agent.

*   -agent=agent_name1

    Name of the Management Agent on which the target has to be disabled.

*   -ignore_invalid_target

    When specified, the process ignores invalid targets.

Examples

Example 1

The following command disables the target on the Management Agent and OMS and it fails if the target is invalid.

```
    emcli disable_target
        -type="oracle_em_service"
```

```
        -name="TestService"
        -agent="TestAgent"
```

Example 2

The following command disables all the targets of "oracle_em_service" type on the "TestAgent" Management Agent both on the OMS and Management Agent side. It ignores invalid targets.

```
emcli disable_target
    -type="oracle_em_service"
    -name="%"
    -agent="TestAgent"
    -ignore_invalid_target
```

disable_test

Disables monitoring of a Services test.

Format

```
emcli disable_test
    -name=<target_name>
    -type=<target_type>
    -testname=<test_name>
    -testtype=<test_type>
```

Options

*   **name**

    Service target name.

*   **type**

    Service target type.

*   **testname**

    Test name.

*   **testtype**

    Test type.

Examples

This example disables the HTTP test named `MyTest` for the `generic_service` target named `MyTarget`.

```
emcli disable_test -name='MyTarget' -type='generic_service'
    -testname='MyTest' -testtype='HTTP'
```

disassociate_user_profile

Removes the association between a user profile and a set of users.

Format

Standard Mode

```
emcli disassociate_user_profile
      -name="profile name"
      -users="users to be disassociated"
```

Interactive or Script Mode

```
emcli disassociate_user_profile(
      name="profile name";
      [,description="profile desc"]
      [,users="users to be associated"]
      [,included_profiles="profile to be included"]
      )
```

Options

• name

  The name of the user profile that is to be disassociated.

• users

  The users from whom the profile is to be disassociated.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example disassociates the `profile1` user profile from `user1` and `user2`.

```
emcli disassociate_user_profile
      -name=profile1
      -users="user1";"user2"
```

discover_bda_cluster

Performs Big Data discovery for the specified host. Can be used for new discovery or for rediscovery of the latest configuration changes.

Format

```
emcli discover_bda_cluster
        -hostname="host_name"
        -host_credential="host_named_cred"
        -ilom_credential="ilom_named_cred"
        -infiniband_credential="ibswitch_named_cred"
        -cloudera_credential="cloudera_named_cred"
        -snmp_string="SNMP_community_string"
```

Options

• **hostname**

  The name of host in the Big Data Network.

• **host_credential**

  Named credentials for the `oracle` OS account that owns a Management Agent home.

• **ilom_credential**

Named credentials for the `root` OS account on an Oracle Integrated Lights Out Manager (Oracle ILOM) server in the Big Data Network.

- **infiniband_credential**

  Named credentials for the `nm2user` OS account on an InfiniBand switch in the Big Data Network.

- **cloudera_credential**

  Named credentials for the `admin` account of the Cloudera Manager that manages the CDH cluster.

- **snmp_string**

  SNMP community string for PDU and Cisco switch traps. The read-only string is public.

Example

The following example performs BDA cluster discovery on the host named `acme101.com`. If the cluster already exists, updates the latest configuration.

```
emcli discover_bda_cluster
        -hostname="acme101.com"
        -host_credential="HOST_CRED"
        -ilom_credential="ILOM_CRED"
        -infiniband_credential="IB_CRED"
        -cloudera_credential="CM_CRED"
        -snmp_string="public"
```

discover_cloudera_cluster

Discovers the Hadoop cluster for the specified Cloudera Manager host. Can also be used for rediscovery of the latest cluster configuration changes.

Format

```
emcli discover_cloudera_cluster
        -hostname = "host_name"
        -cloudera_credential = "cloudera_named_cred"
        -host_credential  = "host_named_cred"
```

Options

- hostname

  Name of one of the hosts that form the cluster.

- cloudera_credential

  Named credentials for the Cloudera Manager managing the cluster.

- host_credential

  Named credentials for the specified host.

Example

The following example discovers the Hadoop cluster that includes a host named acme101.com, using the provided named credentials:

```
emcli discover_bda_cluster
        -hostname="acme101.com"
```

```
                    -cloudera_credential="CM_CRED"
                    -host_credential="HOST_CRED"
```

discover_coherence

Discovers one or more non-managed Coherence clusters (Managed Coherence clusters are discovered as part of the WLS domain discovery process).

Format

```
emcli discover_coherence
      -input_file=coherence_discovery_file:file_path
      [-debug]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **input_file**

    Fully-qualified path to a CSV-formatted file containing one line of details per Coherence cluster. The structure of the CSV file is as follows:

    ```
    <Management Node host machine name>,
             <Management Node listen port>,
             <Management Node username - optional>,
             <Management Node password - optional>,
             <Management Node service name - optional>,
             <Agent url>
    ```

    For example:

    ```
    host1.companyA.com,9910,,,,https://host1.companyA.com:3872/emd/main/,
    ```

    For information about the input_file parameter, see -input_file Syntax Guidelines.

*   **debug**

    Runs the verb in verbose mode for debugging purposes.

Examples

This example reads the my_clusters_info.csv file to determine the clusters to be added to Cloud Control.

```
emcli discover_coherence
  -input_file=coherence_discovery_file:"c:\emcli\my_clusters_info.csv
```

discover_db

Discovers and adds or promotes single instance, Real Application Cluster (RAC) and pluggable databases along with their associated targets, according to the preferences provided in the input file.

Format

Standard Mode

```
emcli discover_db
        -input_file="db_discovery_file:/path/db_discovery_file"
        [-debug]
```

```
                  [-promote]
                  [-add_missing_cluster]
                  [-check]
```

Interactive or Script Mode

```
discover_db(
        ddb_discovery_file="db_discovery_file"
        [,debug=True/False]
        [,promote=True/False]
        [,add_missing_cluster=True/False]
        [,check=True/False]
         )
[ ]  indicates that the parameter is optional.
```

Options

- db_discovery_file

  This file contains the preferences required to add a database.The file must organized into blocks, with each block specific to a particular host or cluster. Each input block includes lines for a host or cluster name, database credentials, and the target type. Optionally, each input block can include discovery hints such as discovery time-out and database status.

  Input blocks are separated by an empty line.

  Database credentials are given as database name/credentials (user name, password, role) pairs.

  If you want to specify common database credentials for all of the databases on a given host, then '*' can be specified instead of the database name.

  Specify the database target type as follows:

  – `rac_database` in case of a RAC database

  – `oracle_database` in case of a single instance database.

  Specify ASM credentials as follows:

  `asm_creds=Username,Password,Role`

  To add targets to groups, specify the group names separated by commas as follows:

  `group_names=Group Name 1,Group Name 2,Group Name 3`

  The names of targets to be added or promoted can include a prefix or a suffix, as follows:

  `db_name_prefix=Prefix pattern`

  `db_name_suffix=Suffix pattern`

  The target properties (global or user added properties) can be saved by specifying the input as follows:

  `global_props=Property 1 Name: Property 1 Value, Property 2 Name: Property 2 Value`

  `user_added_props=Property 1 Name: Property 1 Value, Property 2 Name: Property 2 Value`

  The structure of the input file when the host name is provided is as follows:

  ```
  <host_name=Host Machine Name>
          <db_creds=Target Name:Username,Password,Role>
  ```

```
<target_type=Target Type -optional>
<discovery_hints - optional>
   <asm_creds=Username,Password,Role -optional>
   <group_names=Group Name 1,Group Name 2,Group Name 3 -optional>
   <db_name_prefix=Prefix pattern -optional>
   <db_name_suffix=Suffix pattern -optional>
<global_props=Property 1 Name: Property 1 Value -optional>
<user_added_props=Property 1 Name: Property 1 Value  -optional>
```

The structure of the Input file when the cluster name is provided is as follows:

```
<cluster_name=Name of the Cluster>
   <db_creds=Target Name:Username,Password,Role>
   <target_type=Target Type -optional>
   <discovery_hints - optional>
      <asm_creds=Username,Password,Role -optional>
      <group_names=Group Name 1,Group Name 2,Group Name 3 -optional>
      <db_name_prefix=Prefix pattern -optional>
      <db_name_suffix=Suffix pattern -optional>
   <global_props=Property 1 Name: Property 1 Value -optional>
   <user_added_props=Property 1 Name: Property 1 Value  -
optional>

      -----------------
      cluster_name=slc00dsno-cls
      db_creds=db1:sys,oracle,SYSDBA
      db_creds=db2:sys,welcome,SYSDBA
      target_type=oracle_database
         asm_creds=sys,oracle,SYSDBA
         group_names=group1,group2
      cluster_name=slc00dtfg-r
      db_creds=*:sys,oracle,SYSDBA
      target_type=rac_database
      db_status=up
         db_name_prefix=prefix_
         db_name_suffix=_suffix
      global_props=Comment:Test,Contact:4444,Cost Center:CC
```

For example, the contents of the file can be (2 blocks of input in the file):

```
      -----------------
      host_name=slc00dsn.mycompany.com
      db_creds=db1:sys,oracle,SYSDBA
      db_creds=db2:sys,welcome,SYSDBA
      target_type=oracle_database
         asm_creds=sys,oracle,SYSDBA
         group_names=group1,group2

      host_name=slc00dtf.mycompany.com
      db_creds=*:sys,oracle,SYSDBA
      target_type=rac_database
      db_status=up
      db_name_prefix=prefix_
         db_name_suffix=_suffix
         global_props=Comment:Test,Contact:4444,Cost Center:CC
```

- debug

  Specifies if debug is on or off.

- promote

Promotes targets obtained from auto discovery.

* add_missing_cluster

   Adds the cluster in case the mentioned host is valid and the corresponding cluster is not added.

* check

   Runs discovery and shows the results. It does not add the targets.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example discovers a database target and promotes it:

```
emcli discover_db
      -input_file=db_discovery_file:/emcli/test.txt <-debug> <-promote> <-
add_missing_cluster> <-check>
```

discover_fa

Discovers multiple Fusion Applications domains by reading the Fusion Applications domain discovery file and saving the host-wise discovered targets to the Agents provided in the Host Agent Mapping file. If the Host Agent mapping file is not provided, the local Agent (that is, the Agent on the same host as the target) is used to save/monitor the discoverd targets as well. If a local Agent is not found, the default discovery Agent is used to save/monitor the discoverd targets as well.

> **Note:**
>
> Although this verb supports discovering multiple Fusion instances at one time by adding all the details in one file, it is advisable to discover each Fusion instance separately using individual EM CLI discover_fa commands run multiple times.

Format

```
emcli discover_fa
      -input_file=fa_domain_discovery_file:file_path
      [-input_file=host_agent_mapping_file:file_path]
      [-input_file=pf_domain_cred_mapping_file:file_path]
      [-debug]

[ ]  indicates that the parameter is optional
```

Options

* **input_file=fa_domain_discovery_file**

   Fully-qualified path to a CSV-formatted file containing one line of details per domain to be added. The valid WebLogic version value is 10. The structure of the CSV file is as follows:

   ```
   <WebLogic Server version>,
   <Administration Server host machine name>,
   ```

```
<Administration Server listen port>,
<Administration Server username>,
<Administration Server password>,
<External Options - optional>,
<JMX Protocol - required only if SSL enabled>,
<JMX Service URL - required only if SSL enabled>,
<Unique Domain Identifier>,
<Agent URL/>,
<Discover Down Servers - optional - Default if not specified is false
starting <PS1. Before PS1 the default for this is true>,
<Use Same Credentials for All Domains in the Fusion Instance - optional -
Default if <not specified is true>
```

For example:

```
10,mco01.mycompany.com,7001,weblogic,welcome1,,,,my_farm_
    01,https://mco01.mycompany.com:3872/emd/main/,,
10,mco01.mycompany.com,7001,weblogic,welcome1,,,,my_farm_
    01,https://mco01.mycompany.com:3872/emd/main/,true,
10,mco01.mycompany.com,7001,weblogic,welcome1,,,,my_farm_
    01,https://mco01.mycompany.com:3872/emd/main/,true,true
10,mco01.mycompany.com,7001,weblogic,welcome1,,,,my_farm_
    01,https://mco01.mycompany.com:3872/emd/main/,false,true
```

For information about the input_file parameter, see -input_file Syntax Guidelines.

- **input_file=host_agent_mapping_file**

  Fully-qualified path to a CSV-formatted file containing multiple lines of host system names where Managed Servers are to be monitored, and the Agent to be used to monitor each host's Managed Servers.

  For example:

  ```
  mycompany.com,https://mco01.mycompany.com:3872/emd/main
  ```

  For information about the input_file parameter, see -input_file Syntax Guidelines.

- **input_file=pf_domain_cred_mapping_file**

  Fully-qualified path to a CSV-formatted file containing multiple lines of WebLogic admin credentials for each domain of a fusion instance, where the credentials are different from those added in the fa_domain_discovery file.

  The same credentials are used for all the domains in a Fusion Application instance unless the credentials are overwritten in the pf_domain_cred_mapping file.

  For example:

  ```
  <UniqueKey - "<Fusion Instance
       Identifier><CommonDomainDisplayName>">,<Administration Server
       username>,<Administration Server password>,
  <UniqueKey - "<Fusion Instance
       Identifier>-<CommonDomainDisplayName>">,<Administration Server
       username>,<Administration Server password>,<Administration Server Host
       Name>

  Example:
  fi9-FS,weblogic12,welcome1,
  fi9-PRJ,faadmin,fusionfa1,
  fi9-PRC,faadmin,fusionfa1,myhost.us.example.com
  fi9-PRC,,,myhost.us.example.com
  ```

  For information about the input_file parameter, see -input_file Syntax Guidelines.

- **debug**

  Runs the verb in verbose mode for debugging purposes.

Examples

Example 1

This example reads the my_domains_info.csv file to determine the Fusion Instances to be added to Cloud Control, reads the my_agent_mapping.csv file to determine which Agents should monitor which host's Managed Servers, and reads the my_domain_cred_mapping.csv file to determine which credentials are to be used to discover an individual product family.

```
emcli discover_fa
      -input_file=fa_domain_discovery_file:c:\emcli\my_domains_info.csv
      -input_file=host_agent_mapping_file:c:\emcli\my_agent_mapping.csv
      -input_file=pf_domain_cred_mapping_file:c:\emcli\my_domain_cred_mapping.csv
```

Example 2

```
emcli discover_fa -input_file=fa_domain_discovery_file:/tmp/emcli/
domain_discovery_file.txt -input_file=host_agent_mapping_filee:/tmp/emcli/
host_agent_mapping_file.txt -input_file=pf_domain_cred_mapping_file:/tmp/emcli/
pf_domain_cred_mapping_file.txt -debug
```

discover_gf

Discovers Multiple GlassFish Domains by reading the Domain Discovery file and saving the discovered targets of the host to the Agents provided in the Host Agent Mapping file. If the Host Agent mapping file is not provided, the local Agent (the Agent on the same host as the target) is used to save/monitor the discovered targets. If a local Agent is not found, the default discovery Agent is used to save/monitor the discovered targets.

Format

```
$emcli discover_gf
      -input_file=domain_discovery_file:file_path
      [-input_file=host_agent_mapping_file:file_path]
      [-debug]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **input_file=domain_discovery_file**

  Fully-qualified path to a CSV-formatted file containing one line of details per domain to be added. The structure of the CSV file is as follows:

```
<Administration Server host machine name>,
<Administration Server listen port>,
<Administration Server username>,
<Administration Server password>,
<Unique Domain Identifier>,
<Agent url - optional >,
<Protocol - optional >,
<Service URL - optional>,
<External Options - optional>,
<Discover Down Servers - optional - Default if not specified is false>,\n" +
```

  For example:

```
mco01.mycompany.com,4848,admin,welcome1,my_domain_01,https://
mco01.mycompany.com:3872/emd/main
mco01.mycompany.com,4848,admin,welcome1,my_domain_01,https://
mco01.mycompany.com:3872/emd/main,http,,,true
```

For information about the input_file parameter, see -input_file Syntax Guidelines.

- **input_file=host_agent_mapping_file**

    Fully-qualified path to a CSV-formatted file containing multiple lines of host system names where Managed Servers are to be monitored, and the Agent to be used to monitor each host's Managed Servers. The structure of the CSV file is as follows:

    ```
    <target_host1>,<save_to_agent1>
    <target_host2>,<save_to_agent3>
    ```

    For example:

    ```
    mycompany.com,https://mco01.mycompany.com:3872/emd/main
    ```

    For information about the input_file parameter, see -input_file Syntax Guidelines.

- **debug**

    Runs the verb in verbose mode for debugging purposes.

Examples

Example 1

```
$emcli discover_gf -input_file=domain_discovery_file:/tmp/emcli/
domain_discovery_file.txt
```

Example 2

```
$emcli discover_gf -input_file=domain_discovery_file:/tmp/emcli/
domain_discovery_file.txt -input_file=host_agent_mapping_file:/tmp/emcli/
host_agent_mapping_file.txt -debug
```

discover_jboss

Discovers a JBoss target in Oracle Enterprise Manager. JBoss targets include JBoss JEE servers, JBoss application servers, JBoss domains, and JBoss partitions (for example, jboss_jeeserver, jboss_app_server , jboss_domain or jboss_partition).

For additional information on how to use the discover_jboss EM CLI verb, refer to Oracle Enterprise Manager Command Line Interface documentation available on the Oracle Technology Network.

Format

```
emcli discover_jboss
      -host
      -version
      -port
      -agent
      -auth_type
      [-username]
      [-password]
      [-debug]

[ ]  indicates that the parameter is optional.
```

Options

- host

  Fully qualified name of the JBoss host in the case of a standalone/JBoss 6 target or fully qualified name of the JBoss Domain Controller in the case of a domain based discovery.

- version

  JBoss target version (only versions 6 and 7 are allowed).

- port

  HTTP management port/JMX connector port.

- agent

  Monitoring Agent.

- auth_type

  JBoss authentication type (digest_authentication, basic_authentication, or none). Digest authentication is not allowed for JBoss 6 targets.

- username

  JBoss host user name.

- password

  JBoss host password.

- debug

  Runs the verb in verbose mode for debugging purposes.

Examples

Example 1

The following example discovers JBoss targets of version 7 in Oracle Enterprise Manager.

```
emcli discover_jboss
      -host=host1.example.com
      -version=7
      -port=1234
      -agent=host1.example.com:31453
      -auth_type=digest_authentication
      -username=jboss
      -password=jboss123
```

Example 2

The following example discovers JBoss targets of version 6 in Oracle Enterprise Manager.

```
emcli discover_jboss
      -host=host1.example.com
      -version=6
      -port=1234
      -agent=host1.example.com:31453
      -auth_type=none
```

discover_siebel

Discovers Siebel Enterprise instances.

Format

```
emcli discover_siebel
       -input_file=enterprise_info_file:<file_path>
       [-out_file='<fully_qualified_path_of_output_file>']
       [-precheck]
       [-debug]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **input_file**

  The input file should be in a CSV format. The structure of the CSV file is as follows:

  ```
  GATEWAY_HOST = < Gateway Server Host >,
  PORT = < Gateway Server Port - optional Default if not specified is 2320 >,
  INSTALL_DIR = < Gateway Server Install Directory - optional >,
  ENTERPRISE_NAME = < Siebel Enterprise Name >,
  SIEBEL_USERNAME = < Siebel Enterprise User Name >,
  SIEBEL_PASSWORD = < Siebel Enterprise Password >,
  DATABASE_USERNAME = < Database User Name >,
  DATABASE_PASSWORD = < Database Password >
  ```

  > **Note:**
  >
  > INSTALL_DIR is a mandatory parameter for discovering Siebel version 8.2.2 and above.

  This example shows discovery of a Siebel Enterprise (siebel) with the gateway located at host 'host1', installed at location 'Location1' and running at port '23201', with a Siebel user name and password of 'sbluser' and 'SBLpass' respectively, and a database user name and password of 'dbuser' and 'DBpass' respectively.

  ```
  GATEWAY_HOST=host1,PORT=23201,INSTALL_DIR=Location1,
  ENTERPRISE_NAME=siebel,SIEBEL_USERNAME=sbluser,
  SIEBEL_PASSWORD=SBLpass,DATABASE_USERNAME=dbuser,
  DATABASE_PASSWORD=DBpass
  ```

  Special cases for commas:

  - If any entry, such as a password, has a comma (,) you need to add it as a backslash comma (\,) in the CSV file. For instance, if SIEBEL_PASSWORD is we,lco,me1 the entry in the CSV file would be SIEBEL_PASSWORD = we\,lc\,ome1 .

  - If any entry, such as a password, has a backslash followed by a comma(\,) you need to add it as as two backslashes followed by a comma(\\,) in the CSV file. For instance, if SIEBEL_PASSWORD is we\,lco\,me1 the entry in the CSV file would be SIEBEL_PASSWORD = we\\,lc\\,ome1 .

    For information about the input_file parameter, see -input_file Syntax Guidelines.

- **out_file**

Command output is redirected to this file. If not specified, output is printed on the console.

- **debug**

  Executes in verbose mode and generates additional debug log messages in the output. If specified, detailed output is printed.

- **precheck**

  Performs a mock discovery of the Siebel enterprise by executing all of the checks and validations. This option lists the results of these steps to the user for review prior to an actual discovery. It ensures that all prerequisite are met, and discovery does not occur if prerequisites are met.

Examples

Example 1

This example reads the my_enterprise_info.csv file to determine the Siebel Enterprise instances to be added to Cloud Control. The output of the command is redirected to the discovery_output.txt file.

```
emcli discover_siebel
      -input_file=enterprise_info_file:'c:\emcli\my_enterprise_info.csv'
      -out_file='c:\emcli\discovery_output.txt'
      -debug
```

Example 2

This example is the same as the example above, except it adds the -precheck option, which confirms if the precheck is successful, or shows errors if it failed.

```
emcli discover_siebel
      -input_file=enterprise_info_file:'c:\emcli\my_enterprise_info.csv'
      -out_file='c:\emcli\discovery_output.txt'
      -debug
```

discover_was

Discovers an IBM Websphere target in Oracle Enterprise Manager. Targets include an IBM Webshpere cell or an IBM Websphere JEE server (for example,websphere_cell or websphere_jeeserver).

For additional information on how to use the discover_was EM CLI verb, refer to Oracle Enterprise Manager Command Line Interface documentation available on the Oracle Technology Network.

Format

```
emcli discover_was      -host     -port     -version     -key_file     -dir     -
agent      [-username]     [-password]     [-debug]

[ ]  indicates that the parameter is optional.
```

Options

- host

  Fully qualified name of the host where the WebSphere Application Server is running.

- version

Websphere Application Server version. Versions 6.0, 6.1, 7.0, 8.0 or 8.5 are allowed.

• port

SOAP connector port of the server.

• key_file

The absolute path of the Monitoring Agent Trust file. This option is required if the port is SSL enabled.

• dir

The absolute path of the directory where Deployment Manager is installed. For remote discovery, specify the path of the directory on the Agent Host where the required jar files have been copied.

• agent

Monitoring Agent.

• username

Websphere host username if security is enabled

• password

Websphere host password if security is enabled.

• debug

Runs the verb in verbose mode for debugging purposes.

Examples

Example 1

The following example discovers targets in an IBM Websphere cell.

```
emcli discover_was
     -host=host1.example.com
     -version=8.5
     -port=1234
     -key_file=/path/to/monitoring/agent/trust/store/file
     -dir=/to/websphere/server/install/home
     -agent=host1.example.com:12345
     -username=was
     -password=was123
```

Example 2

The following example discovers a standalone IBM Websphere application server.

```
emcli discover_was
     -host=host1.example.com
     -version=8.5
     -port=1234
     -key_file=/path/to/monitoring/agent/trust/store/file
     -dir=/path/to/websphere/server/install/home
     -agent=host1.example.com:12345
     -username=was
     -password=was123
```

discover_wls

> **Note:**
>
> The following Weblogic Server versions are supported in Enterprise Manager 13.2:
>
> 12.2.1.x, 12.1.3.x, 12.1.2.x
>
> 10.3.6.x, 10.3.5.x, 10.3.4.x, 10.3.3.x, 10.3.2.x, 9.2.x
>
> The information in this section applies only to these versions.

Purpose

Used to discover one or more WebLogic Domains (along with Oracle Fusion Middleware 11*g* and 12*c* software deployed to it), and to specify which Management Agent should monitor which hosts' Managed Servers.

Function

This verb discovers one or more Oracle WebLogic Server Domains. It reads a file labeled domain_discovery_file to discover WebLogic Server. Note that if you attempt to discover an already discovered WebLogic Server, the discovered WebLogic Server domain will be refreshed.

Requirements

To discover the WebLogic Server, the Administration Server must be up and running. After initial discovery or during refresh of domain membership, the Administration Server is not required to be up for general WebLogic Server monitoring. After initial discovery or during refresh of domain membership, the Managed Server is not required to be up for general WLS monitoring. Oracle recommends ensuring all Managed Servers to be managed by Cloud Control be up during discovery.

domain_discovery_file is required; discovery cannot occur without it. You must create the CSV (comma-separated values) formatted file before performing discovery. To save the discovered components to a specific Management Agent for monitoring, the discover_wls verb reads a second file labeled host_agent_mapping_file. If host_agent_mapping_file does not exist, the Management Agent specified in domain_discovery_file that performs the actual discovery is used as the Agent that monitors all discovered targets.

Usage With generate_discovery_input Verb

The generate_discovery_input verb creates a discovery input file automatically based on the targets discovered from the automatic discovery operation. You can then use this discovery input file in conjunction with the discover_wls verb to further automate the process of promoting discovered domains as fully managed targets.

Format

```
emcli discover_wls
        -input_file=domain_discovery_file:file_path
        [-input_file=host_agent_mapping_file:file_path]
        [-input_file=disable_target_types_file:file_path]
        [-input_file=global_target_props_file:file_path]
```

```
        [-debug]
```

```
[ ]   indicates that the parameter is optional
```

Options

- **input_file=domain_discovery_file**

  Fully-qualified path to a CSV (Comma-Separated Values) formatted file that contains one line of details per domain to be added. Each line has the format shown for domain_discovery_file in the "File Structures" section below.

  Note the following points about the format of domain_discovery_file:

  Options —

  – The order of parameters is fixed. You must provide the parameters in the same order as shown for domain_discovery_file in the "File Structures" section below.

  – If you want to use a comma (,) in any of the parameters provided, you must escape the comma with a backslash as shown in This example, in which a backslash precedes the comma in the password my,pwd:

  ```
  10,domain123.xyx.us,11990,weblogic,my\,pwd,,,farm_demo,https://
  myco01.mycompany.com:3872/emd/main/
  ```

  Delimeters and Requirements —

  – Use a comma (,) as the delimiter.

  – Delimiters must be present even if the corresponding parameter is not provided. See the last line for domain_discovery_file in the "File Structures" section below.

  – If you want to use a comma (,) in one of the parameters provided, you must escape the comma (,) with a backslash. In This example, the password contains a comma:

  ```
  10,mco01.mycompany.com,7001,weblogic,welco\,me1,,,,my_farm_01,https://
  mco01.mycompany.com:3872/emd/main/
  ```

  – If you want to use a backslash in one of the parameters provided, you must escape the backslash with another backslash. In This example, the password contains a backslash:

  ```
  10,mco01.mycompany.com,7001,weblogic,we\,lco \ \
  me1,,,,my_farm_01,https://mco01.mycompany.com:3872/emd/main/,true,false
  ```

  – The minimum number of tokens is 10, the maximum is 18 tokens. The following are the 10 required tokens:

    * <WebLogic Server version>

    * <Administration Server host machine name>

    * <Administration Server listen port>

    * <Administration Server username>

    * <Administration Server password>

    * <External Parameters - optional>

    * <JMX Protocol - required only if SSL enabled>

    * <JMX Service URL - required only if SSL enabled>

* *   <Unique Domain Identifier>

* *   <Agent URL>

  –   The order of parameters is fixed. You must provide the parameters in the same order as specified in the sample file structure shown in the "File Structures" section below.

  For information about the input_file parameter, see -input_file Syntax Guidelines.

* **input_file=host_agent_mapping_file**

  Fully-qualified path of the CSV (Comma-Separated Values) formatted file that contains multiple lines of host system names where managed servers are to be monitored, and specifies the Management Agent used to monitor each host's managed servers. Each line has the following format:

  ```
  <Discovered_target_host_machine_name>,<Agent_URL_to_save/monitor_the_host>
  ```

  For example:

  ```
  myco01.mycompany.com,https://myco01.mycompany.com:3872/emd/main/
  myco02.mycompany.com,https://myco02.mycompany.com:3872/emd/main/
  myco03.mycompany.com,https://myco03.mycompany.com:3872/emd/main/
  ```

  Definitions for the parameters are as follows:

  – **Discovered_target_host_machine_name**

    Host machine with installed WebLogic Servers that need to be discovered. Use full host names, such as myco01.mycompany.com instead of myco01.

  – **Agent_URL_to_save/monitor_the_host**

    URL for the Management Agent to be used to monitor all discovered targets on the corresponding host.

  Delimeters and Requirements —

  –   Use a comma (,) as the delimiter.

  –   The total number of tokens in each line is fixed and should be equal to 2.

  –   The order of parameters is fixed. You must provide the parameters in the same order as shown in the sample file structure shown in the "File Structures" section below.

  –   <target_host1> and <save_to_agent1> are both mandatory parameters.

  For information about the input_file parameter, see -input_file Syntax Guidelines.

* **input_file=disable_target_types_file**

  Fully-qualified path to a CSV (Comma-Separated Values) formatted file containing multiple lines of internal target type names that should not be discovered.

  For example:

  ```
  oracle_soa_composite
  j2ee_application
  ```

  If the discover_wls verb is run against a Fusion Applications WebLogic Server domain, the disabled target types can include Fusion Applications target types.

* **global_target_props_file**

  Fully qualified path to a CSV formatted file containing target properties. The structure of the file is as follows:

  <Administration Server host machine name>,

<Administration Server listen port>,

<Comment>,

<Contact>,

<Cost Center>,

<Department>,

<Life Cycle Status. Valid Values: None, Mission Critical, Production, Staging, Test, Development>,

<Line Of Business>,

<Location>,

<Apply to all targets - true/false. Default is false>

To set global properties for a domain specified in a discovery input file, the administration server host and administration server port specified in the global properties file should match host and port specified in domain discovery input file.

- **debug**

  Runs this verb in verbose mode for debugging purposes.

File Structures

domain_discovery_file for WebLogic Server versions

This example shows the structure of a sample domain_discovery_file for WebLogic Server. OPT signifies an optional parameter. The last entry shows the format when optional parameters External Options, JMX Protocol, JMX Service URL, Management Agent URL, Node Manager Username, and Node Manager Password are not provided.

```
<WebLogic Server version>,
<Administration Server host machine name>,
<Administration Server listen port>,
<Administration Server username>,
<Administration Server password>,
<External Options - optional>,
<JMX Protocol - Required only if SSL enabled>,
<JMX Service URL - Required only if SSL enabled>,
<Unique Domain Identifier>,
<Agent URL/>,
<Discover Down Servers - optional - Default if not specified is false>,
<Use Credential Store - optional - Default if not specified is false>
<Enable Refresh Job - optional - Default if not specified is false>
<Use Host Name in Service URL - optional - Default if not specified is false>
<Node Manager Username> - optional - Default if not specified is the
Administration Server Username>
<Node Manager Password> - optional - Default if not specified is the
Administration Server password>
```

For example:

```
10,mco01.mycompany.com,7001,weblogic,welcome1,,,,my_farm_01,https://
mco01.mycompany.com:3872/emd/
main/,false,false,false,false,NodeUsername,NodePassword
```

Definitions for the parameters are as follows:

- **WebLogic Server Version**

  Valid values are 9, 10, or 12. This example shows a sample entry in domain_discovery_file to discover WebLogic Server version 10:

  ```
  10,myco01.mycompany.com,7001,weblogic,welcome1,,,,soa_farm,
  https://myco02.mycompany.com:8723/emd/main/
  ```

- **Administration Server Host**

  Full host name of the WebLogic Administration Server that needs to be discovered; for example, myco01.mycompany.com. This is a mandatory parameter.

- **Port**

  Listen port of the WebLogic Administration Server.

- **Username**

  Login user name for the WebLogic Administration Server.

- **Password**

  Login password for the WebLogic Administration Server.

- **External Options**

  These parameters are passed to the Java process, which connects to the Administration Server. All of these parameters must begin with -D.

- **JMX Protocol**

  The Management Agent makes a JMX connection to the Administration Server to discover the domain's members. Valid values are t3, t3s, iiop, and iiops. If you do not provide a protocol, the t3 default is used.

- **JML Service URL**

  Makes a JMX connection to the Administration Server. If you do not specify this option, it is created based on the input parameters.

- **Unique Domain Identifier**

  Creates a unique target name. This option can contain only alphanumeric characters and the special character '_' and cannot contain any other special characters.

- **Agent URL**

  URL for the Management Agent used to discover the targets. If you do not provide a value, the local Management Agent present on the target WebLogic Server is used. If a Management Agent is not found on the target WebLogic Server, an error is displayed.

- **Discover Down Servers**

  If this value is true, the servers that are down are discovered. If false, the servers that are down are not discovered.

- **Use Credential Store**

  If this value is set to true, the verb retrieves the WebLogic credentials from the credential store.

- **Discover App Versions**

  Optional - Default if not specified is true

  If only the active version of the app is needed, then set it to false.

- **Enable Refresh Job**

Optional - Default if not specified is false

If you would like to perform a daily job to refresh the domain discovery, then set this to true.

- **Use Host Name in Service URL**

  Optional - Default if not specified is false

- **Create Incident for Discovery Failure**

  Optional - Default if not specified is false

- **Node Manager Username**

  Optional - Default if not specified is Administration Server Username

  If you would like the Node Manager discovered as a target to be monitored, then username/password are required.

- **Node Manager Password**

  Optional - Default if not specified is Administration Server Password

Examples

This example reads the my_domains_info.csv file to determine the domains to be added to Cloud Control, and reads the my_agent_mapping.csv file to determine which Management Agents should monitor which host's managed servers.

```
emcli discover_wls
    -input_file=domain_discovery_file:\emcli\my_domains_info.csv
    -input_file=host_agent_mapping_file:\emcli\my_agent_mapping.csv
    -debug
```

This example manually redirects the output of discover_wls to a file using standard output redirect.

```
emcli discover_wls
input_file=domain_discovery_file:"<fully_qualified_path_of_domain_discovery_file/
domain_discovery_file.csv>" > /tmp/emcli/output_file.out
```

discover_wmq

Discovers an IBM Websphere MQ target in Oracle Enterprise Manager. IBM Websphere MQ targets include an individual IBM Websphere MQ target or IBM Websphere MQ clusters (for example, wmq_cluster or wmq).

Format

```
emcli discover_wmq
    -host
    -port
    -username
    -jarpath
    [-channel]
    -agent
    [-disc_pref]
    -password
    [-debug]

[ ]  indicates that the parameter is optional.
```

Options

- host

  Fully qualified name of the host or the IP address of the host on which the Webshpere MQ Queue Manager is running.

- port

  TCP/IP listener port of the Websphere MQ Queue Manager.

- username

  User name of the user who has access to the Websphere MQ Queue Manager.

- jarpath

  The absolute path of the required jar files wherein each of them should be separated by colon in case of Linux and semicolon in case of Windows.

- channel

  The name of the Server Connection Channel to be used for monitoring. The Server Connection Channel should be of type SVRCONN, for example: SYSTEM.DEF.SVRCONN.

- agent

  Monitoring Agent.

- disc_pref

  The value of discovery preference should be set to 'true' if the discovery has to be restricted to a single Websphere MQ Queue Manager. If it is not specified it is considered as false.

- password

  Websphere MQ password if security is enabled.

- debug

  If specified, runs the verb in verbose mode for debugging purposes.

Examples

Example 1

The following example discovers IBM Websphere MQ targets by specifying jarpath for Linux in Oracle Enterprise Manager.

```
emcli discover_wmq
  -host=host1.example.com
  -port=1416
  -username=mqm
-jarpath=/install/home/com.ibm.mq.commonservices.jar:/install/home/
com.ibm.mq.headers.jar:/install/home/com.ibm.mq.jar:/install/home/com.ibm.mq.jmqi.jar:/
install/home/com.ibm.mq.pcf.jar:/install/home/connector.jar
  -channel=SYSTEM.DEF.SVRCONN
  -agent=host1.example.com:12345
  -password=admin
```

Example 2

The following example discovers IBM Websphere MQ targets by specifying jarpath for Windows in Oracle Enterprise Manager.

```
emcli discover_wmq
  -host=host1.example.com
  -port=1416
  -username=mqm
-jarpath=/install/home/com.ibm.mq.commonservices.jar;/install/home/
com.ibm.mq.headers.jar;/install/home/com.ibm.mq.jar;/install/home/
com.ibm.mq.jmqi.jar;/install/home/com.ibm.mq.pcf.jar;/install/home/connector.jar
  -channel=SYSTEM.DEF.SVRCONN
  -agent=host1.example.com:12345
  -password=admin
```

discover_workloads

Describes the captures (and replays) contained in a directory.

Output Columns: Entity Name, Database Name, Start Time, Status.

Sample XML File:

```
<?xml version="1.0" encoding="UTF-8"?>
      <cliImportData xmlns="http://xmlns.oracle.com/sysman/db/dbreplay">
        <targetName>database</targetName>
        <targetType>oracle_database</targetType>
        <dbHostName>host.example.com</dbHostName>
        <dbCredRef>
          <credName>testDB121</credName>
          <credOwner>sysman</credOwner>
        </dbCredRef>
        <dbHostCredRef>
          <credName>testDBHost121</credName>
          <credOwner>sysman</credOwner>
        </dbHostCredRef>
        <directoryPath>/storage/dbr/copyDir_task4Caps</directoryPath>
        <consolidatedDirectory>true</consolidatedDirectory>
      </cliImportData>
```

Format - Standard Mode

```
emcli discover_workloads         [-input_file="template:<input file
path>"]         [-include_replays]         [-format="[name:<pretty|script|csv>];
[column_separator:"column_sep_string"];[row_separator:"row_sep_string"]"]
```

Format - Interactive or Script Mode

```
discover_workloads(         [input_file="template:<input file path>"]
[,include_replays=True/False]         [,format="[name:<pretty|script|csv>];
[column_separator:"column_sep_string"];
[row_separator:"row_sep_string"]"]         ]
```

Options

•   input_file

    Fully qualified path to an XML file containing parameters for the verb.

•   include_replays

    Describes the replays, in addition to the captures, that are contained in a directory.

- format

  Specifies how the output is formatted. The default value is "name:pretty". You can use this option in the following ways:

  – format="name:pretty" Prints the output table in a readable format not intended to be parsed by scripts.

  – format="name:script" Sets the default column separator to a tab and the default row separator to a newline in the output. You can override the column and row separator strings with your own values.

  – format="name:script;column_separator:<column_sep_string>" Causes the verb output to be column-separated by <column_sep_string>. Rows are separated by the newline character.

  – format="name:script;row_separator:<row_sep_string>" Causes the verb output to be row-separated by <row_sep_string>. Columns are separated by the tab character.

  – format="name:script;column_separator:<column_sep_string>;row_separator:<row_sep_string>" Causes the verb output to be column-separated by <column_sep_string> and row-separated by <row_sep_string>.

  – format="name:csv" Sets the default column separator to a comma and the default row separator to a newline in the output.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1 - Standard Mode

The following example describes the captures and replays contained in a directory.

```
emcli discover_workloads  -include_replays  -input_file=template:/storage/xml/
import4Captures.xml
```

Example 2 - Interactive or Script Mode

The following example describes the captures and replays contained in a directory.

```
discover_workloads(include_replays=True,   input_file="template:/storage/xml/
import4Captures.xml"  )
```

Example 3 - Standard Mode

The following example describes only the captures contained in a directory.

```
emcli discover_workloads  -input_file=template:/storage/xml/import4Captures.xml
```

Example 4 - Interactive or Script Mode

The following example describes only the captures contained in a directory.

```
discover_workloads(input_file="template:/storage/xml/import4Captures.xml"  )
```

See Also

import_replays
import_workloads

download_ats_test_databank_file

Downloads the specified databank file corresponding to the given ATS test. If no databank alias is specified, the command downloads all databanks for the test.

Format

```
emcli download_ats_test_databank_file
      -name=<target_name>
      -type=<target_type>
      -testname=<test_name>
      -testtype=<test_type>
      [-databankAlias=<databank_alias>]
      [-output_dir=<output_directory>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the target.

- **type**

  Name of the target type.

- **testname**

  Name of the test.

- **testtype**

  Type of test.

- **databankAlias**

  Databank alias.

- **output_dir**

  Output directory. If the directory does not exist, it is created.

Examples

Example 1

This example downloads the databank corresponding to alias1 for the specified test.

```
emcli download_ats_test_databank_file -name="Service Name"
                                      -type="generic_service"
                                      -testname="Test Name"
                                      -testtype="OATS"
                                      -databankAlias="alias1"
```

Example 2

This example downloads all databanks corresponding to the specified test.

```
emcli download_ats_test_databank_file -name="Service Name"
                                      -type="generic_service"
                                      -testname="Test Name"
                                      -testtype="OATS"
```

download_ats_test_zip

Downloads the zip bundle corresponding to the specified ATS test.

Format

```
emcli download_ats_test_zip
       -name=<target_name>
       -type=<target_type>
       -testname=<test_name>
       -testtype=<test_type>
       [-output_dir=<output_directory>]

[ ]  indicates that the parameter is optional
```

Options

* **name**

  Name of the target.

* **type**

  Name of the target type.

* **testname**

  Name of the test.

* **testtype**

  Type of test.

* **output_dir**

  Output directory. If the directory does not exist, it is created.

Examples

```
emcli download_ats_test_zip -name="Service_Name"
                            -type="Generic_Service"
                            -testname="Test_Name"
                            -testtype="OATS"
                            -output_dir="outputDirectory"
```

download_update

Downloads an update.

Format

```
emcli download_update
       -id="internal id"
```

Options

* **id**

  Internal identification for the update to be downloaded.

Examples

This example submits a job to download an update, and prints the job execution ID upon submission.

```
emcli download_update
        -id="914E3E0F9DB98DECE040E80A2C5233EB"
```

dump_activity_list

Prints the list of all current activities.

Format

```
emcli dump_activity_list
```

edit_dbprofile

Edits the schedule and purge policy of an existing database profile.

Format

```
emcli edit_dbprofile
      -comp_loc="Database Profile component location in software library"
        [-schedule=          [NONE]|[frequency:interval|weekly|monthly|yearly];
          start_time:yy-MM-dd HH:mm;
          end_time:yy-MM-dd HH:mm;
          [repeat:#m];
          [months:#,#,#,...];
          [days:#,#,#,...];
          [tz:{java timezone ID}];
          [grace_period:xxx];
        ]
        [-purge_policy= DAYS|SNAPSHOTS: number]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   comp_loc

    A combination of the database profile location and name.

*   schedule

    –   frequency: The frequency type with which the databse profile will be created. It can be an interval (in minutes), weekly, monthly, or yearly.

    –   start_time: Denotes the start time of Database Profile Component Creation in the format yy-MM-dd HH:mm.

    –   end_time: Denotes the end time of Database Profile Component Creation Repetition in the format yy-MM-dd HH:mm

    –   repeat: The repetition rate at which database profile will be created. If the frequency is an interval, then repeat is in minutes.

    –   months: The number of months after which the repetition of Database Profile Component Creation will occur.

- days: The number of days after which repetition of Database Profile Component Creation will occur.

- tz: The time zone ID, for example tz:America/New_York.

- grace_period: A period of time in minutes that defines the maximum permissible delay when attempting to create a databasse profile. If the job system cannot start the execution within a time period equal to the scheduled time plus the grace period, it will set the create database profile to be skipped. By default, the grace period is indefinite.

- purge_policy

  You can purge the collected data based on a specified number of days (DAYS) or a count of snapshots (SNAPSHOT). If the purge_policy parameter is not specified, then it is defaulted to NONE.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example edits the schedule and purge policy database profile RMAN_Profile with the location `Database Provisioning Profiles/11.2.0.4.0/linux_x64`.

```
emcli edit_dbprofile      -comp_loc="Database Provisioning Profiles/11.2.0.4.0/
linux_x64/RMAN_Profile"
    -schedule="frequency:interval;start_time:14-10-05 05:30;end_time:
        14-10-12 05:23;repeat:30;grace_period:60;tz:America/New_York"
    -purge_policy=DAYS:2
```

edit_sl_rule

Edits the service-level rule for the specified service.

Format

```
emcli edit_sl_rule
    -name="target name"
    -type="target type"
    [-expSL="expected service level value"]
    [-repeatSequence="days repeat sequence"]
    [-startTime="start time"]
    [-endTime="end time"]
    [-availStatesInclude="included availability states"]
    [-availStatesExclude="excluded availability states"]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Identifies the target name.

- **type**

  Identifies the target type. Use `emcli get_targets` to get the target type.

- **expSL**

Specifies the expected service-level rule. Values must be any number between 0 and 100.

- **repeatSequence**

  Specifies the days in which the service-level rule is to be applied. Identify the days value from these comma-separated values: MON, TUE, WED, THU, FRI, SAT, SUN.

- **startTime**

  Specifies the time of day that the application of the service-level rule is to begin. Enter the time format as: HH:min

- **endTime**

  Specifies the time of day that the application of the service-level rule is to end. Enter the time format as: HH:min

- **availStatesInclude**

  Specifies the availability states (apart from UP) that are to be included while computing the service-level rule. Values are: BLACKOUT|UNKNOWN

- **availStatesExclude**

  Specifies the availability states (apart from UP) that are to be excluded while computing the service-level rule. Values are: BLACKOUT|UNKNOWN

Examples

Example 1

Update the MyService service-level rule to begin at 6 a.m. on Mondays and Tuesdays:

```
emcli edit_sl_rule
     -name="MyService"
     -type="generic_service"
     -expSL="90.0"
     -repeatSequence="MON,TUE"
     -startTime="06:00"
     -endTime="23:00"
     -availStatesInclude="BLACKOUT"
     -availStatesExclude="UNKNOWN"
```

edit_storage_ceiling

Sets the storage ceiling for a storage pool.

Format

Standard Mode

```
emcli edit_storage_ceiling
     -storage_name="<storage name>"
     -pool_name="<pool name>"
     -storage_ceiling="<storage ceiling in GB>"
```

Interactive or Script Mode

```
edit_storage_ceiling(
     storage_name="<storage name>"
     ,pool_name="<pool name>"
```

```
          ,storage_ceiling="<storage ceiling in GB>"
          )
```

```
[ ]  indicates that the parameter is optional.
```

Options

- storage_name

  Name of the storage.

- pool_name

  Name of the storage pool.

- storage_ceiling

  Storage Ceiling to be set on the storage pool.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example sets the storage ceiling for a storage pool `sunzfs1`:

```
edit_storage_ceiling
              -storage_name="sunzfs1"
              -pool_name="pool-01"
              -storage_ceiling="100"
```

em_asr_asset_actcred

Adds or removes an activation credential for an Oracle Auto Service Request (ASR) target.

Format

Standard Mode

```
emcli em_asr_asset_actcred
      [-targetName="Target Name"]
      [-targetType="Target Type"]
      [-mosid="MOS user name"]
      [-passwd="MOS user password"]
```

Interactive or Script Mode

```
em_asr_asset_actcred(
      [,targetName="Target Name"]
      [,targetType="Target Type"]
      [,mosid="MOS user name"]
      [,passwd="MOS user password"]
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

- targetName

  Identifies the target name.

- targetType

  Identifies the target type.

- mosid

  Valid My Oracle Support (MOS) user name.

- passwd

  MOS user password.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

Example 1

Assigns the global activation credentials. User will be prompted to enter their MOS user password.

Standard Mode

```
emcli em_asr_asset_actcred
      -add -mosid="MOS_user_name"
```

Interactive or Script Mode

```
em_asr_asset_actcred(add=True,
      mosid="MOS_user_name"
      )
```

Example 2

Assigns the target activation credentials. User will be prompted to enter their MOS user password.

Standard Mode

```
emcli em_asr_asset_actcred
      -targetName="ASR_target1"
      -targetType="host"
      -mosid="MOS_user_name"
```

Interactive or Script Mode

```
em_asr_asset_actcred(
      targetName="ASR_target1",
      targetType="host",
      mosid="MOS_user_name"
      )
```

em_asr_asset_activation_details

Downloads a CSV file with activation details about all Oracle Auto Service Request (ASR) targets.

Format

Standard Mode

```
emcli em_asr_asset_activation_details
```

Interactive or Script Mode

```
em_asr_asset_activation_details()
```

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

em_asr_asset_activation_job

Manages the Oracle Auto Service Request (ASR) activation job. Use this verb to show the job status, start the job, or reschedule it.

> **Note:**
>
> Only the Enterprise Manager super user can perform this operation.

Format

Standard Mode

```
emcli em_asr_asset_activation_job
      [-start]
      [-rescheduleNow]
```

Interactive or Script Mode

```
em_asr_register(
      [start=True/False]
      [,rescheduleNow=True/False]
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   start

    Starts the activation job to run daily (including today) at the current time, if not started.

*   rescheduleNow

    Reschedules the activation job to run daily (including today) at the current time, if it is already scheduled.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

Example 1

Shows the status of the activation job:

Standard Mode

```
emcli em_asr_asset_activation_job
```

Interactive or Script Mode

```
em_asr_asset_activation_job()
```

Example 2

Starts an activation job, if not already scheduled. An error is returned if the activation job is already scheduled.

Standard Mode

```
emcli em_asr_asset_activation_job
    -start
```

Interactive or Script Mode

```
em_asr_asset_activation_job(
    start=True
    )
```

em_asr_asset_exclude_list

Adds or removes targets to or from the Oracle Auto Service Request (ASR) exclude list.

Format

Standard Mode

```
emcli em_asr_asset_exclude_list
    [-add]
    [-remove]
    [-all]
    [-targetName="Target Name"]
    [-targetType="Target Type"]
```

Interactive or Script Mode

```
em_asr_asset_exclude_list(
    [add=True/False]
    [,remove=True/False]
    [,all=True/False]
    [,targetName="Target Name"]
    [,targetType="Target Type"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

• add

    Adds ASR targets to the exclude list.

• remove

    Removes ASR targets from the exclude list.

• all

    Sets a flag to select all eligible ASR targets.

- targetName

  Identifies the target name.

- targetType

  Identifies the target type.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

Example 1

Adds a specific target to the Oracle ASR exclude list:

Standard Mode

```
emcli em_asr_asset_exclude_list
     -add
     -targetName="ASR_target1"
     -targetType="host"
```

Interactive or Script Mode

```
em_asr_asset_exclude_list(
     add=True,
     targetName="ASR_target1",
     targetType="host"
     )
```

Example 2

Removes all targets from the Oracle ASR exclude list:

Standard Mode

```
emcli em_asr_asset_exclude_list
     -remove
     -all
```

Interactive or Script Mode

```
em_asr_asset_exclude_list(
     remove=True,
     all=True
     )
```

em_asr_asset_include_list

Adds or removes target to or from the Oracle Auto Service Request (ASR) include list.

Format

Standard Mode

```
emcli em_asr_asset_include_list
     [-add]
     [-remove]
     [-all]
```

```
          [-targetName="Target Name"]
          [-targetType="Target Type"]
```

Interactive or Script Mode

```
em_asr_asset_include_list(
      [add=True/False]
      [,remove=True/False]
      [,all=True/False]
      [,targetName="Target Name"]
      [,targetType="Target Type"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

* add

   Adds targets to the Oracle ASR include list.

* remove

   Removes targets from the Oracle ASR include list.

* all

   Sets a flag to select all eligible ASR targets.

* targetName

   Identifies the target name.

* targetType

   Identifies the target type.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

Example 1

Adds all targets to the Oracle ASR include list:

Standard Mode

```
emcli em_asr_asset_include_list
      -add
      -all
```

Interactive or Script Mode

```
em_asr_asset_include_list(
      add=True,
      all=True
)
```

Example 2

Removes all targets from the Oracle ASR include list:

Standard Mode

```
emcli em_asr_asset_include_list
      -remove
      -all
```

Interactive or Script Mode

```
em_asr_asset_include_list(
      remove=True,
      all=True
      )
```

em_asr_deregister

Removes the Oracle Auto Service Request (ASR) registration and incident rules associated with Oracle ASR.

> **✎ Note:**
>
> Only the Enterprise Manager super user can perform this operation.

Format

Standard Mode

```
emcli em_asr_deregister
      [-all]
      -default
```

Interactive or Script Mode

```
em_asr_deregister(
      [all]
      ,default
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   all

    Removes all Oracle ASR user registration and incident rules associated with Oracle ASR.

*   default

    Removes only the Oracle ASR user registration associated with Oracle ASR.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

Example 1

Removes the Oracle ASR user registration and incident rules:

```
emcli em_asr_deregister
      -all
```

Example 2

Removes only the Oracle ASR user registration:

```
emcli em_asr_deregister
      -default
```

em_asr_register

Enables Oracle Auto Service Request (ASR) functionality in Enterprise Manager. My Oracle Support (MOS) credentials are used to activate any new asset discovered for ASR.

> **Note:**
>
> Only the Enterprise Manager super user can perform this operation.

Format

Standard Mode

```
emcli em_asr_register
      -mosid="My Oracle Support (MOS) user name"
      [-passwd = <My Oracle Support (MOS) user password>]
      [-dontStartJob = <Do not start activation job as part of registration>]
```

Interactive or Script Mode

```
em_asr_register(
      mosid="mosid"
      [,passwd = <My Oracle Support (MOS) user password>]
      [,dontStartJob = <Do not start activation job as part of registration>]
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

- mosid

  Valid My Oracle Support (MOS) user name.

- passwd

  MOS user password.

- dontStartJob

  Flag to tell if activation job should not be started.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

Example 1

Enables ASR and starts the activation job:

```
emcli em_asr_register
     -mosid="user1@123.com"
```

Example 2

Enables ASR but do not start the activation job:

```
emcli em_asr_register
     -mosid="user1@123.com"
     -dontStartJob
```

em_asr_xsl_upload

Uses an XSL file location and content to generate a create, update, or close service request (SR) message for Oracle Auto Service Request (ASR) as input and updates it to the OMS.

> **Note:**
>
> Only the Enterprise Manager super user can perform this operation.

Format

Standard Mode

```
emcli em_asr_xsl_upload
     -input_file="xsl_file:<full filepath>"
```

Interactive or Script Mode

```
em_asr_xsl_upload(
     input_file="xsl_file:<full filepath>"
     )
```

Options

- input_file

  The full path of the XSL template file.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

Example 1

Uploads an XSL file to the OMS:

```
emcli em_asr_xsl_upload
     -input_file="xsl_file:/tmp/scratch/myfile1.xsl"
```

emcli deploy_jvmd usage

Deploys JVMD agent in a WebLogic Domain.

Format

```
emcli deploy_jvmd
            -domain_name="Domain Target Name"
        [-engine="JVMD Engine URL"]
        [-clusters="Clusters Name"]
        [-servers="Servers Name"]
        [-host_cred="Oracle Home Owner Credentials"]
        [-wls_cred="WebLogic Administrator Credentials"]
        [-deploy_mode="Deploy Mode"]
        [-java_args="Client java arguments"]
        [-schedule=
                    start_time:yyyy/MM/dd HH:mm;
            [tz:{java timezone ID}];
            [grace_period:xxx];
        ]
        [-workDir="Working Directory Location"]
        [-libDir="Libraries Directory Location"]
```

```
[ ] indicates that the parameter is optional.
```

Options

*   -domain_name

    Name of the WebLogic Domain Target where the agent will be deployed.

*   -engine

    Either JVMD Engine URL or configured JVMD Load Balancer URL. You can see all the acceptable URLs list in the JVMD agent download dialog box.

*   -clusters

    Name of the cluster(s) the JVMD agent will be targeted.

*   -servers

    Name of the server(s) the JVMD agent will be targeted.

*   -host_cred

    Named credential used to access the reference host. This is an optional parameter.To pass the credential parameter, enter a name:value pair in the following format: `credential_name:credential_owner`.

    where,

    –   Credential_name is the name of the named credential.

    –   Credential_owner is the credentials of the Oracle home owner on the Administration Server host.

> **Note:**
>
> All the operations will be performed on the Administration Server host. If no named credential is provided, the preferred host credentials for the Oracle home target will be used.

- -wls_cred

  Named credential used to access the Administration Server. This is an optional parameter. To pass the credential parameter, enter a name:value pair in the following format: credential_name:credential_owner.

  where,

  - Credential_name is the name of the named credential.
  - Credential_owner is the credentials of the Oracle home owner on the Administration Server host.

> **Note:**
>
> If no named credential is provided, the preferred administrator credentials for the domain target will be used.

- -deploy_mode

  Deployment mode of the JVMD agent. This is an optional parameter. The valid modes are deploy, redeploy and undeploy. If no mode is provided, the 'deploy' mode will be used by default.

- -java_args

  Client java arguments that will be used to connect to the Administration Server. This is an optional parameter.

- -schedule

  Specify when to run the deployment procedure. If no value is entered, by default, the procedure runs immediately. To schedule a procedure, provide:

  - start_time: when the procedure should start
  - tz: the timezone ID
  - grace_period: grace period in minutes

- -workDir

  Specify the working directory to be used. This is an optional parameter.

- -libDir

  Default directory for extracting native libraries. This is an optional parameter.

Example 1

A JVMD agent is deployed in the domain base_domain at the specified schedule and targeted to two clusters. Also, since the credentials haven't been specified, the preferred credentials for the target are used.

```
emcli deploy_jvmd
      -domain_name="/Farm01_base_domain/base_domain"
```

```
                 -engine="protocol://myhost.mycompany.com:port"
                 -clusters="Cluster1,Cluster2"
                 -schedule="start_time:2016/6/21
21:23;tz:America/New_York;grace_period:60"
                 -workDir="/u01/mytemp"
```

emcli jvm_configchange

Propagates the mentioned configuration to all the JVM targets of the JVM Pool target mentioned or updates the given JVM targets.

In jvm_pools and jvms parameter options one of the parameter is mandatory. At least one of the configuration parameters in the dump directory, or log level, or bci enabled is required to be updated.

Format

```
emcli jvm_configchange
       [-jvm_pools=<JVM Pool Targets separated by ;>]
       [-jvms=<JVM Targets separated by ;>]
       [-heap_or_jrf_dump_directory=<Directory path>]
       [-log_level=<Log level>]
       [-bci_enabled=<true|false>]
       [-monitoring_enabled=<Y|N>]
       [-debug]
```

```
[ ] indicates that the parameter is optional.
```

Options

*   -jvm_pools

    One or more fully qualified JVM Pool Targets separated by a semi colon (;).

*   -jvms=

    One or more fully qualified JVM Targets separated by semi colon (;).

*   -heap_or_jrf_dump_directory=

    Fully qualified directory path.

*   -log_level=

    Log level options; 1-ERROR, 2-WARN, 3-INFO, 4-DEBUG, 5-TRACE, 6-ALL.

*   -bci_enabled=

    To enable or disable bytecode instrumentation options; true or false.

*   -monitoring_enabled=

    To enable or disable Jvm monitoring options; Y or N.

*   -debug

    Runs the verb in verbose mode for debugging purposes.

Example

Example 1

```
emcli jvm_configchange
     -jvm_pools="/EMGC_EMGC_DOMAIN/EMGC_DOMAIN_jvmpool"
     -jvms="/EMGC_EMGC_DOMAIN/EMGC_DOMAIN_jvmpool/Server1_jvm"
     -heap_or_jrf_dump_directory="/tmp/"
```

```
    -log_level="2"
    -bci_enabled="true"
    -monitoring_enabled="Y"
```

emcli jvmpool_configchange

Updates the mentioned configuration to all the JVM Pool targets mentioned.

At least one of the configuration parameter in the poll is enabled, poll interval or JVM pool name is required to be updated.

jvm_pools should have single value when jvmpool_name is having valid JVM pool name.

Format

```
emcli jvmpool_configchange
    -jvm_pools=<JVM Pool Targets separated by ;>
    [-poll_enabled=<true|false>]
    [-poll_interval=<Time in milli seconds>]
    [-jvmpool_name=<JVM Pool new name>]
    [-debug]
```

[ ] indicates that the parameter is optional.

Options

- -jvm_pools

  One or more fully qualified JVM Pool Targets separated by a semi colon (;).

- -poll_enabled=

  To enable or disable, polling value should be either true or false.

- -poll_interval=

  Value should be time in milli seconds.

- -jvmpool_name=

  New name of the JVM Pool which does not exists in system.

- -debug

  Runs the verb in verbose mode for debugging purposes.

Example

Example 1

```
emcli jvmpool_configchange
    -jvm_pools="/EMGC_EMGC_DOMAIN/EMGC_DOMAIN_jvmpool"
    -poll_enabled="true"
    -poll_interval="100000"
    -jvmpool_name="NewPoolName"
```

emcli scaledown_domain

Removes specified managed servers from a WebLogic Domain.

Format

```
emcli scaledown_domain
        -domain_target="Domain Target Name"
```

```
-server_list="Server List"
[-host_cred="Host Credentials"]
[-wls_cred="WebLogic Administrator Credentials"]
[-schedule=
        start_time:yyyy/MM/dd HH:mm;
        [tz:{java timezone ID}];
        [grace_period:xxx];
    ]
```
`[ ] indicates that the parameter is optional.`

Options

- -domain_target

  Name of the WebLogic domain target.

- -server_list

  Comma separated list of the managed servers names to be removed.

- -host_cred

  Named credential list for OS user that has write permission on the managed servers domain homes. To pass the credential parameter, enter a name:value pair in the following format: `credential_name:credential_owner.`

  where,

  – Credential_name is the name of the named credential.

  – Credential_owner is the credentials of the Oracle home owner on the Administration Server host.

  > **Note:**
  >
  > All the operations will be performed with the same credential if only one is provided, if each managed server requires a different credential, then a list of comma separated credentials will be matched against the list of managed servers. If no named credential is provided, the preferred host credentials will be used.

- -wls_cred

  Named credential used to access the Administration Server. This is an optional parameter. To pass the credential parameter, enter a name:value pair in the following format: `credential_name:credential_owner.`

  where,

  – Credential_name is the name of the named credential.

  – Credential_owner is the credentials of the Administrator of the WebLogic Domain.

  > **Note:**
  >
  > If no named credential is provided, the preferred administrator credentials for the domain target will be used.

- -schedule

Specify when to run the deployment procedure. If no value is entered, by default, the procedure runs immediately. To schedule a procedure, provide:

– start_time: when the procedure should start

– tz: the timezone ID

– grace_period: grace period in minutes

Example 1

A single Managed Server named mServer needs to be removed from the WebLogic Domain target /Farm01_base_domain/base_domain at the specified schedule. Since the Managed Server host credentials haven't been specified, the preferred credentials for the target host and domain target are used.

```
        emcli scaledown_domain
            -domain_target="/Farm01_base_domain/base_domain"
            -server_list="mServer"
            -schedule="start_time:2014/6/21
21:23;tz:America/New_York;grace_period:60"
```

emcli_tde

Performs TDE operations on a given target database.

Format

```
emcli tde
    -target_name="target_name"
    -target_type="target_type"
    -operation="operation"
    -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"DBNamedCreds:unamedcreds" or "DBCredsOwner:ucredsowner;DBNamedCreds:unamedcreds"
    [-wallet_password="wallet_password"]
    [-wallet_new_password="new_wallet_password"]
    [-wallet_location="wallet_location"]
    [-keystore_type ="HSM"]
    [-keystore_status ="open"]
    [-tablespace="tablespace"]
    [-backup_id="new_backup_101"]
    [-backup_location="\scratch\oracle\wallet\"]
    [-algorithm="algorithm"]
    [-master_key_tag="NEW MASTER KEY"]
    [-keep="yes/no" default "no" ]
    [-encrypted="yes/no" default "no" ]
    [-validate_only="yes/no" default "no"]
[ ]  indicates that the parameter is optional.
```

Options

• target_name

  The name of the target.

• target_type

  The type of the target. The default value for this argument is oracle_database. The possible values are: oracle_database and rac_database.

• operation

The name of the tde operation. It is case sensitive. The following operations are supported in tde: rekey, open_keystore, close_keystore, make_tablespace_offline, make_tablespace_online, encrypt_tablespace_online, decrypt_tablespace_online, rekey_tablespace_online, finish_tablespace_online, encrypt_tablespace_offline, and decrypt_tablespace_offline.

- master_key_tag

From 12.1 database version, the administrator can provide tag to identify the master key with rekey operation. This is an optional parameter, by default "NEW MASTER KEY" would be used as the TAG.

- connect_as

This option enables the user to change the password of a different user without logging in as that user. It should be specified in any of following formats: `--"DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]"` and `--"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"`. If this is not specified, the password will be changed by self.

- wallet_password

In the context of rekey, close, and open operation, wallet password is required to perform the TDE wallet management operation.

- wallet_new_password

In the context of change password operation, new wallet password is required to change password.

- wallet_location

In the context of TDE wallet management operation, wallet location is required to identify the wallet.

- keystore_type

In the context of list_databases operation, keystore_type option can be used to filter the result.

- keystore_status

In the context of list_databases operation, keystore_status option can be used to filter the result.

- backup_location

In the context of TDE keystore backup operation, backup keystore location is optional, if this input is not provided, then the current keystore location would be used to store the backed up keystore file.

- backup_id

In the context of TDE keystore backup operation, backup keystore id is optional, if this input is not provided, then the current server timestamp would be used as the backup id.

- tablespace

In the context of TDE operation, Tablespace name is required to perform encrypt, decrypt, and rekey on given tablespace name.

- algorithm

In the context of TDE operation, the encryption algorithm is required to perform encryption and rekey on the given tablespace.

- keep

  Indicates whether to keep the old datafiles post tablespace encryption/decryption/rekey operation. The possible values:

  – Yes- Keep the old datafiles.

  – No- do not keep the old datafiles.

  The default value of this option is NO.

  > **Note:**
  >
  > On execution (Wallet Management operations: rekey, open, and close), you will be prompted to enter the wallet_password in non-echo mode.

  This option hides the password to be displayed on command line.

Examples

Example 1

The following example tries to close the TDE wallet target tdeEnabledDatabase:

```
emcli tde
    -target_name=tdeEnabledDatabase
    -target_type=oracle_database
    -operation=close
    -wallet_password=Testing_1234
    -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 2

The following example tries to open the TDE wallet target tdeEnabledDatabase:

```
emcli tde
    -target_name=tdeEnabledDatabase
    -target_type=oracle_database
    -operation=open
    -wallet_password=Testing_1234
    -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 3

The following example tries to rekey the master key on target tdeEnabledDatabase:

```
emcli tde
    -target_name=tdeEnabledDatabase
    -target_type=oracle_database
    -operation=rekey
    -master_key_tag="new_rekey_101"
    -wallet_password=Testing_1234
    -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 4

The following example tries to backup the keystore. Backup ID and location are optional arguments. By default the backup location of the keystore would be in the current keystore location and the backup id would be the current server timestamp. The backup_keystore operation is supported starting from 12.1 database version.

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=backup_keystore
   -backup_id="new_backup_101"
   -backup_location="\scratch\oracle\wallet\"
   -wallet_password=Testing_1234
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 5

The following example tries to change keystore password. The change_keystore_password operation is supported starting from 12.1 database version.

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=change_keystore_password
   -wallet_password=Testing_1234
   -wallet_new_password=Testing#4321
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 6

The following example fetch keystore details for the given target.

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=keystore_details
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 7

The following example make the tablespace go online on the target tdeEnabledDatabase:

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=make_tablespace_online
   -tablespace=EXAMPLE_TS_1
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 8

The following example encrypt tablespace using online feature on the target tdeEnabledDatabase:

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
```

```
   -operation=encrypt_tablespace_online
   -tablespace=EXAMPLE_TS_1
   -algorithm=AES256
   -keep=yes
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 9

The following example rekey tablespace using online feature on the target
tdeEnabledDatabase:

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=rekey_tablespace_online
   -tablespace=EXAMPLE_TS_1
   -algorithm=AES192
   -keep=yes
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 10

The following example decrypt tablespace using online feature on the target
tdeEnabledDatabase:

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=decrypt_tablespace_online
   -tablespace=EXAMPLE_TS_1
   -keep=yes
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 11

The following example uses finish operation to finish the previously run online tablespace
operation that has run into issues:

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=finish_tablespace_online
   -tablespace=EXAMPLE_TS_1
   -algorithm=AES192
   -keep=yes
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

> **Note:**
>
> Offline feature is supported only on target database version 11.2.0.4, 12.1, and
> 12.2.

Example 12

The following example make the tablespace go offline on the target tdeEnabledDatabase:

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=make_tablespace_offline
   -tablespace=EXAMPLE_TS_1
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 13

The following example encrypt tablespace using offline feature on the target
tdeEnabledDatabase.

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=encrypt_tablespace_offline
   -tablespace=EXAMPLE_TS_1
   -algorithm=AES256
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 14

The following example decrypt tablespace using offline feature on the target
tdeEnabledDatabase:

```
emcli tde
   -target_name=tdeEnabledDatabase
   -target_type=oracle_database
   -operation=decrypt_tablespace_offline
   -tablespace=EXAMPLE_TS_1
   -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 15

The following example list tablespaces for a given target.

```
emcli tde
  -target_name=tdeEnabledDatabase
  -target_type=oracle_database
  -operation=list_tablespaces
  -encrypted=yes/no
  -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
```

Example 16

The following example list databases with TDE details. Target name, target type,
keystore type, and keystore status are the optional filter arguments that can be used to
further refine the result.

```
emcli tde
   -target_name=TargetDBName
   -target_type=oracle_database
   -operation=list_databases
   -keystore_type=HSM/WALLET/OKV
   -keystore_status=open/closed/unknown/not_configured
```

enable_audit

Enables auditing for ALL and BASIC user operations. For other operations, see the update_audit_settings verb.

Format

```
emcli enable_audit
      [-level=basic]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **level=basic**

    Enables auditing for BASIC user operations.

Examples

Example 1

This example enables auditing for all operations.

```
emcli enable_audit
```

Example 2

This example enables auditing for LOGIN, LOGOUT, DB_LOGIN, and DB_LOGOUT.

```
emcli enable_audit -level=basic
```

enable_config_history

Enables configuration history computation for a target type.

Format

```
emcli enable_config_history -target_type="{target type|'*'}"
```

Options

*   **target_type**

    Target type for which the configuration history is being enabled. The value should be the internal name or "*" to indicate all target types.

Examples

Example 1

This example enables configuration history computation for the host target type.

```
emcli enable_config_history -target_type="host"
```

Example 2

This example enables configuration history computation for all target types.

```
emcli enable_config_history -target_type="*"
```

enable_forwarder_agents

Takes a list of agents and marks each agent as a forwarder agent.

Format

```
emcli enable_forwarder_agents
      -agent_list="agent_list"
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   agent_list

    List of agents that need to be registered as forwarders. The agents must be separated by space.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example enables agent1 and agent2 as forwarding agents.

```
emcli enable_forwarder_agents
      -agent_list="agent1 agent2..."
```

enable_mda_finding_types_for_targets

Enables the specified MDA finding types for the specified targets. The finding types are enabled for a target only if the finding is applicable for the specified target type, and the EM CLI user has permissions on the target.

Format

```
emcli enable_mda_finding_types_for_targets
      [-targets="<list of targets and their target types >"]
      [-finding_types="<list of finding types>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   targets

    List of targets and their target types.

*   finding_types

    List of finding types.

Examples

Example 1

The following example enables a single finding type on a single target:

```
emcli enable_mda_finding_types_for_targets
      -finding_types="oracle.sysman.emas.wls_gc_overhead"
      -targets="/EMGC_EMGC_DOMAIN/EMGC_DOMAIN/EMGC_OMS1:weblogic_j2eeserver"
```

Example 2

The following command enables multiple finding types on multiple targets:

```
emcli enable_mda_finding_types_for_targets
      -
finding_types="oracle.sysman.emas.wls_gc_overhead;oracle.sysman.emas.wls_heap_config"
     -targets="/EMGC_EMGC_DOMAIN/EMGC_DOMAIN/EMGC_OMS1:weblogic_j2eeserver;/
test_base_domain/base_domain/MS1:weblogic_j2eeserver"
```

enable_metric_data_load

Reenables the metric data loading for targets or metrics where data loading was disabled due to a quarantine process.

This verb supports bulk operation for the following cases:

- All metrics of specified targets for a specified target type
- Specified metrics of all targets for a specified target type

Format

```
emcli enable_metric_data_load
      [-target_type="host"]
      [-targets="host1;host2;"]
      [-metrics="Load;Filesystems;"]

[ ]  indicates that the parameter is optional.
```

Options

- -target_type

  Target type that you are reenabling metric data loading for. If you want to perform the operation on all targets, then skip this option.

- -targets

  Semicolon separated list of targets for a specified target type. You must use the target_type option with this option. If you want to perform the operation on all targets for a specified target type, then skip this option.

- -metrics

  Semicolon separated list of metrics for a specified target type. You must use the target_type option with this option. If you want to perform the operation on all metrics for the specified target type, then skip this option.

Examples

Example 1

The following command reenables the metric data loading for all metrics of all the host targets.

```
 emcli enable_metric_data_load
      -target_type="host"
```

Example 2

The following command reenables the metric data loading for all metrics of the myhost1.example.com and myhost2.example.com host targets.

```
 emcli enable_metric_data_load
         -target_type="host"
         -targets="myhost1.example.com;myhost2.example.com;"
```

enable_or_disable_event_correlation_rule

Enables or disables an event correlation rule.

Format

```
enable_or_disable_event_correlation_rule
    -rule_name="event correlation rule name"
    -enable=true/false
```

Options

- rule_name

  Event correlation rule name.

- enable

  Enable a rule by setting the value to *true* or disable the rule by setting the value to *false*.

Example

The following command enables the event correlation rule *add_event_to_incident*.

```
enable_or_disable_event_correlation_rule
      -rule_name="add_event_to_incident"      -enable=true
```

enable_sla

Enables an SLA for a target.

Format

```
emcli enable_sla
    -targetName=<target_name>
    -targetType=<target_type>
    -slaName=<SLA_name>
    [-now]
    [-versionStart=<MM/dd/yyyy hh:mm a>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **targetName**

  Name of the target.

- **targetType**

  Type of target.

- **slaName**

  Name of the SLA.

- **now**

  Enables the SLA now, or uses versionStart for a specific time.

- **versionStart**

  Specifies when the computation of the SLA should start.

Examples

Example 1

This example immediately enables an SLA named 'gold_sla' for target my_service (generic_service).

```
emcli enable_sla
      -targetName='my_service' -targetType='generic_service'
      -slaName='gold_sla' -versionNum=2 -now
```

Example 2

This example enables a SLA named 'gold_sla' for target my_service (generic_service). It becomes active and starts computing at '09/23/2012 3:30 PM'.

```
emcli enable_sla
      -targetName='my_service' -targetType='generic_service'
      -slaName='gold_sla' -versionNum=2 -versionStart='09/23/2012 3:30 PM'
```

enable_snapclone

Enables the Snap Clone feature for a database.

Format

Standard Mode

```
emcli enable_snapclone
      -db_name="<database name>"
```

Interactive or Script Mode

```
enable_snapclone(
      db_name="<database name>"
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

- db_name

  Name of the database.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example enables the Snap Clone feature for the database `testmstr`:

```
emcli enable_snapclone
            -db_name="testmstr"
```

enable_target

Enables the target on both the Management Repository and Management Agent side.

Syntax

```
 emcli enable_target
        -type="target_type1"
        -name="target_name1"
        -agent="agent_name1"
        [-ignore_invalid_target]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   -type=target_type1

    Target type of the target being enabled.

*   -name=target_name1

    Name of the target. You can use the percentage character (%) as a wild character to enable all targets of a specified type on a specified Management Agent.

*   -agent=agent_name1

    Name of the Management Agent on which the target has to be enabled.

*   -ignore_invalid_target

    When specified, the process ignores invalid targets.

Examples

Example 1

The following command enables the target on the Management Agent and OMS and raises an error if the target is invalid.

```
emcli enable_target
    -type="oracle_em_service"
    -name="TestService"
    -agent="TestAgent"
```

Example 2

The following command enables all the targets of this type on the "TestAgent" Management Agent both on the OMS and Management Agent side. It ignores invalid targets.

```
emcli enable_target
    -type="oracle_em_service"
    -name="%"
```

```
                    -agent="TestAgent"
                    -ignore_invalid_target
```

enable_test

Enables monitoring of a Services test. It pushes the Service test collection to all the beacons.

Format

```
emcli enable_test
      -name=<target_name>
      -type=<target_type>
      -testname=<test_name>
      -testtype=<test_type>
```

Options

*   **name**

    Service target name.

*   **type**

    Service target type.

*   **testname**

    Test name.

*   **testtype**

    Test type.

Examples

This example enables the HTTP test named `MyTest` for the `generic_service` target named `MyTarget`.

```
emcli enable_test -name='MyTarget' -type='generic_service'
      -testname='MyTest' -testtype='HTTP'
```

execute_hostcmd

Executes a host command across a set of targets.

Format

```
emcli execute_hostcmd
      -cmd=<host_command"
      -osscript=<script_to_be_executed>
      -targets=<name1:type1;name2:type2;...>
      -credential_set_name=<name>
      [-input_file=<parameter_tag:script_file>]

[ ]  indicates that the parameter is optional
```

Options

*   **cmd**

    Host_command can be any valid host command or group of host commands.

- **osscript**

  OS script to be executed with the cmd parameter.

- **targets**

  List of target-name, target-type pairs. The host command is executed across this list of Enterprise Manager targets. All targets must be of the type `host` or `composite`, which represents a group of targets. If it is a group, the group is expanded to extract all the host targets, and the host command is executed across these host targets.

- **credential_set_name**

  The `credential_set_name` parameter refers to the set name of the preferred credentials stored in the Enterprise Manager repository. If this option is not present, `HostCredsNormal` is used for executing host commands. For the `host` target type, two credential sets exist:

  – `HostCredsNormal` — Default unprivileged credential set for a host target

  – `HostCredsPriv` — Privileged credential set for a host target

    The credential set parameter can only be specified when the override credential parameters such as `username` and `password` are not present.

    If provided, the you must fully specify the override credential parameters. For host command, `username` and `password` must be specified together.

- **input_file**

  Used in conjunction with `-osscript` , this enables you to load the contents of an OS script. The `-input_file` specifies a mapping between a tag and a local file path. The tag is specified in lieu of actual osscript contents of the `-osscript` . The tag must not contain colons (:) or semi-colons (;).

  For information about the input_file parameter, see -input_file Syntax Guidelines.

Examples

Example 1

This example executes the host command `ls -l;` against the target `stach.example.com:host` and host targets contained in the group `grp`. The stored `HostCredsPriv` preferred credentials are used for all the targets.

```
emcli execute_hostcmd
      -cmd="ls -l;"
      -credential_set_name="HostCredsPriv"
      -targets="stach.example.com:host;grp:composite"
```

Example 2

This example loads the contents of the script `/scratch/dba_scripts/shellscript.sh` into the value of `-osscript` and executes it against target `reference.example.com:host` and host targets contained in the group `grp`. The stored `HostCredsNormal` preferred credentials are used for all the targets.

```
emcli execute_hostcmd
      -cmd="/bin/sh -s"
      -osscript="FILE"
      -input_file="FILE:/scratch/dba_scripts/shellscript.sh"
```

```
                      -credential_set_name="HostCredsNormal"
                      -targets="reference.example.com:host;grp:composite"
```

execute_sql

Executes a SQL command across a set of targets.

Format

```
emcli execute_sql
      -sql=<sql_command>
      -targets=<name1:type1;name2:type2;...>
      -credential_set_name=<name>
      [-input_file=<parameter_tag:script_file>]

[ ]  indicates that the parameter is optional
```

Options

- **sql**

  "sql command" is a single SQL statement.

- **targets**

  List of target-name, target-type pairs. The SQL command executes across this list of Enterprise Manager targets. All targets must be of the type `oracle_database` or `composite`, which represents a group of targets. If it is a group, the group expands to extract all the database targets, and the SQL command is executed across these database targets.

- **credential_set_name**

  Refers to the set name of the preferred credentials stored in the Enterprise Manager repository. If this option is not present, the `DBCredsNormal` and `DBHostCreds` credential set is used for executing SQL commands. For each target type, several credential sets exist:

  – HostCredsNormal — Default unprivileged credential set for a host target

  – HostCredsPriv — Privileged credential set for a host target

  – DBHostCreds — Host credential set for an `oracle_database` target

  – DBCredsNormal — Default normal credential set for an oracle_database target

  – DBCredsSYSDBA — sysdba credential set for an `oracle_database` target

  You can only specify the `credential_set_name` parameter when the override credential parameters such as `[db_|host_]username` and `[db_|host_]password` are not present. If provided, the override credential parameters must be specified fully. For the SQL commands, `db_username`, `db_password`, `db_role`, `host_username`, and `host_password` must be present.

- **input_file**

  Used in conjunction with the `-sql` option, this option enables you to load the contents of a SQL script. The `-input_file` option specifies a mapping between a tag and a local file path. The tag is specified in lieu of an actual SQL command for the `-sql`. The tag must not contain colons (:) or semi-colons (;).

  For information about the input_file parameter, see -input_file Syntax Guidelines.

Examples

Example 1

This example executes the SQL command `select * from sysman.mgmt_targets;` against the target `database:oracle_database` and database targets contained in the group `grp`. The stored SYSDBA preferred credentials are used for all the targets.

```
emcli execute_sql
     -sql="select * from sysman.mgmt_targets;"
     -credential_set_name="DBCredsSYSDBA"
     -targets="database:oracle_database;grp:composite"
```

Example 2

This example loads the contents of the script `/scratch/dba_scripts/enterprise_schema.sql` into the value of `-sql`, and executes it against target `database:oracle_database` and database targets contained in the group `grp`. The stored SYSDBA preferred credentials are used for all the targets.

```
emcli execute_sql
     -sql="FILE"
     -input_file="FILE:/scratch/dba_scripts/enterprise_schema.sql"
     -credential_set_name="DBCredsSYSDBA"
     -targets="database:oracle_database;grp:composite"
```

export_adm

Exports an Application Data Model to the specified directory with the specified file name.

Format

```
emcli export_adm
     -adm_name=<application_data_model_name>
     [-directory=<directory_path>]
     [-file_name=<file_name>]

[ ]  indicates that the parameter is optional
```

Options

- **adm_name**

  Application data name that will be exported.

- **directory**

  Directory where the Application Data Model is to be exported. If the directory is not specified, the file is saved in the current directory.

- **file_name**

  Name of the file where the Application Data Model will be exported. If the file name is not specified, the default file name is the same as the specified Application Data Model name. If the file name does not have an extension, '.xml' is the default extension.

Output

Success/error messages.

Examples

Example 1

This example exports the Application Data Model Sample_ADM to the sample_adm.xml file.

```
emcli export_adm
        -directory=/home/user
        -adm_name=Sample_ADM
        -file_name=sample_adm.xml
```

export_admin_group

Exports the Administration group hierarchy.

Format - Standard Mode

```
emcli export_admin_group
```

Format - Interactive or Script Mode

```
export_admin_group()
```

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example creates an admin group with one level - Lifecycle status (with all 5 values).

```
ade:[ system_23 ] [system@sys01apq emgc]$ emcli export_admin_group
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AdminGroup>
    <groupList>
        <parent>ADMGRP0</parent>
        <child>Deve-Grp</child>
        <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:Development</propertyValuePair>
    </groupList>
    <groupList>
        <parent>ADMGRP0</parent>
        <child>MC-Grp</child>
        <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:MissionCritical</propertyValuePair>
    </groupList>
    <groupList>
        <parent>ADMGRP0</parent>
        <child>Prod-Grp</child>
        <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:Production</propertyValuePair>
    </groupList>
    <groupList>
        <parent>ADMGRP0</parent>
        <child>Stag-Grp</child>
```

```
            <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:Stage</propertyValuePair>
    </groupList>
    <groupList>
        <parent>ADMGRP0</parent>
        <child>Test-Grp</child>
        <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:Test</propertyValuePair>
    </groupList>
    <levelList>
        <levelNumber>1</levelNumber>
<levelProperty>orcl_gtp_lifecycle_status</levelProperty>
        <propertyValues>Development</propertyValues>
        <propertyValues>MissionCritical</propertyValues>
        <propertyValues>Production</propertyValues>
        <propertyValues>Stage</propertyValues>
        <propertyValues>Test</propertyValues>
    </levelList>
    <rootNode>ADMGRP0</rootNode>
</AdminGroup>
emcli awrwh_add_src_db -target_name=sample_database -target_type=oracle_database
```

export_charge_plans

Exports charge plan metadata to an XML file.

Format

```
emcli export_charge_plans
        [-charge_plan="plan_name"[-entity_type = chargeback_entity_type]]
        [-start_date=ddmmyyyy]
        -file=file_name
```

```
[ ]  indicates that the parameter is optional
```

Options

* **charge_plan**

  Name of the charge plan to be exported. If this option is not specified, all charge plan metadata is exported.

* **entity_type**

  Name of the Chargeback entity type whose charge plan is to be exported. If this option is not specified, all entity type charge rates in the charge plan are exported.

* **start_date**

  Start date of the report cycle whose charge plan metadata is to be exported. The start date value must be in ddmmyyyy format. If this option is not specified, the start date of the current report cycle is used.

* **file**

  Absolute path to which to export the metadata.

Examples

Example 1

This example exports metadata of the host entity type associated with charge plan Plan A, if active in the current report cycle, to the file /home/plans.xml:

```
emcli export_charge_plans
        -charge_plan="Plan A"
        -entity_type=host
        -file=/home/plans.xml
```

Example 2

This example exports metadata of charge plan Plan A, if active in the report cycle starting on 01062014, to the file /home/plans.xml:

```
emcli export_charge_plans
        -charge_plan="Plan A"
        -start_date=01062014
        -file=/home/plans.xml
```

export_compliance_group

Exports a compliance group definition and all of its element definitions given the name, author, and version.

Format

```
emcli export_compliance_group
        -name=<name>
        -author=<author>
        -version=<name>
        -output_file=<file>
```

Options

- **name**

  Name of the group to be exported.

- **author**

  Author of the group to be exported.

- **version**

  Version of the group to be exported.

- **output_file**

  Name of the exported file.

Examples

Example 1

```
emcli export_compliance_group \
      -name="foo" \
      -author="Jonas" \
      -version="99" \
      -output_file="$HOME/reports/group.xml"
```

export_compliance_standard_rule

Exports a rule to the specified files.

Format

```
export_compliance_standard_rule
  -name=<name>
  -target_type=<target_type>
  -output_file=<file>
```

Options

- **name**

  Name of the rule to be exported.

- **target_type**

  Target type of the rule to be exported.

- **output_file**

  Name of the exported file.

Examples

Example 1

```
emcli export_compliance_standard_rule \
     -name="foo" \
     -target_type="weblogic_j2eeserver" \
     -output_file="$HOME/reports/rule.xml"
```

export_config_compare_result

Exports the comparison result to a file.

Format

```
emcli export_config_compare_result
     -compare_check_name="<check_name>"
     -filename="<path_to_zip_file>"
     [-result_format="<S|L>"]
     [-content="<D|A>"]

[ ]  indicates that the parameter is optional.
```

Options

- compare_check_name

  Name of the comparison check.

- filename

  Output zip file with the absolute path.

- result_format

  Providing "S" exports the results in side-by-side format. Providing "L" exports the results in list format. Default value is "S".

- content

  Exports only differences if the value is "D". Exports all results if the value is "A". Default value is "D".

Examples

Example 1

The following example exports the results of comparison check with name "test check" to the file "TestReport.zip". Exports differences only, in side-by-side format.

```
emcli export_config_compare_result
     -compare_check_name="test check"
     -filename="/tmp/TestReport.zip"
```

Example 2

The following example exports the results of comparison check with name "test check" to the file "TestReport.zip" . Exports differences only, in list format.

```
emcli export_config_compare_result
     -compare_check_name="test check"
     -filename="/tmp/TestReport.zip"
     -result_format="L"
```

export_config_onetimecompare

Exports the saved one-time comparison result from the repository to a zip file.

Format

```
emcli export_config_onetimecompare
     -filename="<name_of_output_zip_file>"
     -name="<comparison_name>"
     [-result_format="<S|L>"]
     [-content="<D|A>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   filename

    Output zip file with absolute path.

*   name

    Name of the saved one-time comparison.

*   result_format

    Providing "S" exports the results in side-by-side format. Providing "L" exports the results in list format. Default value is "S".

*   content

    Exports only differences if the value is "D". Exports all results if the value is "A". Default value is "D".

Example

The following example exports the saved one-time comparison result from the repository to the "/tmp/TestReport.zip" output file with the name "cmp_host1".

```
emcli export_config_onetimecompare
      -filename="/tmp/TestReport.zip"
      -name="cmp_host1"
```

export_config_search

Exports the saved configuration search to an XML file.

Format

```
emcli export_config_search
      -name="<Configuration Search UI Name>"
      -output_file="<XML file name>"
```

Options

*   name

    Display name of the configuration search. It is a mandatory parameter.

*   output_file

    The name of the xml file with an absolute path. It is a mandatory parameter.

Example

The following example exports the configuration to the testSearch.xml file:

```
emcli export_config_search      -name="testSearch"
      -output_file="/tmp/testSearch.xml"
```

export_config_compare_template

Exports the comparison template to a file with the name provided as the argument to "filename".

Format

```
emcli export_config_compare_template
      -template_id="<id_of_template>"
      -filename="<name_of_xml_file>"
```

Options

*   template_id

    Identifier of the comparison template to be exported.

*   filename

    Output XML file with absolute path.

Example

The following example exports the comparison template with the identifier "1" to the file "TestHost.xml".

```
emcli export_config_compare_template
      -template_id="1"
      -output_file="/tmp/TestHost.xml"
```

export_custom_charge_items

Exports user-defined charge item metadata to the specified XML file.

Format

```
emcli export_custom_charge_items
        -entity_type="entity_type"
        -file=output_file
```

Options

- **entity_type**

  Name of the entity whose charge item metadata you want to export.

- **file**

  Full path of the file to which to write user-defined charge items associated with the specified entity type.

Examples

This example writes user-defined charge item metadata associated with the host entity type to the myhost.xml file in the home directory:

```
emcli export_custom_charge_items
        -entity_type="host"
        -file=/home/myhost.xml
```

export_facet

Exports a real-time monitoring facet to the specified file.

Format

```
emcli  export_facet
        -name="<name>"
        -target_type="<target_type>"
        -entity_type="<entity_type>"
        -output_file="<file>"
```

Options

- Name

  Name (internal) of the facet to be exported.

- target_type

  Target type (internal) of the facet to be exported. To see all target types available for your environment, check the $AGENT_HOME/sysman/admin/metadata directory. A metadata file (XML file) exists for each target type.

- entity_type

  Entity type (internal) of the facet to be exported, for example, osfile, osprocess, osuser, and so on.

- output_file

Name of the exported file.

Example

The following example exports the foo facet to the `$HOME/reports/facet.xml` file.

```
emcli export_facet
     -name="foo"
     -target_type="host"
     -entity_type="osfile"
     -output_file="$HOME/reports/facet.xml"
```

export_incident_rule_set

Exports an incident rule set from list of enterprise rule set(s).

> **Note:**
>
> Oracle-supplied out-of-box rule sets cannot be exported.

Privilege Requirements

Any user can export an enterprise rule set.

Format

```
emcli export_incident_rule_set
     -rule_set_name=<rule set name>
     [-rule_set_owner=<ruleset owner>]
     -export_file=<export file >

[ ]  indicates that the parameter is optional.
```

Options

*   rule_set_name

    Name of an enterprise rule set.

*   rule_set_owner (Optional)

    The owner of the rule set.

*   export_file

    XML file name along with the file path for the exported rule set.

    If the filename is specified within directory, this option will create the file with the specified rule set name in that directory.

Example

This command exports an enterprise rule set named *TEST_RULESET* owned by *sysman* to an XML file (TEST_RULESET.xml) located in the `tmp` directory (`/tmp/TEST_RULESET.xml`).

```
emcli export_incident_rule_set -rule_set_name=TEST_RULESET -
rule_set_owner=sysman -export_file="/tmp/"
```

export_jobs

Exports all matching job definitions in Enterprise Manager, including Corrective Actions. System jobs and nested jobs are excluded.

Format

```
emcli export_jobs
   -export_file=<zip_file_name>"
   [-name="job_name1;job_name2;..."]
   [-type="job_type1;job_type2;..."]
   [-targets="tname1:ttype1;tname2:ttype2;..."]
   [-owner="owner1;owner2;..."]
   [-preview]

[ ]  indicates that the parameter is optional
```

Options

- **export_file**

  Zip file name to be created.

- **name**

  Job name pattern to be used for filtering. Semicolon-separated job names can be provided. When filtering by a single value, wildcard char(% or _) can also be used. Wildcard "%"matches one or more characters. "_"(underscore) matches exactly one character.

- **type**

  Job type pattern to be used for filtering. Semicolon-separated job types can be provided. When filtering by sa ingle value, wildcard chars(% or _) can be used.

- **targets**

  Target name, type pattern to be used for filtering. Semicolon-separated target names and types can be provided. When filtering by a single value, wildcard chars(% or _) can be used.

- **owner**

  Owner of the jobs to be used for filtering. Semicolon-separated job owners can be provided.

- **preview**

  Jobs in the Enterprise Manager site matching the filter criteria are shown to stdout. Jobs are not exported to any file.

Output Columns

Success/Error messages.

Examples

Example 1

This example exports job definitions for jobs MYJOB1 and MYJOB2 to job_data.zip:

```
emcli export_jobs -name=MYJOB1;MYJOB2 -export_file=jobsdata.zip
```

Example 2

This example exports job definitions for any jobs owned by user name starting with ADMIN.

```
emcli export_jobs -owner=ADMIN% -export_file=jobsdata.zip
```

export_latest_config

Exports the latest configuration to a file with name provided as an argument "output_file".

Format

```
emcli export_latest_config
      -target_type="host"
      -target_name="test_host"
      [-export_members="true"]
      [-output_file="ExportConfig.zip"]
```

Options

- target_type

  Target type where the configuration is saved. Default is internal name.

- target_name

  Target name.

- export_members

  Exports the target members.

  – True

  – False

  Default is false.

- output_file

  Output zip file with an absolute path.

Example

The following example exports the latest configuration to a host file named "test_host" to the file "ExportConfig.zip". The target members are not exported.

```
emcli export_latest_config      -target_type="host"
      -target_name="test_host"
      -export_members="false"
      -output_file="ExportConfig.zip"
```

export_masking_definition

Exports a masking definition in XML format.

Format

```
emcli export_masking_definition
      -definition_name=<masking_definition_name>
      [-path=file_path]
```

```
        [-file=file_name]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **definition name**

  Masking definition name.

- **path**

  Path for the file name to save the masking script. The file name is auto-generated. -path and -file are mutually exclusive. Only an absolute path is allowed.

- **file**

  File name to save the masking script. The file name must include the absolute path. -path and -file are mutually exclusive.

Output Columns

Success/Error messages.

Examples

Example 1

This example exports the masking definition mask_hr_data to an XML file at the specified path:

```
emcli export_masking_definition
      -definition_name=mask_hr_data
      -path=/tmp/
```

Example 2

This example exports the masking definition mask_hr_data to an XML file named abc.xml:

```
emcli export_masking_definition
      -definition_name=mask_hr_data
      -file=/tmp/abc.xml
```

export_metric_extension

Exports a metric extension archive file.

Format

```
emcli export_metric_extension
      -file_name=<metric_extension_archive_name>
      -target_type=<metric_extension_target_type>
      -name=<metric_extension_name>
      -version=<metric_extension_version>
```

Options

- **file_name**

  Name of the metric extension archive file to export into.

- **target_type**

Target type of the metric extension.

- **name**

  Name of the metric extension.

- **version**

  Version of the metric extension to be exported.

Example

This example creates an archive of a metric extension of a given target type, name, and version.

```
emcli export_metric_extension -file_name=<name of the metric extension archive> -
target_type=<target type of the metric extension> -name=<name of the metric
extension -version=<version of the metric extension>
```

export_report

Exports an Information Publisher report definition and all of its element definitions given its title and owner.

Format

```
emcli export_report
      -title=<report_title>
      -owner=<report_owner>
      -output_file=<file>
```

Options

- **title**

  Title of the report to export. To export copies of Oracle-provided reports, the title value should be the internal report title stored in the repository. To avoid using the internal title, make a copy of the report and provide your own custom title, then use your title to export the report.

- **owner**

  The owner of the report to export. The logged-in emcli user must have view privilege for the report. Target names are not exported. The report is uniquely defined using title and owner, so both must be supplied.

- **output_file**

  Name of the exported file.

Examples

```
 emcli export_report        -title=Maintenance_Report        -
owner=SHIFT1_OPERATOR        -output_file=$HOME/reports/maint_report.xml
```

export_saved_config

Exports a saved target configuration.

Format

```
emcli export_saved_config
      -name="saved config name"
      [-output_file="ExportConfig.zip"]
```

[ ] indicates that the options is not mandatory.

Options

*   name

    The name of the saved configuration to delete.

*   output_file

    Output zip file with an absolute path. This is an optional parameter.

Examples

The following command exports the saved configuration with the name "blr2201989.idc.oracle.com|host|20141210130723" to the file "ExportConfig.zip": :

```
emcli export_saved_config
      -name="blr2201989.idc.oracle.com|host|20141210130723"
      -output_file="ExportConfig.zip"
```

export_sla

Extracts the configuration details of an SLA into a local file. If you do not specify slaName and/or version, multiple SLA are exported to the same output file.

Format

```
emcli export_sla
    -targetName=<target_name>
    -targetType=<target_type>
    [-slaName=<SLA_name>]
    -output_file=<output_filename>
```

[ ] indicates that the parameter is optional

Options

*   **targetName**

    Name of the target.

*   **targetType**

    Type of target.

*   **slaName**

    Name of the SLA.

*   **output_file**

    Output file name of the template. If the file does not exist, it is created; if it already exists, it is overwritten. (This assumes that the extract operation was successful. If the operation fails, no files are created, and any existing files remain unchanged.)

Example

This example creates an output file named 'service_sla.xml' that contains configuration details of the 'gold_sla' SLA for the target 'my_service'.

```
emcli export_sla
        -targetName='my_service'
        -targetType='generic_service'
        -slaName='gold_sla'
        -output_file='service_sla.xml'
```

export_standard

Exports a standard from the repository to an XML file.

Format

```
emcli export_standard
      -name=<name>
      -author=<author>
      -version=<name>
      -output_file=<file>
```

Options

*   **name**

    Name of the standard to be exported.

*   **author**

    Author of the standard to be exported.

*   **version**

    Author of the standard to be exported.

*   **output_file**

    Name of the exported file.

Example

```
emcli export_standard \
      -name=foo \
      -author=Curly \
      -version=99 \
      -output_file=$HOME/reports/standard.xml
```

export_subset_definition

Exports the specified subset definition as an XML file at the specified directory location.

Format

```
emcli export_subset_definition
   -subset_name=<subset_definition_name>
   [-file_name=<file_name>]
   [-directory=<directory_path>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **subset_name**

  Subset definition name to export.

- **file_name**

  File name to save the exported file. If you do not specify the file name, it is saved under the subset definition name. If it is specified without an extension, '.xml' is used as the default extension.

- **directory**

  Directory location to save the exported file. If you do not specify a directory, the file is saved in the current directory.

Output

Export success or error message.

Examples

Example 1

This example exports a subset definition with the name HR_Subset as XML in the current directory.

```
emcli export_subset_definition -subset_name=HR_Subset
```

Example 2

This example exports a subset definition with the name HR_Subset as XML with the name HR_Subset_Export at the directory path /scratch/subset.

```
emcli export_subset_definition -directory=/scratch/subset -subset_name=HR_Subset -
file_name=HR_Subset_Export
```

export_swlib_cache_files

Exports Software Library entity files from a compressed file to a file system location on a host target.

Format

```
emcli export_swlib_cache_files
     -dest_dir_path="dest_dir_path"
     -zip_file_name="zip_file_name"
     -dest_host_name="dest_host_name"
     -urn_file_entry="urn_file_entry"
     -dest_host_tmp_dir="dest_host_tmp_dir"
     -credential_set_name="setname"] | [-credential_name="name" -
credential_owner="owner"]
```

Parameters

- dest_dir_path

  Location on the host where the compressed file is created.

- zip_file_name

  Name of the compressed file created.

- dest_host_name

  Name of the host target on which the compressed file is created.

- urn_file_entry_file

  File on the emcli client host which contains identifier of the entity revision followed by # followed by file name. If the identifier is not followed by #, all file entries of the entity revision will be exported.

- dest_host_tmp_dir

  Directory on the destination host in which temporary files are created for export.

- credential_set_name

  The set name of the preferred credential stored in the Management Repository for the host target. It can be one of the following:

  – HostCredsNormal - default unprivileged credential set

  – HostCredsPriv - privileged credential set

- credential_name

  Named credential stored in the Management Repository. This option must be specified along with the `-credential_owner` option.

- credential_owner

  Owner of a named credential stored in the Management Repository. This option must be specified along with the `-credential_name` option.

Example

Creates a zip file `/u01/export_loc/exportedfile.zip` on the host `export.us.acme.com` which has file modules`/zipUtil.pm` from entity with id `oracle:defaultService:em:provisioning:1:cmp:COMP_Component:SUB_Generic:B1B 1880C6A8C62AAE0548C42832D14:0.1` using the named credential `'MyAcmeCreds'` owned by `'ACME_USER'`. Temporary files will be created in the / temp directory of the host `export.us.acem.com`.

```
emcli export_swlib_cache_files
      -dest_dir_path="/u01/export_loc"
      -zip_file_name="exportedfile.zip"
      -dest_host_name="export.us.acme.com"
      -urn_file_entry_file="/u01/export_urnfile"
      -dest_host_tmp_dir="/tmp"
      -credential_name="MyAcmeCreds"
      -credential_owner="ACME_USER"
```

export_template

Exports a monitoring template. You can export a template to the file system in the form of an XML file, or you can print it on standard output in XML form.

Format

```
emcli export_template
      -name=<name>
```

```
            -target_type=<target_type>
            [-output_file=<file_for_exported_template>]
            [-archive]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the template. The name and target type uniquely identify a template.

- **target_type**

  Target type of the template.

- **output_file**

  Specifies the file to output the template. If not specified, the template prints to `stdout`.

- **archive**

  Indicates that the template must be exported as a zip file. When a Metric Extension is included in the template, this option is required to export the template as a zip file.

Examples

Example 1

This example shows that template XML specified by name `HOST_TEMP1` and target type `host` will be output to the screen.

```
emcli export_template -name=HOST_TEMP1 -target_type=host
```

Example 2

This example shows that template XML specified by name `HOST_TEMP1` and target type `host` will be created in the `test.xml` file.

```
emcli export_template -name=HOST_TEMP1 -target_type=host -output_file=test.xml
```

export_update

Exports a Self Update archive file from Enterprise Manager to the specified location.

Format

```
emcli export_update
        -id="internal id"
        -dir="dir"
        -omslocal
emcli export_update
        -id="internal id"
        -dir="dir"
        -host="hostname"
        [-credential_set_name="setname"] | -credential_name="name"
          -credential_owner="owner"
```

```
[ ]  indicates that the parameter is optional
```

Options

- **id**

Internal identification for the update to be exported.

- **dir**

  Complete path of the directory where the update is to be exported.

- **omslocal**

  Flag specifying that the directory is accessible from the OMS.

- **host**

  Target name for a host target where the update is to be exported.

- **credential_set_name**

  Set name of the preferred credential stored in the repository for the host target. Can be one of the following: HostCredsNormal — Default unprivileged credential set HostCredsPriv — Privileged credential set

- **credential_name**

  Name of a named credential stored in the repository. You must specify this option along with the credential_owner option.

- **credential_owner**

  Owner of a named credential stored in the repository. You must specify this option along with the credential_name option.

Examples

Example 1

This example exports the update archive file to /u01/common/ on host host1.example.com. The host must be the managed host target in Enterprise Manager, and the Management Agent on this host must be up and running. The preferred unprivileged credentials for host host1.example.com are used to push the remote file.

```
emcli export_update
       -id="914E3E0F9DB98DECE040E80A2C5233EB"
       -dir="/u01/common/"
       -host="host1.example.com"
       -credential_set_name="HostCredsNormal"
```

Example 2

This example exports the update archive file to /u01/common/ on host host1.example.com. The host must be the managed host target in Enterprise Manager, and the Management Agent on this host must be up and running. The named credentials "host1_creds" owned by user "admin1" are used to push the remote file.

```
emcli export_update
       -id="914E3E0F9DB98DECE040E80A2C5233EB"
       -dir="/u01/common/"
       -host="host1.example.com"
       -credential_name="host1_creds"
       -credential_owner="admin1"
```

extend_as_home

Clones the specified Application Server Oracle Home or Software Library component from the target host to specified destinations. The new hosts join an existing cluster. For a Portal and Wireless install, OID user and password are also needed. For a J2EE instance connected to only a database-based repository, a DCM Schema password is needed.

Passing Variables Through EM CLI

When working with variables such as `%perlbin%` or `%oracle_home%`, EM CLI passes variable values from the current local environment instead of the variables themselves. To pass variables through an EM CLI command, as might be the case when using the `-prescripts` or `-postscripts` options, you can place the EM CLI command in a batch file and replace all occurrences of `%` with `%%`.

Format

```
emcli extend_as_home
      -input_file="dest_properties:file_path"
      -list_exclude_files="list of files to exclude"
      -isSwLib="true/false"
      -tryftp_copy="true/false"
      -jobname="name of cloning job"
      -iasInstance=instance
      -clustername=name of the cluster to join
      -oldIASAdminPassword=oldpass
      -newIASAdminPassword=newpass
      [-oiduser=oid admin user]
      [-oidpassword=oid admin password]
      [-dcmpassword=dcm schema password]
      [-prescripts=script name to execute"]
      [-run_prescripts_as_root="true/false"]
      [-postscripts=script to execute"]
      [-run_postscripts_as_root="true/false"]
      [-rootscripts=script name to execute"]
      [-swlib_component ="path:path to component;version:rev"]
      [-source_params="TargetName:name;HomeLoc:loc;HomeName:name;
        ScratchLoc:Scratch dir Location"
      [-jobdesc="description"]
```

```
[ ] indicates that the parameter is optional
```

Options

• **input_file=dest_properties**

File containing information regarding the targets. Each line in the file corresponds to information regarding one destination.

Format:

```
Destination Host Name1;Destination Home Loc; Home Name; Scratch Loca
```

For information about the input_file parameter, see -input_file Syntax Guidelines.

• **input_file=list_exclude_files**

Comma-separated list of files to exclude. This is not required if the source is a Software Library. You can use an asterisk "*" as a wildcard.

For information about the input_file parameter, see -input_file Syntax Guidelines.

- **isSwLib**

  Specifies whether it is an Oracle Home database or Software Library.

- **tryftp_copy**

  Try FTP to copy or not. You should set the FTP copy option to false when using EM CLI from the command line.

- **jobname**

  Name of the cloning job.

- **iasInstance**

  Application Server instance.

- **clustername**

  Name of the cluster to join.

- **oldIASAdminPassword**

  Old Application Server administrator password.

- **newIASAdminPassword**

  New Application Server administrator password.

- **oiduser**

  OID administrator user.

- **oidpassword**

  OID administrator password.

- **dcmpassword**

  DCM schema password.

- **prescripts**

  Path of the script to execute.

  > **✏️ Note:**
  >
  > Double-quoted parameters can be passed using an escape (\) sequence. For example:
  >
  > ```
  > prescripts=" <some value here>=\"some value here\" "
  > ```

- **run_prescripts_as_root**

  Run prescripts as `root`. By default, this option is set to false.

- **postscripts**

  Path of the script to execute.

- **run_postscripts_as_root**

  Runs postscripts as `root`. By default, this option is set to false.

- **rootscripts**

Path of the script to execute. You can use the job system environment variables (`%oracle_home%`, `%perl_bin%`) to specify script locations.

- **swlib_component**

  Path to the Software Library to be cloned. `isSwLib` must be true in this case.

- **source_params**

  Source Oracle home information. `isSwLib` must be false in this case.

- **jobdesc**

  Description of the job. If not specified, a default description is generated automatically.

Examples

```
emcli extend_as_home
    -input_file="dest_properties:/home/destinations.txt"
    -list_exclude_files="centralagents.lst"
    -isSwLib="false"
    -tryftp_copy="false"
    -jobname="extend as home"
    -iasInstance="asinstancename"
    -isIas1013="false"
    -clustername=ascluster
    -oldIASAdminPassword="oldpassword"
    -newIASAdminPassword="newpassword"
    -prescripts="/home/abc/myscripts"
    -run_prescripts_as_root="true"
    -rootscripts="%oracle_home%/root.sh"
    -source_params="TargetName:host.domain.com;HomeLoc=/home/oracle/appserver1;
     HomeName=oracleAppServer1;ScratchLoc=/tmp"
```

extract_template_tests

Extracts variables and test definitions from a repository template into a local file.

Format

```
emcli extract_template_tests
      -templateName=<template_name>
      -templateType=<template_type>
      -output_file=<output_filename>
      [-encryption_key=<key>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **templateName**

  Name of the template.

- **templateType**

  Type of template.

- **output_file**

  Name of the output file. If the file does not exist, it will be created; if it already exists, it will be overwritten. (This is assuming the extract operation was successful; if the operation fails, no files are created, and any existing files are left unchanged.)

- **encryption_key**

  Key to encrypt the file contents. The same key should be used to decrypt the file.

Example

This example creates a file named `my_template.xml` containing the variable values and test definitions of the Web Application template `my_template`. The file contents are encrypted using the key `my_password`.

```
emcli extract_template_tests
     -templateName=my_template -templateType=website
     -output_file=my_template.xml -encryption_key=my_password
```

> **Note:**
>
> - The emcli user must have operator privilege on the repository template to perform this operation.
>
> - Beacon-related information is not exported to the file. In particular, the list of monitoring beacons, as well as any beacon-specific properties or thresholds, are not exported.
>
> - The values of password variables are not exported.

failover (deprecated)

Performs a Data Guard failover from a primary database to a standby database.

> **Note:**
>
> Since failover is deprecated, use dg_failover instead.

Format

```
emcli failover
     -primary_target_name="<primary target name>"
     -primary_target_type="oracle_database|rac_database"
     -standby_target_name="<standby target name>"
     -standby_target_type="oracle_database|rac_database"
     [-primary_db_creds_name="<primary database credential name>"]
     [-primary_host_creds_name="<primary database host credential name>"]
     [-standby_db_creds_name="<standby database credential name>"]
     [-standby_host_creds_name="<standby database host credential name>"]
     [-swap_jobs]
     [-swap_thresholds]
     [-immediate]
```

Options

- primary_target_name

  Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- primary_target_type

  Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- standby_target_name

  Standby database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- standby_target_type

  Standby database target type. Specify oracle_database for single instance, or rac_database for cluster.

- primary_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- primary_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle Home. Default is to use preferred credential.

- standby_db_creds_name

  Standby database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- standby_host_creds_name

  Standby database host named credential for an operating system user who can access the standby database Oracle Home. Default is to use preferred credential.

- swap_jobs

  Specifies that scheduled jobs on either the primary or standby database that are of a transferable type (Backup, RMAN Script, and SQL Script) will be moved to the other database after Enterprise Manager detects a fast-start failover. Default is not enabled.

- swap_thresholds

  Specifies that monitoring settings will be swapped between the primary and standby databases after Enterprise Manager detects a fast-start failover. Default is not enabled.

- immediate

  Specifies an immediately failover without waiting for all the available redo data to be applied on the standby database. This is the fastest way to failover. However, data may be lost. Without this option, a complete failover will be performed minimizing the data loss. Oracle recommends a complete failover. Default is not enabled.

Examples

Example 1

The following command performs a complete failover from a primary single-instance database named "database" to a standby single-instance database named "database1".

```
emcli failover
     -primary_target_name="database"
     -primary_target_type="oracle_database"
     -standby_target_name="database1"
     -standby_target_type="oracle_database"
```

Example 2

The following command performs an immediate from a primary single-instance database named "primary" to a standby single-instance database named "standby".

```
emcli failover
      -primary_target_name="primary"
      -primary_target_type="rac_database"
      -standby_target_name="standby"
      -standby_target_type="rac_database"
      -immediate
```

fix_compliance_state

Removes stale associations/results related to targets that have been deleted.

Format

```
fix_compliance_state
```

Example

```
emcli fix_compliance_state
```

fmw_discovery_prechecks

Checks if the host configuration is collected or not. If it is not yet collected, it initiates a configuration collection for the hosts.

Format

```
emcli fmw_discovery_prechecks
      -hostnames=<comma separated list of host names>
```

Options

• hostnames

    CSV (Comma Separated Value) list of host names.

Exit Codes

0 if verb processing is successful.

A non-zero value indicates that verb processing was unsuccessful.

Example

The following example checks if the host configuration is collected:

```
emcli fmw_discovery_prechecks      -hostnames="host1.domain.com,host2.domain.com"
```

generate_activity_report

Generates a current activity report for OMS.

Format

```
emcli generate_activity_report
```

generate_discovery_input

This verb further automates the process of adding several WebLogic Domains to the Cloud Control console. You can run this verb after automatic discovery has already discovered several WebLogic Domains.

This verb creates a discovery input file automatically based on the targets discovered from the automatic discovery operation. You can then use this discovery input file in conjunction with the discover_wls verb to further automate the process of promoting discovered domains as fully managed targets. Consequently, you do not need to manually create a discovery input file to perform domain discovery from EM CLI.

Format

```
emcli generate_discovery_input
        -out_file=<fully_qualified_path_of_output_file>
```

Options

*   **out_file**

    Location where the output file will be generated. Verify that the OMS user has write permissions on the specified location. If you are invoking the verb from the EM CLI client, verify that you have read permissions on the specified location.

Examples

This example creates the ouput file /tmp/myFile.csv.

```
emcli generate_discovery_input -out_file=/tmp/myFile.csv
```

generate_downtime_contact

Generates a list of email addresses of recipients that will be notified if the specified target goes down.

You can optionally save this email list into the downtime contact target property to be used by always-on monitoring.

Format

```
emcli generate_downtime_contact
    -target_name="<target name>"
    -target_type="<target type>"
    [-set]

[ ]  indicates that the parameter is optional.
```

Options

*   target_name

    Target name.

*   target_type

    Target type.

*   set (Optional)

Use this flag to set the downtime contact property for the specified target automatically.

Examples

Example 1

The following command obtains downtime email contacts for the target *localhost.site.com* of type *host*.

```
emcli generate_downtime_contact -target_name="localhost.site.com" -
target_type="host"
```

Example 2

The following command obtains downtime email contacts for the target *localhost.site.com* of type *host*. The downtime contacts property for this target will be automatically updated.

```
emcli generate_downtime_contact -target_name="localhost.site.com" -
target_type="host" -set
```

generate_masking_script

Generates a masking script for the given masking definition.

Format

```
emcli generate_masking_script
      -definition_name=masking_definition_name
      [-tablespace_name=tablespace_name]
      [-parameters=<name1:value1;name2:value2;...>]
      [-credential_name=cred_name]
      [-input_file=<parameter_tag:file_path>]
      [-generate_export=Y|N]
      [-generate_mask=Y|N]
      [-script | -format=[name:<pretty|script|csv>];
                          [column_separator:column_sep_string];
                          [row_separator:row_sep_string];
```

```
[ ]  indicates that the parameter is optional
```

Options

* **definition_name**

  Name of the masking definition.

* tablespace_name

  Name of the masking definition.

* **parameters**

  List of name-value pairs that represent the credentials required for connecting to the database instance. The supported parameters are db_username, db_password, and db_role.

* **credential_name**

  Name of the database credential. This option is mandatory when the db_username and db_password parameters are not specified.

- **input_file**

  Used in conjunction with the 'parameters' option, this enables you to store parameter values, such as username and password, in a separate file. This specifies a mapping between a tag and a local file path. The tag is specified in lieu of specific parameter values of the 'parameters' . The tag must not contain colons (:) or semi-colons (;).

  For information about the input_file parameter, see -input_file Syntax Guidelines.

- **generate_export**

  Specify whether to generate a script to export masked data from the specified source database using Oracle Data Pump. Specify Y or N.

- **generate_mask**

  Specify whether to generate a script to replace sensitive data in-place with masked data on a specified (nonproduction) database. Specify Y or N.

- **script**

  This is equivalent to -format='name: script'.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - format="name:script;row_separator:<row_sep_string>" row-separates the verb output by <row_sep_string>. Columns are separated by the tab character.

Output

Success or error messages as well as the impact report (if generated).

Examples

Example 1

The following example generates a script for the masking definition named mask_hr_data. The database password is read from the pwd.txt file:

```
emcli generate_masking_script
     -definition_name=mask_hr_data
     -parameters=PWD_FILE
     -input_file=PWD_FILE:pwd.txt
```

Example 2

The following example reads the database credentials from the named credential DB_NC and generates a masking script for the masking definition named mask_hr_data:

```
emcli generate_masking_script
    -definition_name=mask_hr_data
    -credential_name=DB_NC
```

generate_subset

Generates a subset using the specified subset definition and target database.

Format

```
emcli generate_subset
    -subset_name=<subset_definition_name>
    -target_name=<target_name>       -target_type=<target_type>
    [-in_place_delete]
    [-db_pref_cred_name=<DBCredsNormal | DBCredsSYSDBA>]
    [-db_cred_name=<database_credential_name>]
    [-host_cred_name=<host_credential_name>]
    [-rule_parameters=<rule_parameters>]
    [-subset_directory=<database_directory_object_name>]
    [-custom_directory_path=<custom_directory_path> ]
    [-use_external_directory]
    [-external_directory=<external_directory_object_name>]
    [-export_file_name=<export_file_name>]
    [-max_file_size=<maximum_file_size>]
    [-max_threads=<maximum_number_of_threads>]
    [-compress_dump_file=<compress_dump>]
    [-encrypt_dump_file]
    [-encryption_password=<encryption_password>
    [-confirm_encryption_password=<encryption_password_confirmation>]
    [-seed_flag]
    [-seed_password=<seed_password>]
    [-log_file_name=<log_file_name>]
    [-job_name=<job_name>]
    [-job_description=<job_description>]
    [-is_non_prod_env]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **subset_name**

  Name of the existing subset definition to generate the subset.

- **target_name**

  Database target name.

- **target_type**

  Type of target. The possible values for target type are 'oracle_database', 'rac_database', and 'oracle_pdb'.

- **in_place_delete**

  Determines whether to generate a subset operation. The default operation is exporting data into a dump file. Set this flag to delete data from the specified target.

- **db_pref_cred_name**

Name of preferred credentials stored in the Enterprise Manager repository. You must provide a value for either db_pref_cred_name_or db_cred_name. The valid values for this option are:

- DBCredsNormal — Default normal credential set for an oracle_database target.

- DBCredsSYSDBA — SYSDBA credential set for an oracle_database target.

- **db_cred_name**

  Name of existing credentials stored in the Enterprise Manager repository to connect a selected target database.

- **host_cred_name**

  Name of existing host credentials stored in the Enterprise Manager repository to access the target host. If you do not specify a value, the preferred host credentials set for this target are used to access the target host.

- **rule_parameters**

  Maps values to rule parameter names.You must specify the value for this option if any of the rule parameters have missing values. However, you can also override the specified values using this option.

  For example:

  ```
  -rule_parameters="department_id_param:80;order_id_param:2400"
  ```

- subset_directory

  Directory location object name to save dump and log files. For example:

  ```
  DATA_PUMP_DIR
  ```

- **custom_directory_path**

  User-specified directory location on the target host to save dump and log files. You must provide a value for either subset_directory or custom_directory_path. For example:

  ```
  /scratch/user/subset_dir
  ```

- **use_external_directory**

  Enables the external directory (clustered/shared file system or ASM) for a faster export dump. If this option is set, you need to provide a value for external_directory.

- **external_directory**

  External directory (clustered/shared file system or ASM) for a faster export dump. For example:

  ```
  DATA_PUMP_DIR
  ```

- **export_file_name**

  File name to save the dump file. If not specified, the default value is EXPDAT%U.DMP. You can wildcard a set of dump files using '%U' in the file name.

- **max_file_size**

  Maximum file size in MB. If not specified, the default value is 100.

- max_threads

  Maximum number of threads created for export operation. If not specified, the default value is 1.

- **compress_dump_file**

Enables data compression during the export operation.

- encrypt_dump_file

  Enables data encryption during the export operation.

- **encryption_password**

  Password key to encrypt data during export operation. If encrypt_dump_file is set and a value for this option is not specified, you are prompted for the encryption password. For a secure operation, it is recommended that passwords should not be stored in the scripts, but specified instead when prompted for them.

- **confirm_encryption_password**

  The value for this option should be the same as encryption_password. If encrypt_dump_file is set and the value for this option is not specified, you are prompted for confirmation of the encryption password. For a secure operation, it is recommended that passwords should not be stored in the scripts, but specified instead when prompted for the them.

- **seed_flag**

  Indicates that the subset definition contains one or more masking definitions, and any of the masking definitions contains a substitute or encrypt format.

- **seed_password**

  Seed string to be used if the subset definition contains one or more masking definitions and any of the masking definitions contains a substitute or encrypt format. If seed_flag is set and a value for this option is not specified, you are prompted for the seed password. The seed can be any text string. For a secure operation, it is recommended that passwords should not be stored in the scripts, but specified instead when prompted for them.

- log_file_name

  File name to save the log file. The default value is EXPDAT.LOG.

- **is_non_prod_env**

  Confirmation that the specified database is not a production database. This option is mandatory for the in-place delete option.

- **job_name**

  Generates the subset job name.

- **job_description**

  Job description.

Output

Success or error messages as well as the job name if applicable.

Examples

Example 1

This example exports data into a dump file using rule parameters.

```
emcli generate_subset
      -subset_name=hr_subset
      -target_name=sample_database
      -target_type=oracle_database
```

```
            -export_file_name=EXPDAT.DMP
            -db_cred_name=db_cred
            -host_cred_name=host_cred
            -subset_directory=DATA_PUMP_DIR
            -rule_parameters="department_id:80;order_id:2400"
```

Example 2

This example exports data into a dump file with encryption enabled. You are prompted for encryption_password and confirm_encryption_password.

```
emcli generate_subset
      -subset_name=hr_subset
      -target_name=sample_database
      -encryption_password
      -target_type=oracle_database
      -export_file_name=EXPDAT.DMP
      -db_cred_name=db_cred
      -host_cred_name=host_cred
      -subset_directory=DATA_PUMP_DIR
```

generate_ui_trace_report

Generates a user interface page performance trace report, which enables you to identify slow rendering pages.

Format

Standard Mode

```
emcli generate_ui_trace_report     [-user_name="user_name"]     [-
start_time="start_time"]     [-duration="duration"]
```

Interactive or Script Mode

```
generate_ui_trace_report(     [user_name="user_name"]
[,start_time="start_time"]     [,duration="duration"]     )

[ ]  indicates that the parameter is optional.
```

Options

*   user_name

    User name for which the page performance trace report will be generated. Default is the currently logged in user.

*   start_time

    Start time in mm:dd:yy format from where page performance trace report has to be generated. Default is current time - 1 hour.

*   duration

    Duration in hh:mm format for which the page performance trace report has to be generated. Default is 1 hour.

Example

The following example generates and downloads the UI page performance trace report for the last 6 hours.

```
emcli generate_ui_trace_report -duration 06:00
```

get_accesspoints

Retrieves all access point targets and their current status for a given server target..
Get all Access point targets and their current status for a given server target.

Format

```
emcli get_accesspoints
      -name="<server target name>"
      -type="<server target type>"
     [-noheader]
     [-limit_rows="<Maximum number of targets to be retrieved>"]
```

[ ] indicates that the parameter is optional.

Options

- name

  Name of server target.

- type

  Type of server target.

- noheader

  Displays tabular output without column headers.

- limit_rows

  The maximum number of targets to be retrieved. Defaults to 2000 rows if not
  specified.

Example

The following example lists the first 10 access points with their current status for the
server target server_name:server_type without headers, in tabular output. Results are
ordered by target type and target name.

```
emcli get_accesspoints
     -name="server_name"
     -type="server_type"
     -noheader
     -limit_rows="10
```

get_add_host_status

Displays the latest status of an Add Host session.

Format

```
emcli get_add_host_status
     -session_name="Session name"
     [-details]
     [-show_only_failed_hosts]
     [-host_name="Host name"]
     [-noheader]
     [-script | -format=
```

```
                [name:<pretty|script|csv>];
                [column_separator:"column_sep_string"];
                [row_separator:"row_sep_string"];
        ]
```

`[ ]  indicates that the parameter is optional.`

Options

- **session_name**

  Name of the session whose status you want to view.

- **details**

  Displays additional information for the given session.

- **show_only_failed_hosts**

  Displays only the hosts on which the Add Host operation failed.

- **host_name**

  Displays the details of the provided host.

- **noheader**

  Display tabular output without column headers.

- **script**

  This is equivalent to `-format="name:script"`.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Host, Platform Name, Initialization, Remote Prerequisite, Agent Deployment, Error

Error Codes

0 - Success

1 and 223 - Syntax Error

Examples

Example 1

This example displays the detailed status of host 'example.com' for the session 'ADD_HOST_SYSMAN_Jun_6_2013_11:26:43_PM_PDT'.

```
emcli get_add_host_status
 -session_name=ADD_HOST_SYSMAN_Jun_6_2013_11:26:43_PM_PDT
 -host_name=example.com
```

Example 2

This example displays only the failed hosts for the session 'ADD_HOST_SYSMAN_Jun_6_2013_11:26:43_PM_PDT'.

```
emcli get_add_host_status
 -session_name=ADD_HOST_SYSMAN_Jun_6_2013_11:26:43_PM_PDT
 -show_only_failed_hosts
```

get_agent_properties

Displays Management Agent properties. You can use this command if you have view privilege for the Management Agent.

Format

```
emcli get_agent_properties     -agent_name="<agent_target_name>"     [-
all]     [-format="<format_name>"]
```

[ ] indicates that the parameter is optional

Options

- **agent_name**

  Name of the Management Agent target.

- **all**

  Shows all Management Agent properties. By default, only basic properties appear.

- **format**

  Format to display Management Agent properties. Valid values are pretty, script, and csv. By default, values are displayed in pretty format.

Examples

This example shows all of the Management Agent properties in CSV format:

```
 emcli get_agent_properties -agent_name=agent.example.com:11850          -
all        -format=csv
```

get_agent_property

Displays the value of a specific Management Agent property. You can use this command if you have view privilege for the Management Agent.

Format

```
emcli get_agent_property       -agent_name=<agent_target_name>      -
name=<agent_property_name>
```

Options

- **agent_name**

  Name of the Management Agent target.

- **name**

  Name of the Management Agent property.

Examples

This example shows the current value of the UploadInterval property in emd.properties.

```
emcli get_agent_property -agent_name=agent.example.com:11850
        -name=UploadInterval
```

get_agent_unsubscribe_status

Displays the status of unsubscribed Management Agents.

Format

```
emcli get_agent_unsubscribe_status
      -version_name | -op_name
      [-agent="agent_name_pattern"]
      [-severity="ERROR|WARNING"]
      [-severity_id="severity_id"]
      [-status="PendingUpdateInprogress|
Updatable|NotUpdatable|NotExecuted|Success|Inprogress|Failed"]
```

```
[ ] indicates that the parameter is optional.
```

Parameters

- version_name

  Version name of the unsubscribed Management Agent gold image.

- op_name

  Operation name of the unsubscribed Management Agent gold image.

- agent

  Agent name of the unsubscribed Management Agent gold image.

- severity

  Severity status of the unsubscribed Management Agent gold image.

- severity_id

  Severity ID of the unsubscribed Management Agent gold image.

  > **✎ Note:**
  >
  > It is mandatory to specify the -op_name parameter or the -version_name parameter. If you have specified -severity or -severity_id, then ensure that you do not specify -version_name or -status.

ORACLE®

- status

  Status of the unsubscribed Management Agent gold image.

Examples

Example 1

The following example displays the details of the unsubscribe operations submitted for the Management Agent gold image 'OPC_AGT_ADC_POD_JUNE', for the Management Agent xyz.domain.com:1243:

```
emcli get_agent_unsubscribe_status
      -version_name="OPC_AGT_ADC_POD_JUNE"
      -agent="xyz.domain.com:1243"
```

Example 2

The following example displays the details of the unsubscribe operation 'UNSUBSCRIBE_JOB123' for the Management Agent xyz.domain.com:1243, having the status 'Failed':

```
emcli get_agent_unsubscribe_status
      -op_name="UNSUBSCRIBE_JOB123"
      -status="Failed" -agent="xyz.domain.com:1243"
```

get_agent_update_status

Displays the Management Agent update results.

Format

```
emcli get_agent_update_status
      -version_name | -op_name
      [-agent="agent_name_pattern"]
      [-severity="ERROR|WARNING"]
      [-severity_id="severity_id"]
      [-status="PendingUpdateInprogress|Updatable|NotUpdatable|NotExecuted|
Success
       |Inprogress|Failed"]

[ ]  indicates that the parameter is optional.
```

Options

- version_name

  Displays the details of the update operation submitted for the specified Management Agent gold image version name.

- op_name

  Displays the details of the specified update operation.

- agent

  Name pattern of the Management Agents for which the update operation details should be displayed.

- status

  Status for which the update operation details should be displayed.

- severity

  Severity level for which the update operation details should be displayed.

- severity_id

  Severity ID for which the update operation details should be displayed.

> **Note:**
>
> It is mandatory to specify the -op_name parameter or the -version_name parameter. If you have specified -severity or -severity_id, then ensure that you do not specify -version_name or -status.

Examples

Example 1

The following example displays the Management Agents of the update operation `UPDATE_JOB123`, for which severity is 'WARNING', and severity ID is `ROOT_RUN_CHECK`.

```
emcli get_agent_update_status
     -op_name="UPDATE_JOB123"
     -severity="WARNING"
     -severity_id="ROOT_RUN_CHECK"
     -op_name="UPDATE_JOB123"
```

Example 2

The following example displays the details of the update operation `UPDATE_JOB123` for the Management Agent `xyz.domain.com:1243`, with severity set to `WARNING` and severity ID set to `ROOT_RUN_CHECK`.

```
emcli get_agent_update_status
     -op_name="UPDATE_JOB123"
     -severity="WARNING"
     -severity_id="ROOT_RUN_CHECK"
     -agent="xyz.domain.com:1243"
```

get_agent_upgrade_status

Shows Agent upgrade results.

Format

```
emcli get_agent_upgrade_status
     [-agent]
     [-job_name]
     [-status]

[ ]  indicates that the parameter is optional
```

Options

- **agent**

  Shows the upgrade job details of the specified Agent names or Agent name patterns separated by commas.

- **job_name**

  Shows the upgrade job details of the specified job name.

- **status**

  Shows the upgrade job details with the specified status.

Permutations for combinations of parameters are as follows:

No parameters — Shows <JOB NAME, JOB STATUS, NUMBER OF AGENTS IN THE JOB, JOB START TIME, JOB END TIME> for each job.

-job_name only — Shows <AGENT_NAME, UPGRADE STATUS OF AGENT, UPGRADE START TIME, UPGRADE END TIME> for each Agent in the job, where job name is passed in the -job_name parameter.

-agent only — Shows <JOB NAME, UPGRADE STATUS OF AGENT IN THE JOB, UPGRADE START TIME, UPGRADE END TIME> for each job where the Agent is present and the Agent name passed in the -agent parameter.

-agent and -status only — Shows <JOB NAME, UPGRADE START TIME, UPGRADE END TIME> for each job in which the Agent and Agent upgrade status are passed in -agent and -status, respectively.

-job_name and -agent only — Shows <JOB STEP NAME, JOB STEP STATUS, JOB STEP START TIME, JOB STEP END TIME> for each step in the job for the Agent passed in the -job_name and -agent parameters.

-job_name and -status only — Shows <AGENT_NAME, UPGRADE START TIME, UPGRADE END TIME> for each Agent in the job in which the Agent upgrade status is passed in -job_name and -status, respectively

-job_name, -agent, and -status — Shows <JOB STEP NAME, JOB STEP START TIME, JOB STEP END TIME> for each step in the job for the Agent in which the step status is passed in -job_name , -agent , and -status, respectively

-status only — Shows <JOB NAME, NUMBER OF AGENTS IN THE JOB, JOB START TIME, JOB END TIME> for each job in which job status is passed in the -status parameter.

Examples

Example 1

This example shows the Agent upgrade job details for the Agent xyz.domain.com:1243 .

```
emcli get_agent_upgrade_status -agent="xyz.domain.com:1243"
```

Example 2

This example shows the Agent upgrade job details for the job UPGRADE_JOB123 .

```
emcli get_agent_upgrade_status  -job_name="UPGRADE_JOB123"
```

get_agentimage

Gets the Management Agent image for the particular platform and version provided as inputs.

Format

```
emcli get_agentimage
   -destination=<download_directory>
   -platform="<platform>"
   [-version=<version>]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   **destination**

    Directory where you want to download the Management Agent software. Ensure that you have write permission on this location.

    If the destination directory is titled with two or more words separated by a space, enclose the directory name with double-quotes. For instance, if the destination directory is titled /tmp/linuxagentimage, enter the value as -destination="/tmp/linuxagentimage"

*   **platform**

    Platform for which you want to download the software; this must match one of the platforms for which the software is available on the OMS host. Use the emcli get_supported_platforms command to determine this.

*   **version**

    Version of the Management Agent software that you want to download. If you do not specify this , the version defaults to the OMS version.

Examples

```
emcli get_agentimage -destination=/tmp/agtImage -platform=Linux x86 -version=12.1.0.1.0
```

get_agentimage_rpm

Gets the Management Agent image for the Linux platform and version provided as inputs, then converts the image as rpm.

Format

```
emcli get_agentimage_rpm
   -destination=<download_directory>
   -platform=<platform>
   [-version=<version>]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   **destination**

    Directory where you want to download the .rpm file. Ensure that you have write permission on this location.

    If the destination directory is titled with two or more words separated by a space, enclose the directory name with double-quotes. For instance, if the destination directory is titled /tmp/linuxagentimage, enter the value as -destination="/tmp/linuxagentimage"

*   **platform**

ORACLE®

Platform for which you want to download the .rpm file; this must match one of the platforms for which the software is available on the OMS host. Use the emcli get_supported_platforms command to determine this.

- **version**

  Version of the Management Agent for which you want to download the .rpm file. If you do not specify this , the version defaults to the OMS version.

Examples

```
emcli get_agentimage_rpm -destination=/tmp -platform=Linux x86 -
version=12.1.0.1.0
```

get_aggregate_service_info

Gets time zone and availability evaluation function information of an aggregate's service instance.

Format

```
emcli get_aggregate_service_info
      -name=<name>
      -type=<type>
      [-noheader]
      [-script|-format=
              [name:<pretty|script|csv>];
              [column_separator:<sep_string>];
              [row_separator:<row_sep_string>]
      ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Aggregate service name.

- **type**

  Aggregate service type.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This is equivalent to `-format="name:script"`.

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

– `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

– row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli get_aggregate_service_info -name=My_Name
    -type=aggregate_service
```

get_aggregate_service_members

Gets sub-services of an aggregate service instance.

Format

```
emcli get_aggregate_service_members
    -name=<name>
    -type=<type>
    [-noheader]
    [-script|-format=
                    [name:<pretty|script|csv>];
                    [column_separator:<sep_string>];
                    [row_separator:<row_sep_string>]
    ]
```

[ ] indicates that the parameter is optional

Options

*   **name**

    Aggregate service name.

*   **type**

    Aggregate service type.

*   **noheader**

    Displays tabular information without column headers.

*   **script**

    This is equivalent to `-format="name:script"`.

*   **format**

    Format specification (default is `-format="name:pretty"`).

    – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

    – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

    – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

    – `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

    – row_separator:"row_sep_string" row-separates the verb output by
       <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli get_aggregate_service_members -name=My_Name
      -type=aggregate_service
```

get_best_implementer

Retrieves the best implementer for a server/MAP target.

Format

```
emcli get_best_implementer
      -name="<server target name>"
      -type="<server target type>"
      [-cap_name="<capability name>"]
      [-cap_type="<capability type>"]
```

```
[ ] indicates that the parameter is optional.
```

Options

*   name

    Name of the server target.

*   type

    Type of the server target.

*   cap_name

    When given, best implementer for only that capability is returned (optional).

*   cap_type

    When given, best implementers for the specified capability type are returned.
    cap_type=1 for COLLECTION cap_type=2 for ACTION

Example

The following example lists the best implementer for the capability "capability_name"
of the type COLLECTION.

```
emcli get_best_implementer
      -name="server_name"
      -type="server_type"
      -cap_name="capability_name"
      -cap_type="1"
```

get_blackout_details

Gets detailed information for a specified blackout.

Format

```
emcli get_blackout_details
      -name=<name>
      [-createdby=<blackout_creator>]
      [-noheader]
```

```
    [-script | -format=
                [name:<pretty|script|csv>];
                [column_separator:<column_sep_string>];
                [row_separator:<row_sep_string>];
    ]
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the blackout.

- **createdby**

  Enterprise Manager user who created the blackout. The default is the current user. For displaying details of a blackout created using emctl, use -createdby="<SYSTEM>".

- **noheader**

  Displays tabular information without column headers.

- **script**

  This is equivalent to `-format="name:script"`.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - format="name:script;row_separator:<row_sep_string>" row-separates the verb output by <row_sep_string>. Columns are separated by the tab character.

Output Columns

Status, Status ID, Run Jobs, Next Start, Duration, Reason, Frequency, Repeat, Days, Months, Start Time, End Time, TZ Region, TZ Offset

Examples

Example 1

This example shows detailed information for blackout `blackout1` that the current user created.

```
emcli get_blackout_details -name=blackout1
```

Example 2

This example shows detailed information for blackout `blackout1` that user `joe` created.

```
emcli get_blackout_details -name=blackout1 -createdby=joe
```

get_blackout_reasons

Lists all blackout reasons, one per line.

Format

```
emcli get_blackout_reasons
```

Examples

This example lists all blackout reasons, one per line.

```
emcli get_blackout_reasons
```

get_blackout_targets

Lists targets for a specified blackout.

Format

```
emcli get_blackout_targets
      -name=<name>
      [-createdby=<blackout_creator>]
      [-noheader]
      [-script | -format=
                  [name:<pretty|script|csv>];
                  [column_separator:<column_sep_string>];
                  [row_separator:<row_sep_string>];
      ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the blackout.

- **createdby**

  Enterprise Manager user who created the blackout. The default is the current user. For listing details of a blackout created using emctl, use -createdby="<SYSTEM>".

- **noheader**

  Displays tabular information without column headers.

- **script**

  This is equivalent to -format="name:script".

- **format**

  Format specification (default is -format="name:pretty").

  – format="name:pretty" prints the output table in a readable format not intended to be parsed by scripts.

  – format="name:script" sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

– `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

– format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

– format="name:script;row_separator:<row_sep_string>" row-separates the verb output by <row_sep_string>. Columns are separated by the tab character.

Output Columns

Target Name, Target Type, Status, Status ID

Examples

Example 1

This example lists targets in the blackout `blackout1` the current user created.

```
emcli get_blackout_targets -name=blackout1
```

Example 2

This example lists targets in the blackout `blackout1` that user `joe` created.

```
emcli get_blackout_targets -name=blackout1 -createdby=joe
```

get_blackouts

Lists all blackouts or just those for a specified target or one or more hosts. Only the blackouts the user has privilege to view are listed.

Format

```
emcli get_blackouts
      [-target=<name1:type1> | -hostnames=<host1;host2;...>]
      [-noheader]
      [-script | -format=
                    [name:<pretty|script|csv>];
                    [column_separator:<column_sep_string>];
                    [row_separator:<row_sep_string>];
      ]

[ ]  indicates that the parameter is optional
```

Options

• **target**

Lists blackouts for this target. When neither this nor the `-hostnames` option is specified, all blackouts the user has privilege to view are listed.

• **hostnames**

Lists blackouts that have a target on one of the specified hosts. The host name is just the target name part of the host target. For example, specify `host.example.com`, rather than `host.example.com:host`. When neither this nor the `-target` option is specified, all blackouts the user has privilege to view are listed.

• **noheader**

Displays tabular information without column headers.

- **script**

  This is equivalent to `-format="name:script"`.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - format="name:script;row_separator:<row_sep_string>" row-separates the verb output by <row_sep_string>. Columns are separated by the tab character.

Output Columns

Name, Created By, Status, Status ID, Next Start, Duration, Reason, Frequency, Repeat, Start Time, End Time, Previous End, TZ Region, TZ Offset

Examples

Example 1

This example shows all blackouts that cover some target on host `myhost.example.com`.

```
emcli get_blackouts -hostnames=myhost.example.com
```

Example 2

This example shows all blackouts that cover some target on host `myhost.example.com` or on host `yourhost.example.com`.

```
emcli get_blackouts -hostnames=myhost.example.com
-hostnames=yourhost.example.com
```

get_ca_info

Displays information about all of the Certificate Authorities (CA) created since the Cloud Control installation. It also displays the Management Agent names whose certificates are issued by the CA(s) when you specify the -details option. The following information is retrieved from the Cloud Control repository:

- Unique identifier of the Certificate Authority (CA) in the Cloud Control repository

- CA description

- CA creation date

- CA expiration date

- Number of Management Agents registered to this CA

- Number of secured Management Agents not registered to any CA

Format

```
emcli get_ca_info
       [-ca_id=<id1;id2;...>]
       [-details]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **ca_id**

   Specifies the Certificate Authority ID.

- **details**

   For each Certificate Authority, displays the list of Management Agent names whose certificates are issued by it.

Examples

This example shows output for the CA with the ID of 2 specified.

```
emcli get_ca_info -ca_id=2

 Info about CA with ID: 2
 CA is configured
 DN: EMAILADDRESS=Enterprise.Manager@myomshost.mycompany.com,
 CN=myomshost.mycompany.com, OU=EnterpriseManager on myomshost.mycompany.com,
 O=EnterpriseManager on myomshost.mycompany.com, L=EnterpriseManager on
 myomshost.mycompany.com1, ST=CA, C=US, DC=com
 Serial# : 87539237298512593900
 Valid From: Mon Oct 25 17:01:15 UTC 2011
 Valid Till: Thu Oct 22 17:01:12 UTC 2020
 Number of Agents registered with CA ID 2 is 1

 Number of Agents to be re-secured, as OMS is secured using force_newca
     : 1
```

Regarding the force_newca option in the last line, the output shows that a new certificate was created with the ID of 2. Two Management Agents have been re-secured to be registered with this new certificate. The OMS running on myomshost.mycompany.com has been re-secured to be registered with the new certificate created. There is still a Management Agent that needs to be secured to be registered to the new certificate. To retrieve the Management Agent name, you need to run the command "emcli get_ca_info -ca_id=2 -details," which is shown in the next example.

This example displays the Management Agent names registered with the CA(s) for ID 2.

```
emcli get_ca_info -ca_id=2 -details

    Info about CA with ID: 2
    CA is configured
    DN: EMAILADDRESS=Enterprise.Manager@myomshost.mycompany.com,
     CN=myomshost.mycompany.com, OU=EnterpriseManager on myomshost.mycompany.com,
     O=EnterpriseManager on myomshost.mycompany.com, L=EnterpriseManager on
     myomshost.mycompany.com2, ST=CA, C=US, DC=com
    Serial# : 87539237298512593900
    Valid From: Mon Oct 25 17:01:15 UTC 2011
    Valid Till: Thu Oct 22 17:01:12 UTC 2020
```

```
        Number of Agents registered with CA ID 2 is 1
        usagent1.mycompany.com:20872

        Following Agents needs to be re-secured, as OMS is secured using
force_newca
            :

        ukagent1.mycompany.com:1830
```

get_cloud_service_instances

Retrieves the list of cloud service instances. All instances are printed if you do not specify any options.

Format

```
emcli get_cloud_service_instances
        [-user="username"]
        [-family="family"]
        [-type="service type"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **user**

  Identifies the name of the user to be used for filtering service instances.

- **family**

  Identifies the name of the service family to be used for filtering service instances.

- **type**

  Identifies the type of service to be used for filtering service instances.

Examples

Example 1

This example shows all cloud instances that belong to a specified service family (family1):

```
emcli get_cloud_service_instances -family="family1"
```

Example 2

This example shows all cloud instances that belong to a specified service type (type1):

```
emcli get_cloud_service_instances -type="type1"
```

get_cloud_service_requests

Retrieves a list of cloud service requests. All requests are printed if you do not provide any options. Options cannot be used simultaneously.

Format

```
emcli get_cloud_service_requests
        [-user="username"]
```

```
        [-family="family"]
        [-ids="id1;id2..."]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **user**

  Identifies the name of the user to be used for filtering service instances.

- **family**

  Identifies the name of the service family to be used for filtering service instances.

- **ids**

  Lists the Request IDs to be used for filtering cloud requests. Separate each ID with a semicolon (;).

Examples

Example 1

This example shows all cloud service requests that belong to a specified service family (`family1`):

```
emcli get_cloud_service_requests -family="family1"
```

Example 2

This example shows all cloud service requests with a specific request ID (`1` and `2`):

```
emcli get_cloud_service_requests -ids="1;2"
```

get_cloud_user_objects

Retrieves a list of cloud user objects, cloud service instances, and cloud service requests. All objects are printed if you do not provide the -user option.

Format

```
emcli get_cloud_user_objects
        [-user="username"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **user**

  Identifies the name of the user to be used for filtering user objects.

Examples

Example 1

This example shows all cloud user objects, cloud service instances, cloud service requests, and any other objects:

```
emcli get_cloud_user_objects
```

Example 2

This example shows all cloud user objects, cloud service instances, cloud service requests, and any other objects for a specified user (`user1`):

```
emcli get_cloud_user_objects -user="user1"
```

get_compliance_rule_ca

Returns a corrective action identifier for the specified rule internal name optionally filtered by the target type.

Format

```
emcli get_compliance_rule_ca
    -rule_iname="<rule internal name>"
    [-target_type="<target_type>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   rule_iname

    Internal name of the compliance standard rule. Rule internal names are available in the MGMT$CS_RULE_ATTRS view.

*   target_type

    Target type. Use this option to restrict the search to the specified type of target.

Example

The following example retrieves the corrective action identifier for the myrule compliance standard rule.

```
emcli get_compliance_rule_ca
    -rule_iname="myrule"
```

get_compliance_rule_violation_event

Returns a list of unique identifiers for the root standard, runtime identifier, root target, target, event instance ID, and optionally context information for the specified rule and target across the different standards from which it is referred.

Format

```
emcli get_compliance_rule_violation_event
    -rule_iname="<rule_internal_name>"
    -target_type="<target_type>"
    -target_name="<target_name>"
    [-attrs="<attribute_list>"]
            root_cs_guid
            rqs_guid
            root_target_guid
            target_guid
            event_instance_id
    [-separator="<separator>"]
    [-show_context]
            column_name
            column_type=<N | S>
            column_value
```

```
[ ]  indicates that the parameter is optional.
```

Options

- rule_iname

  Internal name of the compliance standard rule. Rule internal names are available in the MGMT$CS_RULE_ATTRS view.

- target_type

  Target type associated with the compliance standard rule.

- target_name

  Name of the target.

- attrs

  List of attributes. If no attributes are specified, then all attributes are returned in the order listed below.

  – root_cs_guid: Unique identifier of the root standard with which the rule is associated.

  – rqs_guid: Unique runtime identifier of the rule referenced with the root standard.

  – root_target_guid: Unique identifier of the root target.

  – target_guid: Unique identifier of the target.

  – event_instance_id: Unique identifier of the event_instance.

- separator

  Separator used between column entries. If no separator is specified, then a comma "," is used.

- show_context

  If show_context is not specified, then the context will not be returned. If show_context is specified, then the following attributes are returned in the following order:

  – column_name: Violation event context attribute name.

  – column_value: Violation event context attribute value.

  – column_type: Violation event context attribute type, "N" if number, "S" if string.

Example

```
emcli get_compliance_rule_violation_event
      -rule_iname="myrule"
      -target_type="host"
      -target_name="my_machine"
      -attrs="root_cs_guid,root_target_guid"
      -show_context

If myrule is associated to the specified target through M root_targets, the output
appears as:std_guid,root_tgt1_guid,[column_name1,column_value11,column_type1],
[column_name2,...],...[column_nameN,...]std_guid,root_tgt2_guid,
[column_name1,column_value1M,column_type1],[column_name2,...],...[column_nameN,...]
```

get_config_history_searches

Gets all the saved history configuration searches.

Format

```
emcli get_config_history_searches
      [-target_type="<target_type>"]
      [-owner="<user>"]
      [-no_header]
```

Options

- target_type

  Target type where the configuration search is created. Default is internal name. It can be a full value or a pattern match using "%".

- owner

  Name or ID of the user who created the configuration history search.

- format

  Specifies the format. Allowed values:

  - format="name:pretty" - Prints out the output table in a readable format.

  - format="name:script" - Sets the default column separator to a tab and the default row separator to a new line.

  - format="name:csv" - Sets the column separator to a comma and the row separator to a new line.

  Default is format="name:pretty".

- noheader

  Displays a tabular output without the column headers.

Examples

Example 1

The following command shows all the configuration searches created on target types whose names contain the pattern "data".

```
emcli get_config_history_searches
      -target_type="%data%"
```

Example 2

The following example shows all the history search created by the user name "Test Admin" and that are created on target type "Oracle Database"

```
emcli get_config_history_searches
      -target_type="oracle_database"
      -owner="Test Admin"
```

get_config_onetimecomparisons

Retrieves all one-time comparisons from the repository.

Format

```
emcli get_config_onetimecomparisons
      [-name="<comparison_name>"]
```

```
                    [-template_name="<template_name>"]
                    [-ref_target_name="<target_name>"]
```

[ ] indicates that the parameter is optional.

Options

- name

  Name of the saved one-time comparison.

- template_name

  Template name of the saved one-time comparison.

- ref_target_name

  Reference target name of the saved one-time comparison.

Examples

Example 1

The following example retrieves all the saved one-time comparisons from the repository with template name "default template".

```
emcli get_config_onetimecomparisons
      -template_name="default template"
```

Example 2

The following example retrieves all the saved one-time comparisons from the repository with the reference target name "host.com".

```
emcli get_config_onetimecomparisons
      -ref_target_name="host.com"
```

get_config_searches

Displays information about saved configuration searches.

Format

```
emcli get_config_searches
      [-target_type="<name_or_pattern>"]
      [-search_created_using="<sql or modeler> "]
      [-system_defined="<yes or no>"]
      [-format="name:<format_option>"]
      [-noheader]
```

[ ] indicates that the parameter is optional.

Options

- target_type

  A string matching the target type on which the configuration search is based. Use the internal target type name. Specify the full name or a pattern match using "%" as a wildcard.

- search_created_using

Indicates whether this search was created using the modeler, through an SQL script, or All, hence the value should be either 'modeler', 'sql', or 'All' . The default is 'All'.

- system_defined

  Indicates whether this search was defined by the System, User , or All, hence the value should be either 'yes' or 'no'. The default is 'All'.

- format

  Format specification (default is -format="name:pretty").

  – format="name:pretty" prints the output table in a readable format not intended to be parsed by scripts.

  – format="name:script" sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings may be specified to change these defaults.

  – format="name:csv" sets the column separator to a comma and the row separator to a newline.

    format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – format="name:script;row_separator:<row_sep_string>" row-separates the verb output by <row_sep_string>. Columns are separated by the tab character.

- noheader

  Display tabular output without column headers.

Examples

Example 1

The following example shows all of the configuration searches created on target types whose names contain the pattern "data" with searches created using SQL.

```
emcli get_config_searches
     -target_type="%data%"
     -search_created_using="sql
```

Example 2

The following example shows all of the configuration searches created on target types whose names contain the pattern "data" and searches created using Modeler and the search is defined by the system.

```
emcli get_config_searches
     -target_type="%data%"
     -search_created_using="modeler"
     -system_defined="yes"
```

get_config_templates

Gets all of the comparison templates.

Format

```
emcli get_config_templates
      [-target_type="oracle_database"]
      [-template_name="host_template"]
      [-owner="SYSMAN"]
      [-list_default_templates="yes"]
      [-list_oracle_provided_templates="no"]
      [-format="[name:<pretty|script|csv>];
            [column_separator:"column_sep_string"];
            [row_separator:"row_sep_string"]"]
      [-noheader]

[ ]  indicates that the parameter is optional
```

Options

- target_type

  Target type on which the comparison template is created. The value should be the internal name. To get the internal name, execute the following EM CLI command:

  ```
  emcli get_target_types
  ```

- template_name

  Name of the template, which can be a full value or a pattern match using "%". The value should be an internal name.

- owner

  Owner of the comparison template, which can be a full value or a pattern match using "%".

- list_default_templates

  Valid inputs are "yes" and "no". If the value of this option is "yes", the result will contain default templates. If the value of this option is "no", the result will not contain default templates. If this option is not specified, the result shows all templates.

- list_oracle_provided_templates

  Valid inputs are "yes" and "no". If this option is provided, the result will be only templates provided by Oracle. If the value of this option is "yes", the result contains Oracle-provided templates. If the value of this option is "no", the result will not contain Oracle-provided templates. If this option is not specified, the result shows all templates.

- format

  Format specification (default is -format="name:pretty").

  - format="name:pretty" prints the output table in a readable format not intended to be parsed by scripts.

  - format="name:script" sets the default column separator to a tab, and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

- format="name:csv" sets the column separator to a comma and the row separator to a newline.

- noheader

  Displays tabular output without column headers.

Output columns:

Template IDTemplate NameTarget TypeDefault — Displays "Yes" if the template is the default, "No" otherwiseOracle Provided — Displays "Yes" if the template is provided by Oracle, "No" otherwiseOwnerSaved TimeTime ZoneDescription

Examples

Example 1

This example shows all of the comparison templates created by the user name "Test Admin" that are created on target type "Test Database" and having the template_name as "Test Database Template".

```
emcli get_config_templates -target_type="oracle_database" -template_name="Test
Database Template" -owner="Test Admin"
```

Example 2

This example shows all of the comparison templates provided by Oracle.

```
emcli get_config_templates -list_oracle_provided_templates="yes"
```

get_connection_mode

Gets the My Oracle Supporrt (MOS) connection mode. The two MOS connection modes are online and offline.

Format

```
emcli get_connection_mode
```

Options

None.

See Also

get_siteguard_script_credential_params

create_patch_plan
delete_patches
describe_patch_plan_input
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

get_credtype_metadata

Prints credential-type information for a credential type. The verb prints credential column names. These column names should be used as parameter names for the create_named_credential and modify_named_credential verbs.

Format

```
emcli get_credtype_metadata
        -auth_target_type=<ttype>
        -cred_type=<name>
```

Options

• **auth_target_type**

Authenticating target type.

• **cred_type**

Credential type.

Examples

```
emcli get_credtype_metadata
        -auth_target_type=host
        -cred_type=HostCreds
```

get_db_account

Shows the database account details as username, profile, account status and authentication type for a given search criteria.

Format

```
emcli get_db_account
        -target_name="tname"
        -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
        [-user_name="user_name"]
        [-target_type="ttype"]
        [-profile=profile default ""]
        [-account_status=account_status default "" ]
        [-authentication_type=auth_type default ""]
        [-limit_rows=limit_rows default 25
        [-noheader]
        [-script | -format=
                        [name:<pretty|script|csv>];
                        [column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];
        ]
        [-input_file="tag1:file_path1;tag2:file_path2;..."]
```

```
        [-validate_only="yes/no" default "no"]
```

`[ ]  indicates that the parameter is optional.`

Options

- target_name

  Name of the target.

- user_name

  Usernames to be searched for. You can include multiple usernames separated by a semicolon (;). The default value of this option is an empty string with all users shown in the result. The user_name is case sensitive.

- target_type

  Type of target. The possible values for target type in this verb are:

  – oracle_database

  – rac_database

  The default value for this argument is oracle_database.

- profile

  Profile of the database account for which you want to search. The default value of this option is an empty string. Users with any profile will be shown in the result.

- account_status

  Account status for which you want to search. The default value of this option is an empty string. Users of any account status will be shown in the result.

- authentication_type

  Authentication type for which you want to search. The default value of this option is an empty string. Users of any authentication type will be shown in the result.

- limit_rows

  Maximum rows to be shown in the result. The default value of this option is 25.

- script

  This option is equivalent to -format="name:script".

- format

  Format specification (default is -format="name:pretty"). You can use the parameter in the following configurations:

  – -format="name:pretty" prints the output table in a readable format but is not intended to be parsed by scripts.

  – -format="name:script" sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings may be specified to change these defaults.

  – -format="name:csv" sets the column separator to a comma and the row separator to a newline.

- no_header

  Displays tabular output without column headers.

- connect_as

Specifies how to connect to the target database. It should be specified in one of the two following formats:

– DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]

– [DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds

- input_file

Use this option is if you want to hide passwords. Specify the path to the file containing the old and new passwords. Each path must be accompanied by a tag that is referenced in the password options.

- validate_only

Indicate whether to validate the options mentioned without doing the actions. Use the following two options:

– Yes - Validate the options mentioned. Do not perform any actions. Through any validation errors on to the console.

– No - Perform the actions as per the mentioned options.

The default value of this option is No.

Output

Output columns: Username Profile Account Status Authorization Type

Examples

Refer to the following examples:

Example 1

Use the following example to get the details of User Admin1 on database myDB using the named credentials SYS_myDB.

```
emcli get_db_account     -target_name=myDB     -user_name=Admin1     -
connect_as="DBNamedCreds:SYS_myDB"
```

Example 2

Use the following example to get details of all accounts with search criteria mentioned using the sysdba user.

```
emcli get_db_account     -target_name=myDB     -profile=DEFAULT     -
account_status=OPEN     -authentication_type=PASSWORD     -
connect_as="DBUserName:sys;DBPassword:welcome;DBRole:sysdba"
```

get_db_profile

Shows the database profile details as profile, resource name, resource type and limit for a given search criteria.

Format

```
emcli get_db_profile
       -target_name="tname"
       -connect_as= "DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"
       [-target_type="ttype"]
       [-profile=profile default ""]
```

```
                        [-resource=resource default "" ]
                        [-limit_rows=limit_rows default 25
                        [-noheader]
                        [-script | -format=
                                     [name:<pretty|script|csv>];
                                     [column_separator:"column_sep_string"];
                                     [row_separator:"row_sep_string"];
                        ]
                        [-input_file="tag1:file_path1;tag2:file_path2;..."]
                        [-validate_only="yes/no" default "no"]
```

[ ] indicates that the parameter is optional.

Options

*   target_name

    Name of the target.

*   profile

    Profiles to be searched for. You can include multiple profiles separated by a semicolon (;). The default value of this option is an empty string with all profiles shown in the result. The profile is case sensitive.

*   target_type

    Type of the target. The possible values for target type in this verb are:

    –   oracle_database

    –   rac_database

    The default value for this argument is oracle_database.

*   resource

    Resource name of the database profile for which you want to search. The default value of this option is an empty string. Profiles with any resource name will be shown in the result.

*   limit_rows

    Maximum rows to be shown in the result. The default value of this option is 25.

*   script

    This option is equivalent to -format="name:script".

*   format

    Format specification (default is -format="name:pretty"). You can use the parameter in the following configurations:

    –   -format="name:pretty" prints the output table in a readable format but is not intended to be parsed by scripts.

    –   -format="name:script" sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings may be specified to change these defaults.

    –   -format="name:csv" sets the column separator to a comma and the row separator to a newline.

*   noheader

    Display tabular output without column headers.

- connect_as

  Connect to the target database as. It should be specified in one of the following two formats:

  – DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]

  – [DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds

- input_file

  Use this option to hide passwords. Specify the path to the file containing the old and new passwords. Each path must be accompanied by a tag that is referenced in the password options.

- validate_only

  Indicates whether to validate the options mentioned without doing the actions. Choose from one of the following options:

  – Yes - Validate the options mentioned. Do not do any actions.

  – No - do the actions as per the mentioned options.

  The default value of this option is No.

Output

Output columns: Profile Resource Name Resource Type Limit

Examples

Refer to the following examples:

Example 1

Use the following example to get details of all resources of a profile DEFAULT using SYS_myDB named credentials.

```
emcli get_db_profile
     -target_name=myDB
     -profile=DEFAULT
     -connect_as="DBNamedCreds:SYS_myDB"
```

Example 2

Use the following example to get the details of resource of profile DEFAULT using sysdba credentials.

```
emcli get_db_profile
     -target_name=myDB
     -profile=DEFAULT
     -resource=CONNECT_TIME
     -connect_as="DBUserName:sys;DBPassword:welcome;DBRole:sysdba"
```

get_db_sys_details_from_dbname

Retrieves the details of an Oracle Database System target from a Database Unique Name.

Format

```
emcli get_db_sys_details_from_dbname
     -db_unique_name="database unique name"
```

Options

- db_unique_name

  Identifies the database unique name of the database target. You can find this name on the Last Collected page of the database target, or you can query for it.

Example

The following example shows how to retrieve the details of the `company_e_commerce` database:

```
emcli get_db_sys_details_from_dbname -db_unique_name="company_e_commerce"
```

get_dbaas_quota

Lists the database quota setup for SSA user roles.

Format

```
emcli get_dbaas_quota
```

Example

The following example successfully retrieves quotas for roles:

```
emcli get_dbaas_quota
```

It displays information similar to the following:

```
ROLE_NAME: SSA_USER_ROLE1 NUMBER_OF_SCHEMA_SERVICES: 99
MEMORY: 99
STORAGE: 99
NUMBER_OF_PLUGGABLE_DATABASES: 99NUMBER_OF_DB_INSTANCES : 99

Quotas for Roles retrieved successfully
```

get_dbaas_request_settings

Lists the database request settings.

Format

```
emcli get_dbaas_request_settings
```

Example

The following example successfully retrieves database request settings:

```
emcli get_dbaas_request_settings
```

It displays information similar to the following:

```
Future Reservation Length  : 2 Months
 Maximum Archive Duration : 10 Weeks
 Default Retirement Period : 1 Years
Request Settings retrieved successfully.
```

get_duplicate_credentials

Gets all the target-scoped named credentials that are the same as the given target-scoped named credential. Duplicate credentials are redundant. Named credentials can be managed better if reused. The same named credential can be reused for all of the usages.

Format

```
emcli get_duplicate_credential
        -cred_name=<cred_name>
        [-cred_owner=<cred_owner>]
```

[ ] indicates that the parameter is optional

Options

*   **cred_name**

    Searches duplicates of this credential.

*   **cred_owner**

    Owner of the credential, which defaults to the current user.

Example

This example gets all of the credentials that are the same as the named credential MyOracleCredential and credential owner Joe.

```
emcli get_duplicate_credential
        -cred_name=MyOracleCredential
        -cred_owner=Joe
```

get_engr_sys_patching_logs

Obtains the diagnostic log files generated by the last patching procedure that completed (either successfully or with errors).

Format

```
emcli get_engr_sys_patching_logs
     -system_target_name="system_target_name"
     -system_target_type="system_target_type"
     -target_name="target_name"
     -target_type="target_type" | -component_type="component_type"
     -log_location="log_location"
     -required parameter
```

[ ] indicates that the parameter is optional.

Options

*   system_target_name

    Specifies the engineered system target name.

*   system_target_type

    Specifies the engineered system target type.

- target_name

    Specifies the target name.

- target_type

    Specifies the target type.

- component_type

    Specifies the engineered system component type.

- log_location

    Specifies a location to save the log files.

Examples

Example 1

The following example obtains the log files generated by the last completed patching procedure for the member target clusteradm0102.example.com:cluster of the engineered system DB Machine slcm12.example.com:oracle_dbmachine:

```
emcli get_engr_sys_patching_logs
    -system_target_name="DB Machine slc12.example.com"
    -system_target_type="oracle_dbmachine"
    -target_name="clusteradm0102.example.com"
    -target_type="cluster"
    -log_location="/tmp/log_file
```

Example 2

The following example obtains the log files generated by the last completed patching procedure for the component Oracle Infiniband Switch of the engineered system DB Machine slcm12.example.com:oracle_dbmachine:

```
emcli get_engr_sys_patching_logs
    -system_target_name="DB Machine slcm12.example.com"
    -system_target_type="oracle_dbmachine"
    -component_type="Oracle Infiniband Switch"
    -log_location="/tmp/log_file"
```

get_executions

Gets a list of executions of a submission using a submission GUID.

Format

```
emcli get_executions
   -instance=<Instance_GUID>
```

Options

- **instance**

    Displays all executions of a submission.

Output Columns

ExecutionGUID, Name, Status

Examples

```
emcli get_executions instance=16B15CB29C3F9E6CE040578C96093F61
```

get_ext_dev_kit

Downloads the Extensibility Development Kit to your local system. This verb has no parameters and only downloads a kit called edk.zip to the directory where you execute the command. After extracting the contents, you can use this kit to develop extensible components (plug-ins) of Enterprise Manager.

Format

```
emcli get_ext_dev_kit
```

Options

None.

get_gold_agent_image_activity_status

Displays the status of a Management Agent gold image activity.

Format

```
emcli get_gold_agent_image_activity_status
        -operation_name="gold_image_operation_name"
        [-noheader]
        [-script | -format=
                          [name:<pretty|script|csv>];
                          [column_separator:"column_sep_string"];
                          [row_separator:"row_sep_string"];
        ]
[ ]  indicates that the parameter is optional.
```

Options

*   operation_name

    Displays the status of a particular Management Agent gold image activity. To view a list of the Management Agent gold image activities, run `emcli list_gold_agent_image_activities`.

*   noheader

    Displays a tabular form of the output without column headers.

*   script

    This option is equivalent to `-format="name:script"`.

*   format

    Specifies the type of the output format. The default value of this option is `-format="name:pretty"`.

    `-format="name:pretty"` displays the output table in a readable format that cannot be parsed by scripts.

`-format="name:script"` sets the default column separator to a tab character and the default row separator to a newline character. You can specify the column_separator and row_separator strings to change these default characters.

`-format="name:csv"` sets the column separator to a comma and the row separator to a newline character.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

The following example displays the status of the Management Agent gold image operation GOLDAGENTIMAGE_CREATE_2013_12_22_12_12_52_535:

```
emcli get_gold_agent_image_activity_status
                           -
operation_name=GOLDAGENTIMAGE_CREATE_2013_12_22_12_12_52_535
```

get_gold_agent_image_details

Displays the platform, plug-in, patch, configuration properties, and Management Agent details of a Management Agent gold image.

Format

```
emcli get_gold_agent_image_details
        -version_name="gold_image_version_name"
        [-platform]
        [-plugin]
        [-patch]
        [-config_properties]
        [-agent]
        [-noheader]
        [-script | -format=
                        [name:<pretty|script|csv>];
                        [column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];
        ]
[ ]  indicates that the parameter is optional.
```

Options

*   version_name

    Specifies the name of the Management Agent gold image version whose details you want to view.

*   platform

    Displays the platform details of the Management Agent gold image.

*   plugin

    Displays the plug-in details of the Management Agent gold image.

*   patch

    Displays the patch details of the Management Agent gold image.

*   config_properties

Displays the configuration properties of the Management Agent gold image.

- agent

  Displays the Management Agent details of the Management Agent gold image.

- noheader

  Displays a tabular form of the output without column headers.

- script

  this option is equivalent to `-format="name:script"`.

- format

  Specifies the type of the output format. The default value of this option is `-format="name:pretty"`.

  `-format="name:pretty"` displays the output table in a readable format that cannot be parsed by scripts.

  `-format="name:script"` sets the default column separator to a tab character and the default row separator to a newline character. You can specify the column_separator and row_separator strings to change these default characters.

  `-format="name:csv"` sets the column separator to a comma and the row separator to a newline character.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

The following example displays the platform, plug-in, and patch details of the Management Agent gold image OPC_AGI_DB_JUL_13:

```
emcli get_gold_agent_image_details
                    -version_name=OPC_AGI_DB_JUL_13
```

get_group_members

Lists the members of the specified group.

Note that targets are only listed once, even though they can be in more than one sub-group of the group.

Format

```
emcli get_group_members
     -name=<name>
     [-type=<group>]
     [-depth=#]
     [-noheader]
     [-expand_non_groups]
     [-script | -format=
              [name:<pretty|script|csv>];
              [column_separator:<column_sep_string>];
              [row_separator:<row_sep_string>];
     ]
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Target name of the group.

- **type**

  Group type: group. Defaults to `group`.

- **depth**

  Lists target members in sub-groups to the depth specified. The default is 1. When the depth is set to 0, no group target members are listed, and only the group's existence is verified. When the depth is set to -1, all group and sub-group target members are listed; in this case no groups appear in the output. Note that a target is listed at most once, even though it can be a member of several sub-groups.

- **noheader**

  Displays tabular information without column headers.

- **expand_non_groups**

  Lists members of aggregates and the aggregate target. By default, only sub-group target members are listed.

- **script**

  This is equivalent to `-format="name:script"`.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Target Name, Target Type

Examples

Example 1

This example lists the unique targets in group `my_group:group` and its sub-groups.

```
emcli get_group_members -name=my_group -depth=-1
```

Example 2

This example lists the unique targets in group my_group:group and its sub-groups/ aggregates. The aggregate targets are also listed.

```
emcli get_group_members -name=my_group -depth=-1 -expand_non_groups
```

get_groups

Lists all groups.

Format

```
emcli get_groups
      [-noheader]
      [-script | -format=
                 [name:<pretty|script|csv>];
                 [column_separator:<column_sep_string>];
                 [row_separator:<row_sep_string>];
      ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **noheader**

  Displays tabular information without column headers.

- **script**

  This is equivalent to -format="name:script".

- **format**

  Format specification (default is -format="name:pretty").

  – format="name:pretty" prints the output table in a readable format not intended to be parsed by scripts.

  – format="name:script" sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – format="name:csv" sets the column separator to a comma and the row separator to a newline.

  – format=column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Target Name, Target Type

Example

This example lists all groups.

```
emcli get_groups
```

get_instance_data

Downloads instance submission data.

Format

```
emcli get_instance_data
       [-instance=<instance_guid>]
       [-exec=<execution_guid>]
       [-name=<execution name>]
       [-owner=<execution owner>]
```

[ ] indicates that the parameter is optional

Options

- **instance**

  Instance GUID.

- **exec**

  Execution GUID.

- **name**

  Execution name.

- **owner**

  Execution owner.

Output

Instance properties data.

Examples

```
emcli get_instance_data -instance=16B15CB29C3F9E6CE040578C96093F61 > data.xml
```

get_instance_status

Displays the procedure instance status identified by the GUID on the command line.

Format

```
emcli get_instance_status
       -instance=<instance_guid>
       [-exec=<execution_guid>]
       [-name=<execution_name>]
       [-owner=<execution_owner>]
       [-xml [-details] [-showJobOutput [-tailLength=<last_n_characters>]]]
```

[ ] indicates that the parameter is optional

Options

- **instance**

  Display the details of a procedure instance identified by the GUID number. You can find the GUID number by using the emcli get_instances command.

- **exec**

  Execution GUID.

- **name**

  Execution name.

- **owner**

  Execution owner.

- **xml**

  Shows the complete status of each of the steps in XML format.

- **details**

  Displays more details for the command output. This option also requires the -xml option.

- **showJobOutput**

  Shows the output or errors for the job execution steps. This option also requires the -xml option.

- **tailLength**

  Limits the number of characters in the job step output or error. This option also requires the -showJobOutput option.

  <Last N Characters> is a positive non-zero number until which the characters are chosen from the end of the job step output. The system sets the maximum permissible characters to dump. If you do not provide this option, the maximum permissible characters are dumped.

Output Columns

GUID, Procedure Type, Instance Name, Status

Status Values

Possible status/return values are as follows:

SUCCEEDED

FAILED

EXECUTING

COMPLETED

Examples

Example 1

This example shows procedure details in CSV format:

```
emcli get_instance_status -guid=123456789012345678901234566789012
```

Example 2

This example shows details in XML format with the last 1024 characters of output:

```
 emcli get_instance_status -guid=16B15CB29C3F9E6CE040578C96093F61 -xml -showJobOutput -
tailLength=1024
```

See Also

get_instances
get_job_execution_detail

get_instances

Displays a list of procedure instances.

Format

```
emcli get_instances
      [-type=<procedure_type>]
```

[ ] indicates that the parameter is optional

Options

- **type**

  Displays all the procedure instances of type procedure_type.

Output Columns

Instance GUID, Execution GUID, Procedure Type, Instance Name, Status

Examples

Example 1

This example lists all procedure instances:

```
emcli get_instances
```

Example 2

This example lists all procedure instances of type 'PatchOracleSoftware':

```
emcli get_instances -type=PatchOracleSoftware
```

See Also

get_procedure_types

get_internal_metric

Gets the value of an internal metric from the specified OMS. This verb obtains metric values for any of the internal metrics returned by the list_internal_metrics verb.

Format

```
emcli get_internal_metric
      -metric_name=<metric name>
      [-script | -format=
                        [name:<pretty|script|csv>];
                        [column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];
      [-oms_name=<specific oms name> ]
      ]
```

[ ] indicates that the parameter is optional.

Options

- metric_name

  The name of the internal metric whose value you want to extract from the OMS. A list of internal metrics can be obtained using the list_internal_metrics verb.

- oms_name

  The name of the target OMS. The explicit OMS name can be found in the Cloud Control console Management Services page. To navigate to this page, from the Setup menu select Manage Cloud Control and then Management Services. In the Servers area, look for the full name of the Management Service (<host name>:<port number>_Management_Service).

  Note: You only need to specify the oms_name option if you are attempting to access a specific OMS in a multi-OMS environment. If you omit the oms_name option, the get_internal_metric verb will access the OMS running the current instance of EMCLI.

Examples

Example 1

The following example extracts metric values for the metric "pbc_WorkManagerStatistics" from the OMS named "myserver.myco.com:17999_Management_Service".

```
emcli get_internal_metric -metric_name=pbs_WorkManagerStatistics -
oms_name=myserver.myco.com:17999_Management_Service
```

Example 2

The following example extracts metric values for the metric "pbc_WorkManagerStatistics" from the OMS currently running EMCLI.

```
emcli get_internal_metric -metric_name=pbs_WorkManagerStatistics
```

get_job_execution_detail

Displays details of a job execution.

Format

```
 emcli get_job_execution_detail      -execution=<"execution_id">      [-xml [-
showOutput [-tailLength=<"length">]]]
```

[ ] indicates that the parameter is optional

Options

- **execution**

  Specifies that the ID of the job execution (execution_id) is the job execution ID.

- **xml**

  Shows the execution details as XML.

- **showOutput**

Shows the output of the steps inside the job execution. You can only use this option in conjunction with the -xml option.

- **tailLength**

    Limits the display of the output to the number of characters from the end of the output. (length) is in characters. You can only use this option in conjunction with the -showOutput option. If you do not specify this option, a system-generated hard limit is enforced.

Examples

Example 1

This example shows the details in CSV format:

```
emcli get_job_execution_detail -execution=123456789012345678901234 5678901
```

Example 2

This example shows the details in XML format with complete output:

```
emcli get_job_execution_detail -execution=1234567890123456789012345 6789012 -xml -showOutput
```

get_job_types

Lists all the job types that can be used to create jobs, library jobs, and multi-task jobs from EM CLI.

EM CLI supports the following job types:

```
ASMSQLScript
ASSOCIATE_CS_FA
ASSOCIATE_DOMAIN_FA
AssociateClusterASM
BlockAgent
CoherenceCacheAddition
CoherenceNodesRefresh
Config Log Archive Locations
DbMachineDashboard
DiscoverPDBEntities
FusionMiddlewareProcessControl
GlassFishProcessControl
InstallKernelModuleJob
Log Rotation
OSCommand
OpatchPatchUpdate_PA
RMANScript
RefreshFromEMStore
RefreshFromMetalink
RefreshFusionInstance
SOABulkRecovery
SQLScript
ShutdownDB
StartDepartedCohNodes
StartDepartedCohStoreNodes
StartFusionInstance
StartupDB
StatspackPurge
StopFusionInstance
```

```
Upgrade Exalogic Systems
WebLogic Control
WebLogic Domain Discover
WebLogic Domain Refresh
```

Format

```
emcli get_job_types   [-type="job_type_pattern"]   [-target_type="target_type"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **type**

  Job type internal name pattern. Specify all or part of the job type name.

- **target**

  Target type on which the job type will run.

get_jobs

Lists existing jobs.

Standard Mode

```
emcli get_jobs
        [-name="job_name_pattern"]
        [-owner="job_owner"]
        [-job_ids="ID1;ID2;..."]
        [-targets="type1:name1;type2:name2;..."]
        [-status_ids="status1;status2;..."]
        [-noheader]
        [-script | -format=
                        [name:<pretty|script|csv>];
                        [column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];
        ]
```

```
[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
get_jobs
      [(name="job_name_pattern"]
      [,owner="job_owner"]
      [,job_ids="ID1;ID2;..."]
      [,targets="type1:name1;type2:name2;..."]
      [,status_ids="status1;status2;..."]
      [,noheader=True|False]
      [,script=True|False | ,format=
                [name:<pretty|script|csv>];
                [column_separator:"column_sep_string"];
                [row_separator:"row_sep_string"];
      ])
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Job name pattern to filter on.

- **owner**

  Owner of the jobs to filter on.

- **job_ids**

  Lists job IDs to use as the output filters.

- **targets**

  Lists targets (as name-type pairs) to use as the output filters.

- **status_ids**

  Lists numeric status IDs to use as the output filters.

  The numeric codes for all possible job statuses are as follows:

  - `ABORTED(Error)=3`
  - ACTION_REQUIRED_STATUS=22
  - `COMPLETED(Successful)=5`
  - `EXECUTING(Running)=2`
  - `FAILED=4`
  - `INACTIVE=14`
  - MISSING_CREDS_STATUS=21
  - `QUEUED=15`
  - REASSIGNED_STATUS=20
  - `SCHEDULED=1`
  - SKIPPED=18
  - `STOPPED=8`
  - `STOP_PENDING=12`
  - `SUSPENDED_AGENT_DOWN=7`
  - `SUSPENDED_BLACKOUT=11`
  - `SUSPENDED_EVENT=10`
  - `SUSPENDED_LOCK=9`
  - `SUSPEND_PENDING=13`
  - `SUSPENDED_USER=6`
  - `TARGET_NOT_READY_STATUS =26`

- **noheader**

  Displays tabular information without column headers.

- **script**

This is equivalent to `-format="name:script"`.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=column_separator:"column_sep_string"` column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - `row_separator:"row_sep_string"` row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Name, Type, ID, Execution ID, Scheduled, TimeZone, Completed, Status, Status ID, Owner, Target Type, Target Name

Examples

These examples show the jobs with the specified job IDs 12345678901234567890123456789012 and 09876543210987654321098765432100:

Example 1 - Command-Line

```
emcli get_jobs
     -job_ids="12345678901234567890123456789012;09876543210987654321098765432100"
```

Example 2 - Script and Interactive

```
get_jobs
    (job_ids="12345678901234567890123456789012;
       09876543210987654321098765432100")
```

These examples show all jobs run against a host target named `mainhost.example.com` that are scheduled or have completed.

Example 3 - Command-Line

```
emcli get_jobs
     -status_ids="1;5"
     -targets="mainhost.example.com:host"
```

Example 4 - Script and Interactive

```
get_jobs
    (status_ids="1;5",
     targets="mainhost.example.com:host")
```

get_mda_engine_status

Provides the current status of the Middleware Diagnostics Advisor (MDA) engine.

Format

```
emcli get_mda_engine_status
```

get_metering_data

Gets usage details.

Format

```
emcli get_metering_data
     [-start_date=<start_date_in_mmddyyyy>]
     [-end_date=<end_date_in_mmddyyyy>]
     [-charge]
     [-cost_center=<cost_center_name>]
     [-target_type=<target_type>
     [-target_name=<target_name>]

[ ]  indicates that the parameter is optional
```

Options

- **start_date**

  Report cycle start date in mmddyyyy. If you do not specify the report cycle start date, the latest report cycle is used.

- **end_date**

  Report cycle end date in mmddyyyy. If you do not specify the report cycle end date, the latest report cycle is used.

- **charge**

  Prints charge relation information.

- **cost_center**

  Cost center name. If you do not specify the cost center name, the logged in user is used as the cost center name.

- **target_type**

  If you do not specify the target type, all targets are used. Supported target types for this release are oracle_database, oracle_vm_guest, host, and weblogic_j2eeserver. This option is not valid without the target_name parameter.

- **target_name**

  If you do not specify the target name, all targets of a given target type are used. this option is not valid without the target_type parameter.

Examples

Example 1

This example shows usage data for the cost center cost_center_internal_name for the report cycle with a starting date of 10012011.

```
emcli get_metering_data    -start_date=10012011    -
cost_center=cost_center_internal_name
```

Example 2

This example shows charge data for the my_target Oracle Guest VM target for cost center cost_center_internal_name for a report cycle with a starting date of 10012011.

```
emcli get_metering_data
    -start_date=10012011
    -cost_center=cost_center_internal_name
    -target_type=oracle_vm_guest
    -target_name=my_target
    -charge
```

get_metrics_for_stateless_alerts

For the specified target type, lists the metrics whose alerts are stateless and thus can be manually cleared. Both the metric name and metric internal name are provided in the output of this command. To clear the stateless alerts associated with the specified metric, use the clear_stateless_alerts verb.

Format

```
emcli get_metrics_for_stateless_alerts      -target_type=type
```

Options

- **target_type**

  Internal target type identifier, such as host, oracle_database, oc4j, oracle_emrep, and oracle_emd.

Examples

This example provides a list of all metrics for which stateless alerts can be manually cleared for any Oracle database (internal name for the target type is oracle_database).

```
emcli get_metrics_for_stateless_alerts -target_type=oracle_database
```

get_named_credential

Displays named credential details.

Standard Mode

```
emcli get_named_credential
      -cred_owner=<owner>
      -cred_name=<name>
      -out=<filename>
```

Interactive or Script Mode

```
get_named_credential
      (cred_owner=<owner>
      ,cred_name=<name>
      ,out=<filename>)
```

Options

- **cred_owner**

  Owner of the credential.

- **cred_name**

  Required credential name.

- **out**

  Output file name. The same file can be used as the input properties file for create_named_credential and modify_named_credential.

Examples

These examples display the details of the named credential NC1 owned by the current logged in user.

Example 1 - Command-Line

```
emcli get_named_credential
     -cred_name=NC1
```

Example 2 - Scripting and Interactive

```
get_named_credential
       (cred_name="NC1")
```

These examples display the details of the named credential NC2 owned by the Administrator CREDS_MGR.

Example 3 - Command-Line

```
emcli get_named_credential
     -cred_name=NC2
     -cred_owner=CREDS_MGR
```

Example 4 - Scripting and Interactive

```
get_named_credential
     (cred_name="NC2"
     ,cred_owner="CREDS_MGR")
```

get_not_updatable_agents

Displays the Management Agents that cannot be updated.

Format

```
emcli get_not_updatable_agents
     [-version_name  | -image_name]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- version_name

  Parameter to display the Management Agents that cannot be updated using the specified Management Agent gold image version.

- image_name

  Parameter to display the Management Agents that cannot be updated using the latest Management Agent gold image version of the specified image name.

Examples

Example 1

The following example displays the Management Agents that cannot be updated using the latest Management Agent gold image version of the image `OPC_AGT_ADC_POD`.

```
emcli get_not_updatable_agents
     -image_name="OPC_AGT_ADC_POD"
```

Example 2

The following example displays the Management Agents that cannot be updated using the Management Agent gold image version `OPC_AGT_ADC_POD_JUNE`.

```
emcli get_not_updatable_agents
     -version_name="OPC_AGT_ADC_POD_JUNE"
```

get_oms_config_property

Gets the property value corresponding to the specified property name.

Format

```
emcli get_oms_config_property
       -property_name="propertyName"
       [-oms_name="omsName"]
       [-details]
```

```
[ ]  indicates that the parameter is optional
```

Options

* **property_name**

  Name of the property whose value must be retrieved.

* **oms_name**

  Name of the mangaement server for which the property must be retrieved.

* **details**

  Specifies details about from where the property value has been derived, and also the global and default values for the property.

Examples

Example 1

This example retrieves the property value set for the property name "propName" from the management server myhost:1159_Management_Service.

```
get_oms_config_property -property_name=propName -
oms_name="myhost:1159_Management_Service"
```

Example 2

This example retrieves the property value set for the property name "propName" from all the management servers with details.

```
get_oms_config_property -property_name=propName -details
```

get_oms_inventory

Displays the OMS version, plug-in details and patches applied on each home.

Format

```
emcli get_oms_inventory
      [-xml]|[-map]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- xml

  Displays the output in xml format.

- map

  Diplays the output in name:value format

get_oms_logging_property

Gets the property value corresponding to the specified logging property name.

Format

```
emcli get_oms_logging_property
      -property_name="propertyName"
      [-oms_name="omsName"]
      [-details]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **property_name**

  Name of the logging property whose value must be retrieved.

- **oms_name**

  Name of the mangaement server for which the property must be retrieved.

- **details**

  Specifies details about from where the property value has been derived, and also the global and default values for the logging property.

Examples

Example 1

This example retrieves the property value set for the property name "propName" from the management server myhost:1159_Management_Service.

```
get_oms_logging_property -property_name=propName -
oms_name="myhost:1159_Management_Service"
```

Example 2

This example retrieves the property value set for the property name "propName" from all the management servers.

```
get_oms_logging_property -property_name=propName
```

get_on_demand_metrics

Gets a list of metrics that can be immediately collected with the collect_metric EM CLI verb. From this list, identify the metric you are interested in under the Metric Name column, then use its corresponding Metric Internal name in the collect_metric verb.

Format

```
emcli get_on_demand_metrics       -target_type=type
      -target_name=name
```

Options

- **target_type**

  Internal target type identifier, such as host, oracle_database, oc4j, oracle_emrep, and oracle_emd.

- **target_name**

  Name of the target.

Examples

This example shows a list of collectible metrics for the host target called hostname.example.com.

```
emcli get_on_demand_metrics -target_type=host -target_name=hostname.example.com
```

get_onetime_registration_token

Generates an agent registration token for one-time use.

Format

Standard Mode

```
emcli get_onetime_registration_token
[-validity="number of minutes"]
```

Interactive or Script Mode

```
get_onetime_registration_token([validity="number of minutes"]
)

[ ]  indicates that the parameter is optional.
```

Options

- validity1

  Number of minutes the registration token is valid. The default validity is 15 minutes. The maximum validity allowed is 720 minutes.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

The following command creates a one time registration token with validity of 25 minutes.

```
emcli get_onetime_registration_token
      -validity=25
```

Example 2

The following command creates a one time registration token with validity of 15 minutes.

```
emcli get_onetime_registration_token
```

get_operation_plan_details

Provides detailed step-by-step information about the specified operation plan.

Format

```
emcli get_operation_plan_details
      -name="plan name"
```

Options

- **name**

  Name of the operation plan.

Examples

```
emcli get_operation_plan_details
      -name="BISystem1-switchover"
```

See Also

create_operation_plan
get_operation_plans

get_operation_plans

Lists all configured operation plans.

Format

```
emcli get_operation_plans
      -name=<operation plan_name>
      -operation=<operation_name>
```

Options

- **name**

Name of the operation plan.

- **operation**

  Name of the operation, such as switchover, failover, start, or stop.

Output Columns

Plan Name, Operation Name, Configuration GUID

Examples

```
emcli get_operation_plans
     -name="austin-switchover"
     -operation="switchover"
```

See Also

submit_operation_plan

create_operation_plan

get_organizational_hierarchy

Retrieves the organizational hierarchy of departments, lines-of-business, and cost centers for the specified organizational entity.

Standard Mode

```
emcli get_organizational_hierarchy
     -entity_name="entity name"
     [-tenant_name="tenant name"]
```

[ ] indicates that the parameter is optional.

Interactive or Script Mode

```
emcli get_organizational_hierarchy
     (entity_name="entity name"
     [,tenant_name="tenant name"])
```

[ ] indicates that the parameter is optional.

Options

- entity_name

  Name of the organizational entity whose hierarchy to retrieve.

- tenant_name

  Specifies the name of the tenant to which the specified organizational entity belongs. Default is the tenant of the logged-in user.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example retrieves the organizational hierarchy for an entity named investments that belongs to the Capital Gains tenant.

```
emcli get_organizational_hierarchy
      entity_name="investments"
      tenant_name="Capital Gains"
```

get_paas_zone_detail

Retrieves the PaaS Infrastructure Zone details.

Format

```
emcli get_paas_zone_detail
      -name="<Name of PaaS Zone>"
```

Options

- **name**

  Name of the existing PaaS Infrastructure Zone

Example

This example retrieves the PaaS Infrastructure Zone details for My PaaS Zone:

```
emcli get_paas_zone_detail
    -name="My PaaS Zone"
```

It displays the following information:

```
Name                          My PaaS Zone
Description                   This is a test PaaS Zone
Named Credentials             ZoneNamedCredentials
Number of Hosts               2
Roles                         CLOUD_ADMIN_ROLE
Maximum Memory Allocation (%) 75
Maximum CPU Utilization (%)   85
```

> **Note:**
>
> To retrieve the members of this PaaS Infrastructure Zone, run:
>
> ```
> emcli get_system_members -name="My PaaS Zone" -type="self_service_zone"
> ```

get_patch_plan_data

Gets patch plan user-editable data.

Format

```
emcli get_patch_plan_data
        -name="name"
```

Options

- **name**

  Name of a given patch plan.

Example

get_siteguard_script_credential_params

```
emcli get_patch_plan_data -name="plan_name"
```

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data
show_patch_plan
submit_patch_plan

upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

get_plugin_deployment_status

Displays the status of a specific plug-in deployment or undeployment activity as well as the list of steps.

Format

```
emcli get_plugin_deployment_status
      [-plugin="plugin_id"]
      [-destination_type=(agent|server)]

[ ]  indicates that the parameter is optional
```

Options

- **plugin**

  ID of the plug-in for which you need to view the deployment/undeployment status. If not provided, the command shows the status of the latest plug-in being deployed, or the last one that was deployed or undeployed.

- **destination_type**

  Specifies the destination type. Can be either 'agent' or 'server'. Enables you to view the latest deployment status for the given plug-in only on the server side or only on the agent side. If not specified, defaults to the latest deployment or un-deployment performed for this plug-in, regardless of whether it is server side or agent side.

Examples

Example 1

Displays the status of the last plug-in deployment/undeployment activity.

```
emcli get_plugin_deployment_status
```

Example 2

This example displays the status of the last deployment/undeployment activity of a specific plug-in.

```
emcli get_plugin_deployment_status
     -plugin=oracle.sysman.db
```

get_pool_allowed_placement_constraints

Retrieves the list of placement constraints for a pool target type.

Format

```
emcli get_pool_allowed_placement_constraints
     -target_type="<Target type of Software Pool>"
```

Options

- **target_type**

  Target type of the software pool.

Example

The following example retrieves the list of placement constraints for the mwaas_zone target type:

```
emcli get_pool_allowed_placement_constraints
     -target_type="mwaas_zone"
```

It displays the following output:

```
Name                      Description
MAX_INSTANCES             Maximum Number of Java Servers (per host)
```

get_pool_capacity

Retrieves the capacity details for a software pool including CPU utilization, memory allocation, and number of instances per host.

Format

```
emcli get_pool_capacity
    -name="<Software Pool name>"
    -target_type="<Target type of Software Pool>"
```

Options

- **name**

  Name of an existing Software Pool.

- **target_type**

  Target type of the Software Pool.

get_pool_detail

Retrieves details for a software pool.

Format

```
emcli get_pool_detail
     -name="<Software Pool name>"
     -target_type="<Target type of Software Pool>"
```

Options

- **name**

  Name of an existing software pool.

- **target_type**

  Target type of the software pool.

Example

This example retrieves details for the MyPool software pool:

```
emcli get_pool_detail
       -name="My Pool"
       -target_type="mwaas_zone"
```

It displays the following information:

```
Name                      My Pool
Target Type               mwaas_zone
Description               This is a test Pool
Paas Infrastructure Zone  My PaaS Zone
Number of Members         1
Placement Constraints     MAX_INSTANCES : 25
Member Constraints        VERSION : 10.3.5.0
```

> **Note:**
>
> To retrieve the members of this software pool, run:
>
> ```
> emcli get_system_members -name="My Pool" -type="mwaas_zone"
> ```

get_pool_filtered_targets

Retrieves the filtered targets available for software pool creation based on the given criteria.

Format

```
emcli get_pool_filtered_targets
     -target_type="<Target type of Software Pool>"
     -paas_zone="<Paas Infrastructure Zone of Software Pool>"
```

```
               [-member_constraints="<constraint1=value1, constraint2=value2>"
       ]
```

[] indicates that the parameter is optional.

Options

- **target_type**

  Target type of the Software Pool.

- **paas_zone**

  Name of PaaS infrastructure zone within which the filtered targets are to be retrieved.

- **member_constraints**

  Comma separated key value pairs that restrict the addition of member targets to a software pool with a set criteria.

Example

The following example retrieves the list of allowed possible member constraints for a pool target type:

```
emcli get_pool_allowed_member_constraints -target_type=<Target type>
```

get_procedure_types

Gets the list of all deployment procedure types.

Format

```
emcli get_procedure_types
```

Output Column

Procedure Type

Example

This example lists all procedure types:

```
emcli get_procedure_types
```

get_procedure_xml

Gets the deployment procedure XML file. XML is printed on standard output.

Format

```
emcli get_procedure_xml
       -procedure=[procedure_guid]
       [-name=<procedure_name>]
       [-owner=<procedure_owner>]

[ ]  indicates that the parameter is optional
```

Options

- **procedure**

  Procedure GUID.

- **name**

  Procedure name.

- **owner**

  Procedure owner.

Output

Deployment procedure XML.

Examples

```
emcli get_procedure_xml -procedure=16B15CB29C3F9E6CE040578C96093F61 > proc.xml
```

get_procedures

Gets a list of deployment procedures and pre-saved procedure configurations.

Format

```
emcli get_procedures [-type=<procedure_type>]
      [-parent_proc=<procedure_associate>]

[ ]  indicates that the parameter is optional
```

Options

- **type**

  Displays all the deployment procedures of type procedure_type.

- **parent_proc**

  Procedure associated with procedure configurations.

Output Columns

GUID, Procedure Type, Name, Display Type, Version, Created By, Procedure Name

See Also

get_procedure_types
get_procedure_xml

get_reports

Returns a list of Information Publisher reports owned by or viewable by all users or a specified user. The output of this report is space-separated, quoted strings for the report title and owner, with each report on its own line.

Format

```
emcli get_reports   [-owner="<report_owner>"]

[ ]  indicates that the parameter is optional
```

Options

*   **owner**

    Enables listing of viewable reports that a specific Enterprise Manager owns.

Output

Space-separated quoted strings for the report title and owner, with each report on its own line.

Examples

```
emcli get_reports -owner=username
"report 1","username"
"example report 2","username"


emcli get_reports
"report A","username1"
"report 1","username2"
"example report 2","username2
```

get_resolution_states

Gets the list of existing resolution states used in managing incidents and problems. It also prints the display position of states. It does not list the fixed "New" and "Closed" resolution states.

Format

```
emcli get_resolution_states
```

Options

None.

Examples

This example shows sample output for Incident defined states of OnHold, Waiting, and Processed, and Problem defined states of OnHold and Processed.

```
Incident resolution states
    5     OnHold
   10     Waiting
   25     Processed

Problem resolution states
    5     OnHold
   25     Processed
```

get_retry_arguments

Get arguments of failed steps that can be retried.

Format

```
emcli get_retry_arguments
      [-instance=<instance_guid>]
      [-exec=<execution_guid>]
      [-name=<execution_name>]
      [-owner=<execution_owner>]
      [-stateguid=<state_guid>]

[ ]  indicates that the parameter is optional
```

Options

* **instance**

  Instance GUID.

* **exec**

  Execution GUID.

* **name**

  Execution name.

* **owner**

  Execution owner.

* **stateguid**

  State GUID.

Examples

```
emcli get_retry_arguments -instance=16B15CB29C3F9E6CE040578C96093F61
```

```
emcli get_retry_arguments -instance=16B15CB29C3F9E6CE040578C96093F61 -
stateguid=51F762417C4943DEE040578C4E087168
```

get_runtime_data

Downloads the execution run-time properties data. The execution can be retrieved by using the instance GUID, execution GUID, or a name value pair.

Format

```
emcli get_runtime_data
      [-instance={instance_guid}]
      [-exec={execution_guid}]
      [-name={execution name}]
      [-owner={execution owner}]

[ ]  indicates that the parameter is optional.
```

Options

- **instance**

  Instance GUID.

- **exec**

  Execution GUID.

- **name**

  Execution name.

- **owner**

  Execution owner.

> **Note:**
>
> The name and owner parameters must be used together.

Example

This example displays the execution run-time properties data.

```
emcli get_runtime_data -exec=16B15CB29C3F9E6CE040578C96093F61 > data.xml
```

get_sample_migration_xml

Generates a sample XML migration file at the specified directory location that demonstrates source and destination mappings, based on the chosen migration method.

Format

```
emcli get_sample_migration_xml
      -migration_type=<migration type>
      [-file_name=<file name>]
      [-directory=<directory path>]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- migration_type

  Specifies the migration method, as follows:

  – ONLINE_DATAGUARD: Data Guard Physical Standby (minimal downtime)

  – RMAN_CLONE: RMAN Clone

  – OFFLINE_DATAPUMP: Data Pump (full or schema) Export and Import (cross-platform)

  – ONLINE_TTS: Full Transportable Export and Import (minimal downtime, cross-platform)

- file_name

Name of the sample migration file. Defaults to xml if no extension is specified. Defaults to SampleMigrationXML.xml if no file name is specified.

- directory

  Where to save the sample file. Defaults to the current directory if no directory is specified.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

The following example generates a sample XML migration file named SampleMigrationXML.xml in the current directory. The sample migration uses the Data Guard migration method.

```
emcli get_sample_migration_xml
     -migration_type="ONLINE_DATAGUARD"
```

Example 2

The following example generates a sample XML migration file named RMANMapping.xml in the /scratch/migrations directory. The sample migration uses the RMAN Clone migration method.

```
emcli get_sample_migration_xml
     -migration_type="RMAN_CLONE"
     -file_name="RMANMapping"
     -directory="/scratch/migrations"
```

get_saved_configs

Lists the saved configurations.

Format

```
emcli get_saved_configs
      [-target_type="<target_type>"]
      [-target_name="<target_name>"]
      [-owner="<owner>"]
      [-format=name:<pretty|script|csv>;
      [column_separator:"column_sep_string"];
      [row_separator:"row_sep_string"];]

[ ]  indicates that the parameter is optional
```

Options

- **target_type**

  Internal type name, such as oracle_database for "Oracle Database." You can use the get_target_types command to get the internal name for a target type.

- **target_name**

  Name of the target.Either specify the complete name or a pattern match using "%".

- **owner**

Owner of the saved configuration.

This can be a full value or a pattern match using "%".

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  – format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – format="name:script;row_separator:<row_sep_string>" row-separates the verb output by <row_sep_string>. Columns are separated by the tab character.

- **noheader**

  Display tabular output without column headers.

Output Columns

Name (Saved configuration name, the concatenation of target name, target type and saved time in YYYYMMDDHH24MISS format), Target Type, Target Name, Saved Time (Format of the time is: yyyy/MM/dd HH:mm), Time Zone, Owner, Description

Examples

Example 2

This example lists all of the saved configurations created on target type "host" and target name "test host":

```
emcli get_saved_configs -target_type="host" -target_name="test host"
```

Example2

The example lists all of the saved configurations created by user with name "test user" and created on target type "host" and target name "test host":

get_service_template_detail

Retrieves the Service Template details.

Format

```
emcli get_service_template_detail
        -name="<Service_Template_name>"
        -service_family="<Service_family_name>"

[ ]  indicates that the parameter is optional
```

Options

*   **name**

    Name of the existing Service Template.

*   **service_family**

    Service family to which the Service Template belongs. Examples: DBAAS for Database, and MWAAS for Middleware.

Examples

```
emcli clear_problem
      -problem_key="ORA-600"
      -target_type="oracle_database"-preview
```

displays the following output:

```
Name Middleware service template August
Service Family MWAAS
Description Middleware small instance service template
Roles CLOUD_USER_ROLE_1
Software Pools mwaas_zone:pool1
Configurations
{
"type" : "CDP"
"serviceFamily" : "MWAAS",
"payloads": null,
"configurations" : [ {
"name" : "WebLogic Username *",
"value" : "weblogic",
"id" : "wlsUserName",
"displayName" : null,
"description" : "Username for the WebLogic Server",
"values" : null,
"required" : false,
"secret" : false,
"subconfigurations" : null
}, {
"name" : "WebLogic Password *",
"value" : "Welcome_123",
"id" : "wlsUserPassword",
"displayName" : null,
"description" : "Password for the WebLogic Server",
"values" : null,
"required" : false,
"secret" : true,
"subconfigurations" : null
},{
"name" : "Topology",
"value" : "1",
"id" : "topology",
"displayName" : null,
"description" : "Enter 1 for single cluster, 0 for no cluster. For physical
provisioning it is auto populated based on the profile selected. For virtual
provisioning it is defaulted to 1. Please change based on the actual topology of the
assembly. ",
"values" : null,
"required" : false,
"secret" : false,
```

```
"subconfigurations" : null
},
```

Note that all configurations are not shown in the example above.

get_service_templates

Lists the available service templates.

Format

```
emcli get_service_templates
        [-service_family="<Service_family_name>"]
```

`[ ]  indicates that the parameter is optional`

Options

- **service_family**

  Service family name used for filtering the service templates. Example: DBAAS for
  Database, and MWAAS for Middleware

get_signoff_agents

Shows the available Agents for sign-off.

If you do not specify any options, the command shows all Agents available for sign-off.
If you specify more than one option, the command shows the union of Agents
available for sign-off belonging to each option passed.

Format

```
emcli get_signoff_agents
      [-agents="List_of_agents"]
      [-platforms="List_of_platforms"]
      [-versions="list_of_versions"]
      [-groups="list_of_group_names"]
      [-output_file="location_of_output_file"]
```

`[ ]  indicates that the parameter is optional`

Options

- **agents**

  List of Agents for sign-off matching Agent names or Agent names pattern
  separated by commas.

- **platforms**

  Lists Agents available for sign-off on the specified platforms.

- **versions**

  Lists Agents available for sign-off with the specified version.

- **groups**

  Lists Agents available for sign-off belonging to the specified groups.

- **output_file**

Adds the Agents into the output file, which can be submitted for a clean-up job to remove old Oracle Management Agent homes and old Oracle home targets, and back up directories of upgraded Oracle Management Agents.

Examples

Example 1

This example shows the list of Agents for clean up that match the Agents specified in the option.

```
emcli get_signoff_agents -agents="abc%,xyz.domain.com:1243"
```

Example 2

This example shows the list of Agents for clean up that match the platform specified in the option.

```
emcli get_signoff_agents -platforms="Linux x86,Microsoft Windows x64 (64-bit)"
```

get_signoff_status

Shows Agent sign-off results.

Format

```
emcli get_signoff_status
      [-agent="full_agent_name"]
      [-job_name="job_name"]
      [-status="status"]

[ ]  indicates that the parameter is optional
```

Options

- **agent**

  Shows the sign-off job details of the specified Agent names or Agent names pattern separated by commas.

- **job_name**

  Shows the sign-off job details of the specified job name.

- **status**

  Shows the sign-off job details of the specified status.

Permutations for combinations of parameters are as follows:

No parameters — Shows <JOB NAME, JOB STATUS, NUMBER OF AGENTS IN THE JOB, JOB START TIME, JOB END TIME> for each job.

-job_name — Shows <AGENT_NAME, STATUS OF JOB, START TIME, END TIME> for each Agent in the job, where tje job name is passed in the -job_name parameter.

-status only — Shows <JOB NAME, NUMBER OF AGENTS IN THE JOB, JOB START TIME, JOB END TIME> for each job, where the job status is passed in -status parameter.

-agent only — Shows <JOB NAME, STATUS OF JOB, START TIME, END TIME> for each job, where the Agent is present and the Agent name is passed in the -agent parameter.

-job_name and -agent only — Shows <JOB STEP NAME, JOB STEP STATUS, JOB STEP START TIME, JOB STEP END TIME> for each step in the job for the Agent passed in -job_name , -agent parameter

-job_name, -agent, and -status — Shows <JOB STEP NAME, JOB STEP START TIME, JOB STEP END TIME> for each step in the job for the Agent having step status passed in -job_name , -agent , and -status respectively.

-job_name and -status — Shows <AGENT_NAME, START TIME, END TIME> for each Agent in the job having an Agent upgrade status passed in -job_name and -status respectively.

-agent and -status — Shows <JOB NAME, START TIME, END TIME> for each job having the Agent and clean-up status passed in -agent and -status respectively.

Examples

Example 1

This example shows the sign-off job details for agent xyz.domain.com:1243 .

```
emcli get_signoff_status -agent=xyz.domain.com:1243
```

Example 2

This example shows the sign-off job details with the status Success.

```
emcli get_signoff_status -status="Success"
```

get_siteguard_credential_association

Lists the credential associations configured for a system.

Format

```
emcli get_siteguard_credential_association
     [-system_name=<name_of_system>]
     [-target_name=<name_of_target>]
     [-credential_type=<type_of_credential>]

[ ]  indicates that the parameter is optional
```

Options

- **system_name**

  Name of the system.

- **target_name**

  Name of the target.

- **credential_type**

  Type of the credential, which can be HostNormal, HostPrivileged, WLSAdmin, or DatabaseSysdba.

Output Columns

Target Name, Credential Name, Credential Type

Examples

Example 1

```
emcli get_siteguard_credential_association
      -system_name="austin-system"
      -credential_type="HostNormal"
```

Example 2

```
emcli create_siteguard_credential_association
      -system_name="austin-system"
      -target_name="austin-database-instance"
      -credential_type="HostNormal"
```

See Also

create_siteguard_credential_association
update_siteguard_credential_association

get_siteguard_health_checks

Displays the schedule of health checks for an operation plan.

Format

```
emcli get_siteguard_health_checks
        [-operation_plan="name_of_the_operation_plan"]
```

```
[ ]  indicates that the parameter is optional
```

Parameter

operation_plan

Name of the operation plan for which health checks have been scheduled.

Example

This example displays information about the health checks scheduled on a system for the `austin-switchover` operation plan:

```
emcli get_siteguard_health_checks
        -operation_plan="austin-switchover"
```

get_siteguard_lag

Retrieves and shows the configured limit for the Apply lag and Transport lag for all or selected databases of the system.

Format

```
emcli get_siteguard_lag
        [-system_name="name_of_the_system"]
        [-target_name="name_of_the_target_database"]
        [-property_name="lag_type"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **system_name**

  Name of the system whose configuration details you want to view.

- **target_name**

  Name of the database whose lag configuration details you want to view.

- **property_name**

  Name of the lag property configured. Valid values are `ApplyLag` and `TransportLag`.

Examples

Example 1

This example displays the details of the Apply lag limit configured on all of the databases of the system `austin-system`:

```
emcli get_siteguard_lag
      -system_name="austin-system"
      -property_name="ApplyLag"
```

Example 2

This example displays the details of the Transport lag limit configured on the database `OID-db` of `austin-system`:

```
emcli get_siteguard_lag
      -system_name="austin-system"
      -target_name="OID_db"
      -property_name="TransportLag"
```

get_siteguard_script_credential_params

Retrieves all credentials parameters for a Site Guard script.

Format

```
emcli get_siteguard_script_credential_params     -script_id="Id associated
with the script"          -credential_name="name of the credential"          [-
credential_owner="credential owner"]

[ ]  indicates that the parameter is optional.
```

Options

- script_id

  The script ID.

- credential_name

  Name of the credential. If this option is not specified, all credentials associated as parameters for the script will be listed.

- credential_owner

The owner of the credential. If this argument is not specified, all credentials associated as parameters for the script will be listed.

Examples

Example 1

The following command retrieves the Site Guard credential parameters for the script with the ID 1 and name NAMED_CREDENTIAL_X.

```
emcli get_siteguard_script_credential_params
     -script_id="1"
     -credential_name="NAMED_CREDENTIAL_X"
```

Example 2

The following command retrieves the Site Guard credential parameters for all scripts with the script ID of 3.

```
emcli get_siteguard_script_credential_params
     -script_id=3"
```

get_siteguard_script_hosts

Lists the host or hosts associated with any script where the script is designated to run.

Format

```
emcli get_siteguard_script_hosts
     [-script_id=<script_id>]
```

```
[ ]  indicates that the parameter is optional
```

Options

* **script_id**

  ID associated with the script.

Output Columns

Host Name

Examples

```
emcli get_siteguard_script_hosts
     -script_id="10"
```

See Also

add_siteguard_script_hosts
create_siteguard_script

get_siteguard_scripts

Obtains the Site Guard scripts associated with the specified system.

Format

```
emcli get_siteguard_scripts
      -system_name=<system_name>
      -operation=<operation_name>
      [-script_type=<type_of_script>]
      [-role=<role_of_system>]
```

Parameters

- **system_name**

  Name of the system.

- **operation**

  Name of the operation, such as switchover, failover, start, or stop.

- **script_type**

  Type of the script. For example: mount, unmount, pre-script, post-script, failover, or switchover.

- **role**

  Filters the scripts based on the role associated with the system. For example: Primary or Standby.

Output Columns

Script, ID, Type, Operation, Path, Role

Examples

Example 1

```
emcli get_siteguard_scripts
      -system_name="BISystem1"
      -operation="Switchover"
      -script_type="Pre-Script"
```

Example 2

```
emcli get_siteguard_scripts
      -system_name="austin-system"
      -operation="Switchover"
      -script_type="Pre-Script"
      -role="Primary"
```

See Also

create_siteguard_script
delete_siteguard_script

get_supported_platforms

Lists the platforms for which the Management Agent software is available on the OMS host.

Format

```
emcli get_supported_platforms
```

Output

The output of the command appears like This example:

```
-------------------------
Platform Name : Linux x86
-------------------------
```

get_supported_privileges

Gets the list of available privileges in Enterprise Manager based on the type specified.

Format

```
emcli get_supported_privileges
       -type="ResourceType"
       [-noheader]
       [-script]
       [-format=
               [name:<pretty|script|csv>];
               [column_separator:"column_sep_string"];
               [row_separator:"row_sep_string"];
       ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **type**

  Type of privileges to retrieve from Enterprise Manager. Possible values are:

  – ALL (default value)

  – SYSTEM

  – TARGET

  – JOB

- **noheader**

  Displays tabular information without column headers.

- **script**

  This is equivalent to `-format="name:script"`. Prints the output in a format that can be used in scripting.

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

- `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

- `format=column_separator:"column_sep_string"` column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

- `row_separator:"row_sep_string"` row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Privilege Name, Privilege Type, Resource Class, Resource GUID Column, Resource ID Columns

get_swlib_entity_details

Lists the details of an entity revision.

Format

```
emcli get_swlib_entity_details
    -entity_rev_id="entity_rev_id"
```

Parameters

- entity_rev_id

  Identifier of the entity revision.

get_system_members

Lists the members of the specified system.

Format

```
emcli get_system_members
    -name="name"
    [-type=<generic_system>]
    [-depth=#  (default 1)]
    [-noheader]
    [-script | -format=
        [name:<pretty|script|csv>];
        [column_separator:"column_sep_string"];
        [row_separator:"row_sep_string"];
    ]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Target name of the system.

- **type**

  System type: `generic_system`. Defaults to `generic_system`.

- **depth**

Lists target members in sub-systems to the specified depth. When the depth is set to 0, no system target members are listed, and only the system's existence is verified. When the depth is set to -1, all system and sub-system target members are listed.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This is equivalent to `-format="name:script"`. In interactive and script mode, the value must be True or False.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Source Target Name, Member Target Name, Member Target Type, Level

Examples

Example 1

This example lists the databases in system `db2_system`.

```
emcli get_system_members -name=db2_system
```

Example 2

This example verifies that system `my_system:generic_system` exists.

```
emcli get_system_members -name=my_system -depth=0
```

get_target_properties

Lists all the property names for the target type provided.

Format

```
emcli get_target_properties
        -target_type="target_type"
```

Options

- **target_type**

Target type for which you want to list user-defined property names.

Examples

```
emcli get_target_properties -target_type="host"

Comment
Contact
Deployment Type
Line of Business
Location
Target properties fetched successfully
```

get_target_types

Obtain target types and their details for the input plug-in.

Format

```
emcli get_target_types
    -plugin="Plug-in Id for which the targets types needs to be retrieved"
```

Output columns:Display Target Type, Target Type Is Composite (Y/N)

Options

- plugin

  Plug-in ID for which the target types needs to be retrieved.

Example

The following example shows all target types for the database plug-in:f

```
emcli get_target_types
    -plugin=oracle.sysman.db
```

get_targets

Gets status and alert information for targets.

Standard Mode

```
emcli get_targets
        [-targets="[name1:]type1;[name2:]type2;..."]
        [-alerts]
        [-noheader]
        [-script | -format=
                        [name:<pretty|script|csv>];
                        [column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];
        ]
        [-limit_rows="maximum_targets_to_be_retrieved"]
        [-config_search="configuration_search_UI_name"]
        [-unmanaged]
        [-properties]
        [-separator_properties="properties_sep_string"]
        [-subseparator_properties="properties_subsep_string"]

[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
get_targets
        [(targets="[name1:]type1;[name2:]type2;..."]
        [,alerts=True|False]
        [,noheader=True|False]
        [,script=True|False | ,format=
                    [name:<pretty|script|csv>];
                    [column_separator:"column_sep_string"];
                    [row_separator:"row_sep_string"];
        ]
        [,-limit_rows="maximum_targets_to_be_retrieved"]
        [,-config_search="configuration_search_UI_name"]
        [,-unmanaged]
        [,-properties]
        [,-separator_properties="properties_sep_string"]
        [,-subseparator_properties="properties_subsep_string"])
```

```
[ ]  indicates that the parameter is optional
```

Options

- **targets=name:type**

  Name or type can be either a full value or a pattern match using `%`. Also, name is optional, so the type can be specified alone.

- **alerts**

  Shows the count of critical and warning alerts for each target. In scripting and interactive mode, the value needs to be set to either True or False.

- **noheader**

  Display tabular output without column headers. In scripting and interactive mode, the value needs to be set to either True or False.

- **script**

  This is equivalent to `-format="name:script"`. In scripting and interactive mode, the value needs to be set to either True or False.

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  – `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

- **limit_rows**

Maximum number of targets to be retrieved. This defaults to 2000 rows if not specified.

- **config_search**

  The search UI name should be the display name of the configuration search.

- **unmanaged**

  Gets unmanaged targets (no status or alert information).

- properties

  Maximum number of targets to be retrieved. This defaults to 2000 rows if not specified.

- **separator_properties**

  Displays unmanaged target properties with separator_properties.

- **subseparator_properties**

  Displays unmanaged target properties with subseparator_properties.

Output Columns

Status ID, Status, Target Type, Target Name, Critical, Warning

Examples

These examples show name and type information for unmanaged host targets.

Example 1- Command-Line

```
emcli get_targets
      -targets="host"
      -unmanaged
```

Example 2 - Scripting and Interactive

```
get_targets    (targets="host"    ,unmanaged)
```

These examples show name, type, and properties for unmanaged host targets with the specified separators. By default, the separator_properties is ";" and the subseparator_properties is ":".

Example 3 - Command-Line

```
emcli get_targets
      -unmanaged -properties
      -separator_properties=,
      -subseparator_properties==
```

Example 4 - Scripting and Interactive

```
get_targets
    (unmanaged -properties
    ,separator_properties=,
    ,subseparator_properties==)
```

get_test_thresholds

Shows test thresholds.

Format

```
 emcli get_test_thresholds        -name=<target_name>        -type=<target_type>       -
testname=<test_name>       -testtype=<test_type>       [-script|-format=
[name:"pretty|script|csv"];
[column_separator:"sep_string"];                    [row_separator:"row_sep_string"]
      ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Target name.

- **type**

  Target type.

- **testname**

  Test name.

- **testtype**

  Test type.

- **script**

  This is equivalent to -format="name:script".

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  – `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli get_test_thresholds -name="Service Name"
              -type="generic_service"
              -testname="Test Name"
              -testtype="HTTP"
```

get_threshold

Obtains threshold information for a given target and metric.

Format

```
emcli get_threshold        -target_name="tname"        -target_type="ttype"        [-
metric="metric_group"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **target_name**

  Name of the target associated with the threshold.

- **target_type**

  Type of target associated with the threshold.

- **metric**

  Metric group associated with the threshold. The default without this option is to
  show the threshold of all metrics.

Examples

Example 1

This example gets the threshold data for the Load category on the host
myhost.example.com.

```
emcli get_threshold
         -target_name="myhost.example.com"
         -target_type="host"
         -metric="Load"
```

Example 2

This example gets the DiskActivitybusy threshold for the Disk Activity on the host
myhost.oracle.com.

```
emcli get_threshold
         -target_name="myhost.oracle.com"
         -target_type="host"
         -metric="DiskActivity"
```

get_unsync_alerts

Gets a list of alerts that are out-of-sync between the Management Agent and the
repository for the specified target. You would typically use this command when you
think that the Management Agent has not uploaded the latest alert to the repository.
Under these circumstances, the repository would be out-of-sync with the Management
Agent state.

Format

```
 emcli get_unsync_alerts        -target_type="type"
      -target_name="name"
```

Options

- **target_type**

Internal target type identifier, such as host, oracle_database, emrep, and so forth.

- **target_name**

  Name of the target.

Output Column

Status

Examples

This example shows the out-of-sync alert states for the host target type and abc.example.com target name:

```
emcli get_unsync_alerts -target_type=host -target_name=abc.example.com
```

get_unused_metric_extensions

Gets a list of metric extensions deployed to Agents, but not attached to any targets.

Format

```
 emcli get_unused_metric_extensions
```

Options

None.

get_update_status

Gets the latest status of an update.

Format

```
emcli get_update_status
        -id="internal id"
```

Options

- **id**

  Internal identification for the update.

Examples

This example displays the latest update status.

```
emcli get_update_status
        -id="914E3E0F9DB98DECE040E80A2C5233EB"
```

get_updatable_agents

Displays the Management Agents that can be updated using a particular Management Agent gold image version or a Management Agent gold image.

Format

```
emcli get_updatable_agents
        -version_name  | -image_name
```

```
              [-agents="Full Agent Name"]
              [-versions="List of Versions"]
              [-groups="List of group names"]
              [-output_file="Location of the output file"]
```

`[ ]  indicates that the parameter is optional.`

Options

- version_name

  Specifies the version name of the Management Agent gold image.

- image_name

  Specifies the gold image name to which the created Management Agent gold image must be added.

- versions

  Displays the Management Agents that can be updated, and are of the specified versions.

- agents

  Displays the Management Agents that can be updated, and whose name matches the specified name pattern.

- groups

  Displays the Management Agents that can be updated, and are a part of those groups whose name matches the specified name pattern.

- output_file

  Adds the displayed list of Management Agents that can be updated to an output file.

  Note: It is mandatory to specify the -version_name parameter or the -image_name parameter. If you specify both, a union of the outputs (when each of these parameters is specified individually) is displayed.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

The following example lists the Management Agents that can updated using the Management Agent gold image 'OPC_AGT_ADC_POD_JUNE', and adds the list to the output file '/scratch/agents_file.txt':

```
emcli get_updatable_agents
          -image_name="OPC_AGT_ADC_POD_JUNE" -output_file="/scratch/
agents_file.txt"
```

Example 2

The following example lists the Management Agents that are of version 12.1.0.1.0 or 12.1.0.2.0, and can be updated using the Management Agent gold image version 'OPC_AGT_ADC_POD_JUNE':

```
emcli get_updatable_agents
            -version_name="OPC_AGT_ADC_POD_JUNE" -versions="12.1.0.1.0,12.1.0.2.0"
```

get_upgradable_agents

Shows upgradable Agents. If you do not specify any options, the command shows all upgradable Agents. If you specify more than one option, the command shows the union of upgradable Agents belonging to each option specified.

Format

```
emcli get_upgradable_agents
      [-agents="full_agent_name"]
      [-platforms="list_of_platforms"]
      [-versions="list_of_versions"]
      [-groups="list_of_group_names"]
      [-output_file="output_file_location"]

[ ]  indicates that the parameter is optional
```

Options

*   **agents**

    Lists upgradable Agents matching Agent names or an Agent names pattern.

*   **platforms**

    Lists upgradable Agents on the specified platforms.

*   **versions**

    Lists upgradable Agents with the specified version.

*   **groups**

    Lists upgradable Agents belonging to the specified groups.

*   **output_file**

    Lists upgradable Agents and adds them to the specified file.

Examples

Example 1

This example lists upgradable Agents belonging to groups GROUP1 and GRP2.

```
emcli get_upgradable_agents -groups="GROUP1,GRP2"
```

Example 2

This example lists upgradable Agents and adds them to the file /scratch/agents_file.txt.

```
emcli get_upgradable_agents -output_file="/scratch/agents_file.txt"
```

grant_bipublisher_roles

Grants roles for accessing the BI Publisher catalog.

Format

```
emcli grant_bipublisher_roles
    (-roles="role1[;role2;...role_n]"
    [-users="user"]
    [-external_role="grantee_group"])

[ ]  indicates that the parameter is optional
```

> **Note:**
>
> The values of the roles and users parameters are case sensitive.

Options

- **roles**

  Grants one or more roles to BI Publisher. Specify one or more roles separated by a semicolon. The value of the roles parameters is case sensitive.

- **users**

  Users to receive the granted role. The value of the users parameters is case sensitive.

- **external_role**

  Group to assign the role.

Examples

Example 1

This example grants one role to a group.

```
emcli grant_bipublisher_roles -roles="EMBIPViewer" -external_role="TESTGROUPNAME"
```

Example 2

This example grants more than one role to a group.

```
emcli grant_bipublisher_roles -roles="EMBIPViewer;EMBIPAuthor"
    -external_role="TESTGROUPNAME"
```

grant_license_no_validation

Grants licenses on a set of user-specified packs, or all packs to a set of user-specified targets, or all targets belonging to the input licensable target type.

For 11g database targets, you cannot enable or disable the Database Diagnostic and Tuning Packs through the user interface. You need to set the control_management_pack_access initialization parameter to manage your licenses. For information about this option, see the Enterprise Database Management chapter of *Oracle Enterprise Manager Licensing Information*.

> **Tip:**
>
> You can use this verb to grant licenses for standalone target types, such as hosts and databases, but you cannot use this verb to grant licenses for the parent Application Server (oracle_ias) target type, which has dependent target types of OC4J, Jserv, Web Cache, and so forth. To do this, use the grant_license_with_validation verb instead.
>
> For example, for pack ias_config and an Application Server target of AS1 with an associated dependent target of OC4J1, this verb grants a license to AS1, but this does not propagate to OC4J1.

Format

```
emcli grant_license_no_validation        -type="target_type"       [-
targets="tname1;tname2;..."]       [-packs="pack1;pack2;..."]       [-file="file_name"]
      [-displayAllMessages]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **type**

    Target type as it exists in the database. Names cannot contain colons (:), semi-colons (;), or any leading or trailing blanks. You can specify only one target type at a time; for example, -type="oracle_database".

*   **targets**

    Targets should be specified in the following sequence:

    ```
    TargetName1;TargetName2;
    ```

    For example:

    ```
    -targets="database1;database2;database3;"
    ```

    The semi-colon (;) is the target separator.

    See the "Examples" section below for information about providing arguments for the targets .

*   **packs**

    License packs should be specified in the following sequence:

    ```
    pack1;pack2;
    ```

    For example:

    ```
    -packs="db_diag;db_config;"
    ```

    The semi-colon (;) is the pack separator.

    See the "Examples" section below for information about providing arguments for the packs .

*   **file**

    Specify the file name, including the complete path. For example:

```
-file="/usr/admin1/db_license.txt"
```

The file should contain the list of targets and packs according to the following cases:

– If you only need to provide a list of targets, use the following format:

```
targets=database1;database2;database3;
```

– If you only need to provide a list of packs, use the following format:

```
packs=db_diag;db_config;
```

– If you need to provide a list of both targets and packs, use the following format:

```
targets=database1;database2;database3;packs=db_diag;db_config;
```

• **displayAllMessages**

Displays all messages. Only error messages are displayed by default. "=value" is not allowed on the command line.

Examples

Example 1 and Example 2 below grant licenses to specific packs for specific targets. In order to know which target types and pack names you can pass as arguments, you can use the view named mgmt_license_view to see a list of licensable targets, their target types, and the list of packs licensed on them.

To obtain this information, do the following:

1. Access SQL*Plus with your username and password, using sysman or other user that has access to sysman.mgmt_license_view.

2. Select a distinct pack name from sysman.mgmt_license_view, where:

```
target_type=<oracle_database>
```

This example shows pack names for an Oracle database you specify as the target type.

```
PACK_NAME
-------------------
db_config
provisioning
db_sadm
db_tuning
db_diag
provisioning_db
db_chgmgt

7 rows selected.
```

Based on this information, to grant a license to the database1 target for the db_chgmgt pack, you would enter the following command:

```
emcli grant_license_no_validation -type="oracle_database" -targets="database1" -
packs="db_chgmgt"
```

The only limitation of mgmt_license_view is that it only lists the packs for a target type where the pack is granted to at least one target of that type. That is, if the pack is not granted to any target of that type, mgmt_license_view cannot provide any information.

Example 1

This example grants the license to the db_diag and db_config packs to database1, database2, and database3 targets (oracle_database target type):

```
emcli grant_license_no_validation -type="oracle_database"          -
targets="database1;database2;database3;" -packs="db_diag;db_config;"
```

Example 2

This example grants the license to the db_diag and db_config packs to all database targets in the setup:

```
emcli grant_license_no_validation -type="oracle_database"
        -packs="db_diag;db_config;"
```

grant_license_with_validation

Grants licenses on a set of user-specified packs, or all packs to a set of user-specified targets, or all targets belonging to the input licensable target type as per business rules.

For 11g database targets, you cannot enable or disable the Database Diagnostic and Tuning Packs through the user interface. You need to set the control_management_pack_access initialization parameter to manage your licenses. For information about this option, see the Enterprise Database Management chapter of *Oracle Enterprise Manager Licensing Information*.

> 💡 **Tip:**
>
> You can use this verb to grant licenses for standalone target types, such as hosts and databases, and you also use this verb to grant licenses for the parent Application Server (oracle_ias) target type, which has dependent target types of OC4J, Jserv, Web Cache, and so forth.
>
> For example, for pack ias_config and an Application Server target of AS1 with an associated dependent target of OC4J1, this verb grants a license to AS1 and also propagates to OC4J1 (and all other dependent targets associated with AS1).
>
> To grant licenses for only standalone target types, use the grant_license_no_validation verb.

Format

```
emcli grant_license_with_validation     -type="target_type"     [-
targets="tname1;tname2;..."]     [-packs="pack1;pack2;..."]     [-
file="file_name"]     [-displayAllMessages]

[ ]  indicates that the parameter is optional
```

Options

*   **type**

    Target type as it exists in the database. Names cannot contain colons (:), semi-colons (;), or any leading or trailing blanks. You can specify only one target type at a time; for example, -type="oracle_database".

- **targets**

  Targets should be specified in the following sequence:

  ```
  TargetName1;TargetName2;
  ```

  For example:

  ```
  -targets="database1;database2;database3;"
  ```

  The semi-colon (;) is the target separator.

  See the "Examples" section below for information about providing arguments for the targets .

- **packs**

  License packs should be specified in the following sequence:

  ```
  pack1;pack2;
  ```

  For example:

  ```
  -packs="db_diag;db_config;"
  ```

  The semi-colon (;) is the pack separator.

  See the "Examples" section below for information about providing arguments for the packs .

- **file**

  Specify the file name, including the complete path. For example:

  ```
  -file="/usr/admin1/db_license.txt"
  ```

  The file should contain the list of targets and packs according to the following cases:

  – If you only need to provide a list of targets, use the following format:

    ```
    targets=database1;database2;database3;
    ```

  – If you only need to provide a list of packs, use the following format:

    ```
    packs=db_diag;db_config;
    ```

  – If you need to provide a list of both targets and packs, use the following format:

    ```
    targets=database1;database2;database3; packs=db_diag;db_config;
    ```

- **displayAllMessages**

  Displays all messages. Only error messages are displayed by default. "=value" is not allowed on the cmd line.

Examples

Example 1 and Example 2 below grant licenses to specific packs for specific targets. In order to know which target types and pack names you can pass as arguments, you can use the view named mgmt_license_view to see a list of licensable targets, their target types, and the list of packs licensed on them.

To obtain this information, do the following:

1. Access SQL*Plus with your username and password, using sysman or other user that has access to sysman.mgmt_license_view.

2. Select a distinct pack name from sysman.mgmt_license_view, where:

```
target_type=<oracle_database>
```

This example shows pack names for an Oracle database you specify as the target type.

```
PACK_NAME
-------------------
db_config
provisioning
db_sadm
db_tuning
db_diag
provisioning_db
db_chgmgt

7 rows selected.
```

Based on this information, to grant a license to the database1 target for the db_chgmgt pack, you would enter the following command:

```
emcli grant_license_with_validation -type="oracle_database" -targets="database1" -
packs="db_chgmgt"
```

The only limitation of mgmt_license_view is that it only lists the packs for a target type where the pack is granted to at least one target of that type. That is, if the pack is not granted to any target of that type, mgmt_license_view cannot provide any information.

Example 1

This example grants a license to the db_diag and db_config packs to database1, database2, and database3 targets (oracle_database target type):

```
emcli grant_license_with_validation -type="oracle_database"            -
targets="database1;database2;database3;" -packs="db_diag;db_config;"
```

Example 2

This example grants a license to the db_diag and db_config packs to all database targets in the setup:

```
emcli grant_license_with_validation -type="oracle_database"
        -packs="db_diag;db_config;"
```

grant_privs

Grants the privileges to the existing Enterprise Manager user or Enterprise Manager Role.

> **Note:**
>
> To replace an existing Enterprise Manager administrator role, use the modify_role verb.

Format

```
emcli grant_privs
      -name="username|rolename"
      -privilege="name[;secure_resource_details]"
      [-grant_all_targets_on_host="yes|no"]
      [-separator=privilege="sep_string"]
      [-subseparator=privilege="subsep_string"]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  User name or role name to which privileges will be assigned.

- **privilege**

  Privilege to be granted to the Enterprise Manager user or role. You can specify this option more than once.

  Specify secure_resource_details as:

  ```
  resource_guid|[resource_column_name1=resource_column_value1
  [:resource_column_name2=resource_column_value2]..]"
  ```

  Optionally, you can drop resource column names from this option if you provide resource information in the order described by emcli get_supported_privileges. See the "See Also" section below for more information.

- **grant_all_targets_on_host**

  Indicates if the privilege needs to be granted on all targets of the host specified as part of the privilege parameter. The default value is no.

- **separator=privilege**

  Specify a string delimiter to use between name-value pairs for the value of the -privilege option. The default separator delimiter is a semi-colon (;).

- **subseparator=privilege**

  Specify a string delimiter to use between the name and value in each name-value pair for the value of the -privilege option. The default subseparator delimiter is a colon (:).

Examples

Example 1

This example grants these privileges to user1:

- Privilege to use any beacon

- Full control of the jobs with ID 923470234ABCDFE23018494753091111

- Full control on the target host1.example.com:host

- Full control on the credential cred1:user2

- View Privilege on target with ID 123451234ABCDFE23018494753092222

```
emcli grant_privs
      -name="user1"
```

```
        -privilege="USE_ANY_BEACON"
        -privilege="FULL_JOB;923470234ABCDFE23018494753091111"
        -privilege="FULL_TARGET;TARGET_NAME=host1.example.com:TARGET_TYPE=host"
        -privilege="FULL_CREDENTIAL;CRED_NAME=cred1:CRED_OWNER=user2"
        -privilege="FULL_CREDENTIAL;CRED_GUID=123451234ABCDFE23018494753092222"
```

Example 2

This example grants target privileges to EM Role : Role1:

```
emcli grant_privs
      -name="Role1"
      -privilege="FULL_TARGET;TARGET_NAME=host1.example.com:TARGET_TYPE=host"
```

See Also

To see the complete list of privileges and resource column names, execute the following command:

```
emcli get_supported_privileges
```

To see the list of SYSTEM privileges, which do require resource information:

```
emcli get_supported_privileges -type=SYSTEM
```

To see the list of TARGET privileges:

```
emcli get_supported_privileges -type=TARGET
```

To see the list of JOB privileges:

```
emcli get_supported_privileges -type=JOB
```

grant_roles

Grants roles to an existing Enterprise Manager user or Enterprise Manager role.

Format

```
emcli grant_roles
      -name="username|rolename"
      [-roles="role1;role2;..."]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  User name or role name to which roles will be assigned.

- **roles**

  Roles that will be granted to an Enterprise Manager user or role. You can specify this option more than once.

Examples

```
emcli grant_roles
      -name="user1"
      -roles="SUPER_USER"
```

```
emcli grant_roles
      -name="Role1"
      -roles="BLACKOUT_ADMIN;MAINTAIN_TARGET"
```

help

Shows a summary of all verbs or command-line help for individual EM CLI verbs.

> **✎ Note:**
>
> EM CLI must be set up and configured before command line help is available for all verbs.

Format

```
emcli help [verbname]
```

```
[ ]  indicates that the parameter is optional
```

Options

None.

Examples

Example 1

This example provides an overview for all available verbs:

```
emcli help
```

Example 2

This example provides the description, syntax, and usage examples for the add_target verb:

```
emcli help add_target
```

ignore_instance

Ignores a failed step. An instance cannot be ignored when it completes, completes with an error, is suspended, or is stopped.

Format

```
emcli ignore_instance
      -instance=<instance_guid>
      [exec=<execution_guid>]
      [-name=<execution_name>]
      [-owner=<execution_owner>]
      [-stateguid=<state_guid>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **instance**

  Instance GUID.

- **exec**

  Execution GUID.

- **name**

  Execution name.

- **owner**

  Execution owner.

- **stateguid**

  Comma-separated list of state GUIDs.

Example

```
emcli ignore_instance -instance=16B15CB29C3F9E6CE040578C96093F61 -
stateguid=51F762417C4943DEE040578C4E087168
```

import_adm

Imports an Application Data Model from the specified XML file.

Format

```
emcli import_adm
    -file=<file_name>
    -adm_name=<application_data_model_name>
    -target_name=<target_name>
    -target_type=<target type>
    [-desc=<description>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **file**

  File name with the absolute path of the XML file.

- **adm_name**

  Model name with which the Application Data Model will be imported.

- **target_name**

  Target for which the Application Data Model will be created.

- **target_type**

  Target type of the target for which the Application Data Model will be created.

- **desc**

  Application Data Model description.

Output

Success/error messages.

Examples

This example imports the Application Data Model from the sample_adm_import.xml file as Sample_ADM.

```
emcli import_adm
      -file=/home/user/sample_adm_import.xml
      -adm_name=Sample_ADM
      -target_name=test_database
      -target_type=oracle_pdb
      -desc="Application Data Model for EBS"
```

import_admin_group

Import Administration group hierarchy.

Format - Standard Mode

```
emcli import_admin_group
      -property_file="null"
```

Format - Interactive or Script Mode

```
import_admin_group(
      property_file="null"
      )
```

Options

• No Help available.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example creates an admin group with one level - Lifecycle status (with all 5 values).

```
ade:[ system_23 ] [system@sys01apq emgc]$ emcli export_admin_group
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<AdminGroup>
    <groupList>
        <parent>ADMGRP0</parent>
        <child>Deve-Grp</child>
        <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:Development</propertyValuePair>
    </groupList>
    <groupList>
        <parent>ADMGRP0</parent>
        <child>MC-Grp</child>
        <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:MissionCritical</propertyValuePair>
    </groupList>
```

```
        <groupList>
            <parent>ADMGRP0</parent>
            <child>Prod-Grp</child>
            <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:Production</propertyValuePair>
        </groupList>
        <groupList>
            <parent>ADMGRP0</parent>
            <child>Stag-Grp</child>
            <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:Stage</propertyValuePair>
        </groupList>
        <groupList>
            <parent>ADMGRP0</parent>
            <child>Test-Grp</child>
            <level>1</level>
<propertyValuePair>orcl_gtp_lifecycle_status:Test</propertyValuePair>
        </groupList>
        <levelList>
            <levelNumber>1</levelNumber>
<levelProperty>orcl_gtp_lifecycle_status</levelProperty>
            <propertyValues>Development</propertyValues>
            <propertyValues>MissionCritical</propertyValues>
            <propertyValues>Production</propertyValues>
            <propertyValues>Stage</propertyValues>
            <propertyValues>Test</propertyValues>
        </levelList>
        <rootNode>ADMGRP0</rootNode>
</AdminGroup>
emcli awrwh_add_src_db -target_name=sample_database -target_type=oracle_database
```

import_appreplay_workload

Imports a workload metadata XML file and creates a new application replay workload object. A Workload metadata XML file, which is stored in the workload root directory, is automatically generated as part of the workload capture process. The XML file contains a pointer to the actual raw captured workload data files. If you are importing a workload captured by one Enterprise Manager system to another, make sure the workload storage location specified in the XML file is reachable and contains the workload data files.

Format

```
emcli  import_appreplay_workload
        -input_file=template:<input_filename>

[ ]  indicates that the parameter is optional
```

Options

- **input_file**

  Fully-qualified path to a workload metadata XML file. The workload XML file is automatically created during capture. However, you may need to make necessary changes to the XML file before you import. For example, you may want to change the workload name in the exported file and rename the XML file to match the workload name. You may also need to modify the storage locations to point to where the workload data files are located if you have moved the captured data files.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

import_charge_plans

Imports charge plan metadata from the specified file.

Format

```
emcli import_charge_plans
        [-charge_plan="plan_name"[-entity_type=entity_type_name]]
        [<-create|-create_revision|-validate|-describe>]
        [-start_date=ddmmyyyy]
        -file=file_name
```

```
[ ]  indicates that the parameter is optional
```

Options

- **charge_plan**

  Name of the charge plan to import. If this option is not specified, imports all charge plans within the file.

- **entity_type**

  Name of the Chargeback entity type whose charge rate metadata to import from the specified charge plan within the file. If this option not specified, import all entity type charge rates from the charge plan.

- **create**

  Import charge plan metadata to create a charge plan.

- **create_revision**

  Import charge plan metadata to create a charge plan revision.

- **validate**

  Validate the charge plan metadata file. This is the default action if no import operation is specified.

- **describe**

  Describe the charge plan metadata in the specified file.

- **start_date**

  Start date in ddmmyyyy format of the report cycle for the applicable charge plan import operation. If this option not specified, uses the start date of the current report cycle.

- **file**

  Absolute path of the XML file containing the charge plan metadata to import.

Examples

Example 1

This example describes charge plan metadata in /home/allplans.xml:

```
emcli import_charge_plans
        -file=/home/allplans.xml
        -describe
```

Example 2

This example imports Plan C charge plan metadata in /home/plans.xml to create a plan revision with an effective start date of 01092014:

```
emcli import_charge_plans
        -charge_plan="Plan C"
        -file=/home/plans.xml
        -create_revision
        -start_date=01092014
```

import_compliance_object

Imports a compliance object into the repository.

Format

```
import_compliance_object
      -files=file1;file2;...  [-overwrite] [-deep]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **files**

  Files to be imported.

- **overwrite**

- **deep**

Examples

```
emcli import_compliance_object
        -files=file1.xml;file2.xml -overwrite
```

import_config

Imports a saved configuration.

Format

```
emcli import_config
      -input_file="ImportConfig.zip"
      -name="Host Saved Configuration"
      [-description="Imported Configuration"]
      [-ignore="Yes"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- input_file

  Input zip file with an absolute path that contains the target's saved configuration.

- name

  Name of the saved configuration. The value should be unique and not null.

- description

  Description of the imported target's saved configuration. The default value is "Imported Configuration". This option is not mandatory.

- ignore

  Specifies whether or not to ignore the configuration extension version mismatch. The value is Yes or No. The default value is No. This option is not mandatory.

Example

The following command imports the saved configuration specified in the file "ImportConfig.zip", saves with the name "Host Saved Configuration" and description "Imported Configuration", and ignores any configuration extension version mismatch:

```
emcli import_config
    -input_file="ImportConfig.zip"
    -name="Host Saved Configuration"
    -description="Imported Configuration"
    -ignore="Yes"
```

The following command imports a previously exported configuration file after editing the init parameters as needed:

```
emcli import_saved_config
-name="Edited Baseline Name"
-input_file="../Baseline1.zip" -X
```

> **Note:**
>
> The -X parameter ignores all the additional file attributes that are generated during the export. Failure to use -Xwill generate an error while uploading an edited configuration file.

import_config_compare_template

Imports the comparison template provided as argument "filename".

Format

```
emcli import_config_compare_template
    -filename="<file_name>"
```

Options

- filename

  Input XML file with absolute path which represents a comparison template.

Example

The following example imports the comparison template represented by the file "/tmp/sample_comparison_template".

```
emcli import_config_compare_template
    -filename="/tmp/sample_comparison_template.xml"
```

import_config_search

Imports a configuration search XML file to the repository.

Format

```
emcli import_config_search
    -file="<XML file name>"
```

Options

• file

The name of the xml file. The file name must include the absolute path.

Example

The following example imports the configuration to the importfile.xml file:

```
emcli import_config_search      -file="/tmp/importfile.xml"
```

import_custom_charge_items

Imports user-defined charge item metadata from the specified file.

Format

```
emcli import_custom_charge_items
    -file=file_name
    [-validate]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **file**

Absolute path of the XML file from which to import user-defined charge item metadata.

• **validate**

Validates the XML file.

Examples

Example 1

This example imports user-defined charge item metadata from /home/host.xml:

```
emcli import_custom_charge_items
    -file=/home/host.xml
```

Example 2

This example validates user-defined charge item metadata in /home/host.xml:

```
emcli import_custom_charge_items
    -file=-file=/home/host.xml
    -validate
```

import_custom_plugin_update

Imports a custom plug-in update that was created using the Extensibility Development Kit. The imported plug-in update is used for all subsequent plug-in deployments on the Management Agents.

Format

```
emcli import_custom_plugin_update
     -archive="<path_to_plugin_update_archive>"
      [-overwrite]
```

Options

*   **archive**

    Absolute path to the custom update archive file.

*   **overwrite**

    Overwrites an existing custom plug-in update, if a custom plug-in update already exists for that plug-in. If not provided, the custom plug-in update is not imported for that plug-in. Applies only to subsequent plug-in deployments. Does not automatically redeploy on the Management Agents where the already-existing plug-in was previously deployed. To redeploy on such Management Agents, run the emcli redeploy_plugin_on_agent verb.

Examples

Example 1

The following example imports the `12.1.0.4.0_oracle.sysman.db.008.zip` archive file from the `/u01/oracle/plugin_updates/` location, assuming no custom plug-in update already exists for that plug-in.

```
emcli verb_name
     -archive="/u01/oracle/plugin_updates/12.1.0.4.0_oracle.sysman.db.008.zip"
```

Example 2

The following example imports the `12.1.0.4.0_oracle.sysman.db.008.zip` archive file from the `/u01/oracle/plugin_updates/` location, and overwrites any existing custom plug-in update that already exists for that plug-in.

```
emcli import_custom_plugin_update
     -archive="/u01/oracle/plugin_updates/12.1.0.4.0_oracle.sysman.db.008.zip"
     -overwrite
```

import_incident_rule_set

Imports a rule set from a list of enterprise rule set(s)

> **✏️ Note:**
>
> Oracle-supplied out-of-box rule sets cannot be imported.

Privilege Requirements

Any user who has been granted the *Create Rule Set Privilege* can import the enterprise rule set.

Format

```
emcli import_incident_rule_set
    -import_file=<XML file created by the export_incident_rule_set verb>
    [-alt_rule_set_name=<rule set name>]

[ ]  indicates that the parameter is optional.
```

Options

- import_file

  XML file name along with the file path for the exported rule set earlier.

- alt_rule_set_name (Optional)

  This option allows you to specify an alternate enterprise rule set name that will be used if there is an existing rule set with the same original name.

Example

The following command imports *TEST_RULESET.xml* and creates a rule set named *COPY_OF_TEST_RULESET*.

```
emcli import_incident_rule_set -import_file="/tmp/TEST_RULESET.xml"  -
alt_rule_set_name=COPY_OF_TEST_RULESET
```

import_jobs

Imports all job definitions into Enterprise Manager, including Corrective Actions from a zip file. Library jobs are created. The EM CLI logged-in user is set as the library job owner.

Format

```
emcli import_jobs
   -import_file=<zip_file_name>"
   [-name="job name1;job_name2;..."]
   [-type="job type1;job_type2;..."]
   [-targets="tname1:ttype1;tname2:ttype2;..."]    [-owner="owner1;owner2;..."]
   [-preview]   [-force]   [-stoponerror]

[ ]  indicates that the parameter is optional
```

Options

- **import_file**

  Zip file name that contains job definitions.

- **name**

Job name to be used for filtering. Semicolon-separated job names can be provided. Filtering by using a wildcard character is not supported.

- **type**

  Job type to be used for filtering. Semicolon-separated job types can be provided. Filtering by using a wildcard character is not supported.

- **targets**

  Target name and target type to be used for filtering. Semicolon-separated target names and types can be provided. Filtering by using a wildcard character is not supported.

- **owner**

  Job owner to be used for filtering. Semicolon-separated owners can be provided. Filtering by using a wildcard character is not supported.

- **preview**

  Prints the job definitions in the zip file. Filter values provided are used to show only matching job definitions. Jobs are not created in Enterprise Manager.

- **force**

  Updates the job record if it already exists. Otherwise, the job record is created. When this option is not specified, the default behavior of the system is to always create jobs from the import file.

- **stoponerror**

  Stops the import operation is after the first failure of the job import and rolls back the transaction. All jobs created by using this EM CLI session are deleted.

Output Columns

Success/Error messages.

Examples

Example 1

This example imports all job definitions into Enterprise Manager. If the job already exists, the details are edited. Otherwise, a new job is created.

```
emcli import_jobs  -import_file=job data.zip -force
```

Example 2

This example imports all job definitions into Enterprise Manager, and on the first failure, rolls back the jobs created in this session. The remaining jobs from the import file are not processed. Otherwise, a new job is created.

```
emcli import_jobs  -import_file=job data.zip -stoponerror
```

import_masking_definition

Imports a masking definition from the specified XML file.

Format

```
emcli import_masking_definition
      -file=/tmp/file_name.xml
```

Options

- **file**

  Path of the file containing the masking definition in XML format.

Output

Success or error messages.

Examples

This example imports the masking definition from the hr_mask.xml file.

```
emcli import_masking_definition
        -file=/tmp/hr_mask.xml
```

import_metric_extension

Imports a metric extension archive file.

Format

```
emcli import_metric_extension
       -file_name=<metric_extension_archive>
       -rename_as=<metric_extension_to_import_as>
```

Options

- **file_name**

  Name of the metric extension archive file to be imported.

- **rename_as**

  Imports the metric extension using the specified name, replacing the name given in the archive.

Examples

This example imports the masking definition from the hr_mask.xml file.

```
emcli import_metric_extension
     -file_name=<file name>
     -rename_as=<metric extension name>
```

import_replays

Imports the replays contained in a directory. An XML input file is required that describes the path of the directory containing replay data files; the database target to be used to load replays; and, optionally, replay names to import only specific replays into the Enterprise Manager repository.

Sample XML File:

```
<?xml version="1.0" encoding="UTF-8"?>
    <cliImportData xmlns="http://xmlns.oracle.com/sysman/db/dbreplay">
      <targetName>database</targetName>
      <targetType>oracle_database</targetType>
      <dbHostName>host.example.com</dbHostName>
      <dbCredRef>
         <credName>testDB121</credName>
         <credOwner>sysman</credOwner>
      </dbCredRef>
      <dbHostCredRef>
         <credName>testDBHost121</credName>
         <credOwner>sysman</credOwner>
      </dbHostCredRef>
      <directoryPath>/storage/dbr/copyDir_task4Caps</directoryPath>
      <consolidatedDirectory>true</consolidatedDirectory>
      <replays>
         <taskName>myTask</taskName>
         <replay>
            <nameInEm>myNewConsTrial_1</nameInEm>
            <nameInTarget>myNewConsTrial_1</nameInTarget>
         </replay>
      </replays>
    </cliImportData>
```

Format - Standard Mode

```
emcli import_replays
       [-input_file="template:<input file path>"]
       [-all]
       [-format="[name:<pretty|script|csv>];
[column_separator:"column_sep_string"];[row_separator:"row_sep_string"]"]
```

```
[ ]  indicates that the parameter is optional.
```

Format - Interactive or Script Mode

```
import_replays(
       [input_file="template:<input file path>"]
       [,all=True/False]
       [,format="[name:<pretty|script|csv>];
[column_separator:"column_sep_string"];[row_separator:"row_sep_string"]"]
       )
```

```
[ ]  indicates that the parameter is optional.
```

Options

• input_file

   Fully qualified path to an XML file containing parameters for the verb.

• all

   Imports all replays into the Enterprise Manager repository. No names need to be specified in the XML input file unless there is a naming conflict with a preexisting Enterprise Manager replay.

• format

   Specifies how the output is formatted. The default value is "name:pretty". You can use the parameter in the following ways:

- – -format="name:pretty" Prints the output table in a readable format not intended to be parsed by scripts.

- – -format="name:script" Sets the default column separator to a tab and the default row separator to a newline in the output. You can override the column and row separator strings with your own values.

- – -format="name:script;column_separator:<column_sep_string>" Causes the verb output to be column-separated by <column_sep_string>. Rows are separated by the newline character.

- – -format="name:script;row_separator:<row_sep_string>" Causes the verb output to be row-separated by <row_sep_string>. Columns are separated by the tab character.

- – -format="name:script;column_separator:<column_sep_string>;row_separator:<row_sep_string>" Causes the verb output to be column-separated by <column_sep_string> and row-separated by <row_sep_string>.

- – -format="name:csv" Sets the default column separator to a comma and the default row separator to a newline in the output.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Output

Output Columns: EM Entity Name, Entity Name, Database Name, Start Time, Import Status

Examples

Example 1 - Standard Mode

The following example imports all replays from a directory into the Enterprise Manager repository.

```
emcli import_replays      -all      -input_file=template:/storage/xml/
importReplay_capture121_5Mins.xml
```

Example 2 - Interactive or Script Mode

The following example imports all replays from a directory into the Enterprise Manager repository.

```
import_replays(all=True,        input_file="template:/storage/xml/
importReplay_capture121_5Mins.xml"      )
```

Example 3 - Standard Mode

The following example imports replays from a directory into the Enterprise Manager repository. Only those replays specifically named in the XML input file are imported.

```
emcli import_replays        -input_file=template:/storage/xml/
importReplay_capture121_5Mins.xml
```

Example 4 - Interactive or Script Mode

The following example imports replays from a directory into the Enterprise Manager repository. Only those replays specifically named in the XML input file are imported.

![ORACLE]

```
import_replays(input_file="template:/storage/xml/
importReplay_capture121_5Mins.xml"         )
```

See Also

discover_workloads
import_workloads

import_report

Imports one or more Information Publisher report definitions from an XML file(s) using the title in the XML file and the currently logged-in CLI user as the owner of the report. If the report/owner already exists, the operation fails for this report with an accompanying error message. (You can override this with the -force option.) The report will be changed to a just-in-time report with the target type from the exported report.

You will need to edit schedules and access privileges using the Enterprise Manager user interface. The system enforces title/owner uniqueness, so an error occurs if a report with the same title and owner already exists.

Format

```
emcli import_report
      -files="file1;file2;..."
      [-force]
```

```
[ ]   indicates that the parameter is optional
```

Options

- **files**

  List of path/file name(s) of XML file(s) that contain valid report definition(s).

- **force**

  First delete the report (and all jobs and saved copies) if a report with the same title/owner exists.

Examples

```
emcli import_report      -files="$HOME/reports/maint_report1.xml;$HOME/reports/
file2.xml"
```

import_sla

Imports an SLA configuration XML file for a target. This verb provides the functionality of creating a new SLA, creating a new version, and creating a new copy.

> **✏ Note:**
>
> The XML file can only contain one SLA to be imported; that is, when export_sla has successfully exported a file when slaName and version are specified.

> **✏️ Note:**
>
> The target must have the metrics required by the SLA template's SLI. If the
> template's SLI calls for a metric not found in the target, the SLI cannot be created.

Format

```
emcli import_sla
      -targetName=<target name>
      -targetType=<target type>
      -input_file=slaTemplate:<input filename>
      [-slaName=<SLA name>]

[ ]  indicates that the parameter is optional
```

Options

- **targetName**

  Name of the target.

- **targetType**

  Type of target.

- **input_file**

  Name of the input file. There can only be one SLA root node in the XML document.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **slaName**

  Specifying this name overrides the name contained in the SLA template XML file. This
  effectively creates a new SLA version series starting with version 1.

Examples

This example creates an SLA named 'gold_sla' for the target my_service (generic_service).

```
emcli import_sla
      -targetName='my_service' -targetType='generic_service'
      -slaName='gold_sla' -input_file=slaTemplate:'service_sla.xml'
```

import_subset_definition

Imports a subset definition from the specified XML file.

Format

```
emcli import_subset_definition
      -adm_name=<Application_Data_Model_Name>
      -subset_name=<Subset_Definition_Name>
      -target_name=<Target_Database_Name>
      -target_type=<Target_Type>
      -file=<Import_File>
      [-db_pref_cred_name=<DBCredsNormal | DBCredsSYSDBA>]
      [-db_cred_name=<Database_Credential_Name>]
      [-description=<Description>]
      [-job_name=<Job_Name>]
```

```
    [-job_description=<Job_Description>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **adm_name**

  Application Data Model (ADM) name.

- **subset_name**

  Name of the imported subset definition.

- **target_name**

  Target database name.

- **target_type**

  Type of target. Possible values are 'oracle_database', 'rac_database' and 'oracle_pdb'.

- **file**

  Fully-qualified file name of the file in XML format.

- **db_cred_name**

  Name of existing credentials stored in the Enterprise Manager repository to connect selected target database.

  You must provide a value for either db_cred_name or db_pref_cred_name.

- **db_pref_cred_name**

  Name of preferred credentials stored in the Enterprise Manager repository. You must provide a value for either db_cred_name or db_pref_cred_name.

  Valid values for this option are:

  – DBCredsNormal: Default normal credential set for an oracle_database target.

  – DBCredsSYSDBA: SYSDBA credential set for an oracle_database target.

- **description**

  Description for the imported subset definition.

- **job_name**

  Job name for the import subset definition operation.

- **job_description**

  Job description.

Examples

Example 1

This example imports a subset definition from an XML file at path /scratch/samples/HR_subset.xml.

```
emcli import_subset_definition
      -adm_name=adm
      -file=/scratch/samples/HR_subset.xml
      -subset_name=HR_Subset
      -db_cred=cred
```

```
                     -target_name=sample_database
                     -target_type=oracle_database
```

Example 2

This example imports a subset definition from an XML file at path /scratch/samples/
HR_subset.xml using preferred normal database credentials.

```
emcli import_template -files="e1.xml;e2.xml;e3.xml"
```

import_subset_dump

Imports the dump file into the specified target database.

Format

```
emcli import_subset_dump
      -target_name=<Target Database>
      -target_type=<Target Database type>
      [-db_cred_name=<Database Credential Name>]
      [-db_pref_cred_name=<DBCredsNormal | DBCredsSYSDBA> ]
      [-host_cred_name=<Host Credential Name>]
      [-subset_directory=<Database Directory Object Name> ]
      [-custom_directory_path=<Custom Directory Path>]
      [-use_external_directory]
      [-external_directory=<External Directory Object Name>]
      [-export_file_name=<Exported Dump File Name>]
      [-max_imp_threads=< Maximum Number of Import Threads>]
      [-encrypted_dump_file]
      [-encryption_password=<Encryption Password>]
      [-import_type=<ALL | DATA_T_L| DATA_A_L> ]
      [-tablespace_map=<Tablespace Map>]
      [-schema_map=<Schema Map>]
      [-log_file_name=<Log file name>]
      [-job_name=<Job Name>]
      [-job_description=<Job Description>]
      [-oid_transform]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **target_name**

  Name of the existing target database.

- **target_type**

  Type of target. Possible values target type are 'oracle_database', 'rac_database', and
  'oracle_pdb'.

- **db_cred_name**

  Name of existing credentials stored in the Enterprise Manager repository to connect
  selected target database. You must provide a value for either db_pref_cred_name_or
  db_cred_name.

- db_pref_cred_name

  Name of preferred credentials stored in the Enterprise Manager repository.

  Valid values are:

- DBCredsNormal — Default normal credential set for an oracle_database target.

- DBCredsSYSDBA — SYSDBA credential set for an oracle_database target.

You must provide a value for either db_pref_cred_name_or db_cred_name.

- **host_cred_name**

  Name of existing host credentials stored in the Enterprise Manager repository to access the target host.

- **subset_directory**

  Database Directory where the dump file is stored. For example: DATA_PUMP_DIR

  You must provide a value for either subset_directory or custom_directory_path.

- **custom_directory_path**

  User-specified directory location on the target host where the dump file is present. For example: /scratch/user/subset_dir

  You must provide a value for either subset_directory or custom_directory_path.

- use_external_directory

  Flag to enable using an external directory (clustered/shared file system or ASM) for faster import processing. If you do not set this option, you must provide a value for external_directory.

- **external_directory**

  External directory location (clustered/shared file system or ASM) object for faster host access. For example: DATA_PUMP_DIR

- **export_file_name**

  Name of the dump file to import. If not specified, the default value is EXPDAT%U.DMP.

- max_imp_threads

  Maximum number of import threads. If not specified, the default value is 1.

- **encrypted_dump_file**

  Set this option if an encryption password was specified during the export operation. If you use this option, you must also provide a value for encryption_password.

- **encryption_password**

  Password to decrypt encrypted data during an import operation. The specified password should be same as that specified during the export operation. If the encrypted_dump_file option is set and a value for this option is not specified, you are prompted for the encryption password.For a secure operation, it is recommended that passwords not be stored in the scripts, but instead specified when prompted for them.

- import_type

  Drives an import operation. Valid values are:

  - ALL: Import both metadata and data.

  - DATA_T_L: Data within the preexisting table will be removed. Data in the import source will replace it.

– DATA_A_L: Data contained within the table to be imported will be appended to the end of the preexisting table.

The default value is ALL.

- **tablespace_map**

  This password is required to re-map data from one tablespace to another. For example:

  ```
  -tablespace_map="source_tbsp1:target_tbsp1;source_tbsp2:target_tbsp2"
  ```

- **schema_map**

  This password is required to re-map data from one schema to another.

  ```
  -schema_map="source_schema1:target_schema1;source_schema2:target_schema2"
  ```

- log_file_name

  If not specified, the default value is IMPORT.LOG.

- oid_transform

  By default, the exported OID is imported during table or type creation. Set this option to create a new OID. This is useful when some of the objects already exist in the database and a cloned copy is required. However, selecting this option will cause breakage in REF columns that point to the table.

- **job_name**

  Import subset dump operation job name.

- job_description

  Job description.

Output

Success or error message along with the job name if applicable.

Examples

Example 1

This example imports dump(E.dmp) located at the DATA_PUMP_DIR directory into the target sample_database.

```
emcli import_subset_dump -db_cred_name=db_cred -export_file_name=E.dmp -
host_cred_name=host_cred -subset_directory=DATA_PUMP_DIR -target_type=oracle_database -
target_name=sample_database -import_type=All
```

Example 2

This example imports dump(E.dmp) located at the DATA_PUMP_DIR directory into the target sample_database using preferred database and host credentials.

```
emcli import_subset_dump -export_file_name=E.dmp -db_pref_cred_name=DBCredsNormal -
subset_directory=DATA_PUMP_DIR -target_type=oracle_database -
target_name=sample_database -import_type=All
```

import_swlib_cache_files

Imports Software Library entity files from a compressed file to a cache node.

ORACLE®

Format

```
emcli import_swlib_cache_files
      -source_directory_path="source_directory_path"
      -zip_file_name="zip_file_name"
      -source_host_name="source_host_name"
      -cache_node_name="cache_node_name"
      -cache_node_name="cache_node_name"
      -source_host_tmp_directory="source_host_tmp_directory"
```

Parameters

* source_directory_path

  Location on the host where the compressed file for import is available.

* zip_file_name

  Name of the compressed file for import. Make sure the compressed file is readable by the credentials saved with the cache node.

* source_host_name

  Name of the host target on which the compressed file is created.

* source_host_tmp_directory

  Directory on the source host in which temporary files are created during import. Make sure this directory is writable by the credentials saved with the cache node.

Example

The following example imports the compressed file `/u01/import_loc/exportedfile.zip` from the host import.us.acme.com, to `cache node west_cache_node` on the same host. After import, the files are cached on this cache node and staging of these files to targets in the `west_cache_node` group can be faster.

```
emcli import_swlib_cache_files
      -source_directory_path="/u01/import_loc"
      -zip_file_name="exportedfile.zip"
      -source_host_name="import.us.acme.com"
      -cache_node_name="west_cache_node"
      -source_host_tmp_directory="/tmp"
```

import_template

Imports a monitoring template from an XML or zip file. The resulting definition is saved in the repository.

Format

```
emcli import_template
      -files="file1;file2;..."
```

Options

* **files**

  Path/file name of an XML file, which contains a valid template definition. You can specify multiple files with this option by separating each file with a semi-colon (;).

Examples

Example 1

This example imports a template from `template.xml`.

```
emcli import_template -files="template.xml"
```

Example 2

This example imports three templates — one from each of the files specified.

```
emcli import_template -files="e1.xml;e2.xml;e3.xml"
```

import_update

Imports a Self Update archive file into Enterprise Manager. Upon successful import, the update is displayed on the Self Update Home in downloaded status for further action.

Format

```
emcli import_update
        -file="file"
        -omslocal
emcli import_update
        -file="file"
        -host="hostname"
        [-credential_set_name="setname"] | -credential_name="name"
        -credential_owner="owner"
```

```
[ ]  indicates that the parameter is optional
```

Options

- **file**

  Complete path name of the update archive file.

- **omslocal**

  Flag specifying that the file is accessible from the OMS.

- **host**

  Target name for a host target where the file is available.

- **credential_set_name**

  Set name of the preferred credential stored in the repository for the host target. Can be one of the following: HostCredsNormal — Default unprivileged credential set HostCredsPriv — Privileged credential set

- **credential_name**

  Name of a named credential stored in the repository. You must specify this along with the credential_owner .

- **credential_owner**

  Owner of a named credential stored in the repository. You must specify this option along with the credential_name option.

Examples

Example 1

This example imports the file update1.zip. The file must be present on the OMS host. In a multiple OMS setup, any OMS can process the request, so the file should be accessible from the OMS processing the request. This usually means that the file must be kept on a shared location accessible from all OMSes.

```
emcli import_update
       -file="/u01/common/update1.zip"
       -omslocal
```

Example 2

This example imports the file update1.zip that is present on the host host1.example.com. The host must be a managed host target in Enterprise Manager, and the Management Agent on this host must be up and running. The preferred unprivileged credentials for host host1.example.com are used to retrieve the remote file.

```
emcli import_update
       -file="/u01/common/update1.zip"
       -host="host1.example.com"
       -credential_set_name="HostCredsNormal"
```

import_update_catalog

Imports a Self Update master catalog file when Enterprise Manager is configured in offline mode. All updates present in the catalog are processed, and the applicable updates are displayed on the Self Update Home for further action.

Format

```
emcli import_update_catalog
       -file="file"
       -omslocal
       -file="file"
       -host="hostname"
       [-credential_set_name="setname"] | -credential_name="name"
        -credential_owner="owner"
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **file**

    Complete path name of the self update catalog file.

*   **omslocal**

    Flag specifying that the file is accessible from the OMS.

*   **host**

    Target name for a host target where the file is available.

*   **credential_set_name**

Set name of the preferred credential stored in the repository for the host target. Can be one of the following: HostCredsNormal — Default unprivileged credential set HostCredsPriv — Privileged credential set

- **credential_name**

  Name of a named credential stored in the repository. You must specify this along with the credential_owner option.

- **credential_owner**

  Owner of a named credential stored in the repository. You must specify this option along with the credential_name option.

Examples

Example 1

This example imports the master catalog file p9984818_121000_Generic.zip that is present on the host host1.example.com. The host must be a managed host target in Enterprise Manager, and the Management Agent on this host must be up and running. The preferred unprivileged credentials for host host1.example.com are used to retrieve the remote file.

```
emcli import_update_catalog
       -file="/u01/common/p9984818_121000_Generic.zip"
       -host="host1.example.com"
       -credential_set_name="HostCredsNormal"
```

Example 2

This example imports the master catalog file p9984818_121000_Generic.zip that is present on the host host1.example.com. The host must be a managed host target in Enterprise Manager, and the Management Agent on this host must be up and running. The named credentials "host1_creds" owned by user "admin1" are used to retrieve the remote file.

```
emcli import_update_catalog
       -file="/u01/common/p9984818_121000_Generic.zip"
       -host="host1.example.com"
       -credential_name="host1_creds"
       -credential_owner="admin1"
```

import_workloads

Imports the captures and (optionally) the replays contained in a directory. An XML input file is required that describes the path of the directory containing capture (and replay) data files; the database target to be used to load captures (and replays); and, optionally, capture (and replay) names to import only specific captures (and replays) into the Enterprise Manager repository.

Sample XML File:

```
<?xml version="1.0" encoding="UTF-8"?>
    <cliImportData xmlns="http://xmlns.oracle.com/sysman/db/dbreplay">
      <targetName>database</targetName>
      <targetType>oracle_database</targetType>
      <dbHostName>host.example.com</dbHostName>
      <dbCredRef>
        <credName>testDB121</credName>
        <credOwner>sysman</credOwner>
      </dbCredRef>
      <dbHostCredRef>
```

```
                <credName>testDBHost121</credName>
                <credOwner>sysman</credOwner>
            </dbHostCredRef>
            <directoryPath>/storage/dbr/myConsTask</directoryPath>
            <consolidatedDirectory>true</consolidatedDirectory>
            <replays>
                <taskName>myTask</taskName>
                <replay>
                    <nameInEm>myNewConsTrial_1</nameInEm>
                    <nameInTarget>myNewConsTrial_1</nameInTarget>
                </replay>
            </replays>
        </cliImportData>

        <?xml version="1.0" encoding="UTF-8"?>
        <cliImportData xmlns="http://xmlns.oracle.com/sysman/db/dbreplay">
            <targetName>database</targetName>
            <targetType>oracle_database</targetType>
            <dbHostName>host.example.com</dbHostName>
            <dbCredRef>
                <credName>testDB121</credName>
                <credOwner>sysman</credOwner>
            </dbCredRef>
            <dbHostCredRef>
                <credName>testDBHost121</credName>
                <credOwner>sysman</credOwner>
            </dbHostCredRef>
            <directoryPath>/storage/dbr/copyDir_task4Caps</directoryPath>
            <consolidatedDirectory>true</consolidatedDirectory>
            <captures>
                <capture>
                    <nameInEm>capture01</nameInEm>
                    <nameInTarget>capture01</nameInTarget>
                </capture>
                <capture>
                    <nameInEm>capture10</nameInEm>
                    <nameInTarget>capture10</nameInTarget>
                </capture>
            </captures>
        </cliImportData>
```

Format - Standard Mode

```
emcli import_workloads
        [-input_file="template:<input file path>"]
        [-include_replays]
        [-all]
        [-format="[name:<pretty|script|csv>];
[column_separator:"column_sep_string"];[row_separator:"row_sep_string"]"]

[ ]   indicates that the parameter is optional.
```

Format - Interactive or Script Mode

```
import_workloads(
        [input_file="template:<input file path>"]
        [,include_replays=True/False]
        [,all=True/False]
        [,format="[name:<pretty|script|csv>];
[column_separator:"column_sep_string"];[row_separator:"row_sep_string"]"]
        )
```

`[ ]  indicates that the parameter is optional.`

Options

- input_file

    Fully qualified path to an XML file containing parameters for the verb.

- include_replays

    Imports replays into the Enterprise Manager repository along with the associated captures that are imported.

- all

    Imports all captures (and replays) into the Enterprise Manager repository. No names need to be specified in the XML input file unless there is a naming conflict with a preexisting Enterprise Manager capture (or replay).

- format

    Specifies how the output is formatted. The default value is "name:pretty". You can use the parameter in the following ways:

    – -format="name:pretty" Prints the output table in a readable format not intended to be parsed by scripts.

    – -format="name:script" Sets the default column separator to a tab and the default row separator to a newline in the output. You can override the column and row separator strings with your own values.

    – -format="name:script;column_separator:<column_sep_string>" Causes the verb output to be column-separated by <column_sep_string>. Rows are separated by the newline character.

    – -format="name:script;row_separator:<row_sep_string>" Causes the verb output to be row-separated by <row_sep_string>. Columns are separated by the tab character.

    – -format="name:script;column_separator:<column_sep_string>;row_separator:<row_sep_string>" Causes the verb output to be column-separated by <column_sep_string> and row-separated by <row_sep_string>.

    – -format="name:csv" Sets the default column separator to a comma and the default row separator to a newline in the output.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Output

Output Columns: EM Entity Name, Entity Name, Database Name, Start Time, Import Status

Examples

Example 1 - Standard Mode

The following example imports all captures from a directory into the Enterprise Manager repository.

```
emcli import_workloads      -all       -input_file=template:/storage/xml/
importReplay_capture121_5Mins.xml
```

Example 2 - Interactive or Script Mode

The following example imports all captures from a directory into the Enterprise
Manager repository.

```
import_workloads(all=True,        input_file="template:/storage/xml/
importReplay_capture121_5Mins.xml"      )
```

Example 3 - Standard Mode

The following example imports captures from a directory into the Enterprise Manager
repository. Only those captures specifically named in the XML input file are imported.

```
emcli import_workloads       -input_file=template:/storage/xml/
importReplay_capture121_5Mins.xml
```

Example 4 - Interactive or Script Mode

The following example imports captures from a directory into the Enterprise Manager
repository. Only those captures specifically named in the XML input file are imported.

```
import_workloads(input_file="template:/storage/xml/
importReplay_capture121_5Mins.xml"      )

import_workloads(include_replays=True,        input_file="template:/storage/xml/
importReplay_capture121_5Mins.xml"      )
```

See Also

> discover_workloads
> import_replays

list

Lists resource data. The maximum number of rows displayed is controlled by OMS
property oracle.sysman.core.dataservice.max_fetch_rows. When the property is not
set, it uses the default value of 2000.

Format

```
emcli list
      [-help]
      [-resource="list_resource_name"]
      [-columns="column_options"]
      [-colsize="column_sizes"]
      [-search="search_options"]
      [-bind="bind_parameters"]
      [-sql="sql"]
      [-script | -format=
                 [name:<pretty|script|csv>];
                 [column_separator:"column_sep_string"];
                 [row_separator:"row_sep_string"];
      ]
      [-noheader]

[ ]  indicates that the parameter is optional
```

Options

• **help**

Lists all resource names with their descriptions. You can use this option with the -resource option to see more details about the resource.

- **resource**

  Resource name for which data is displayed.

- **columns**

  Specify columns as shown, separated by commas:

  ```
  -columns="colname,colname,colname"
  ```

  Example:

  ```
  -columns="COL1,COL3,COL5"
  ```

  Specify column size and width as shown below. A colon precedes the size for a given column.

  ```
  -columns="colname:colsize,colname,colname"
  ```

  Example:

  ```
  -columns="COL1:30,COL3,COL5"
  ```

- **colsize**

  Resizes column widths. Most resource columns have some default widths. You can override them with this option.Example: -colsize="col1:30,col2:5"

- **search**

  You can specify multiple search options. The usage is -search=[ColumnName Operator 'Value', ColumnName Operator 'Value']. The search value must be enclosed in quotes unless searching for null or not null.

  The following operators are supported:

  = !+ > < >= <= like

  The option also supports is null and is not null.

- **bind**

  Use for resources that require specific input. The usage is -bind="Name Operator Value".

- **sql**

  Specifies arbitrary SQL against views. This query is executed as MGMT_VIEW user.

- **script**

  Sets the default column separator to a tab and the default row separator to a newline. You can change the column and row separator strings to change these defaults.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

- format="`name:csv`" sets the column separator to a comma and the row separator to a newline.

- format=column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

- **noheader**

  Displays tabular output without column headers.

Output

When run in script mode, returns JSON output that can be easily parsed.

Exit Codes:

- 0 — Appears when successful.

- 1 — Appears when the list service fails to process the request.

Examples

These examples list all resource names.

Example 1 - Command-Line

```
emcli list -help
```

Example 2 - Scripting and Interactive

```
list (help)
```

These examples list column information about the 'Administrators' resources. They also list which columns users can search.

Example 3 - Command-Line

```
emcli list
      -help
      -resource=Administrators
```

Example 4 - Scripting and Interactive

```
list
      (help
      ,resource=Administrators)
```

Example 5 - Searching

```
list(resource="NamedCredentials",
  search=[searchColumn + " like '" + mySearchPattern + "'", "CredOwner='SYSMAN'"]
```

list_active_sessions

Lists active sessions on all OMSes in the environment. By default, the verb prints a summary for each OMS.

Format

```
emcli list_active_sessions
        [-details
        [-table]
        [-script]
        [-format=name:value;name:value]
        [-noheader]]
```

```
[ ]  indicates that the parameter is optional
```

Options

* **details**

  Displays active user sessions on each OMS. The output format is non-tabular.

* **table**

  Prints details in table format.

* **script**

  Prints output that can be processed by script.

* **format**

  Supports the following name/value pairs:csv — Output will be comma-separated script — Output will be in a format that can be processed by script. You can also specify row_separator and column_separator.

* **noheader**

  Skips the header.

Examples

```
emcli list_active_sessions
    emcli list_active_sessions -details
    emcli list_active_sessions -details -table
    emcli list_active_sessions -details -table -script
    emcli list_active_sessions -details -table -script -noheader
    emcli list_active_sessions -details -table -format="name:csv"
    emcli list_active_sessions -details -table -
format="name:script;row_separator:@@;column_separator:!"
```

list_add_host_platforms

Lists the platforms on which the Add Host operation can be performed.

Format

```
emcli list_add_host_platforms
        [-all]
        [-noheader]
        [-script | -format=
                    [name:<pretty|script|csv>];
                    [column_separator:"column_sep_string"];
                    [row_separator:"row_sep_string"];
        ]
```

```
[ ] indicates that the parameter is optional
```

Options

- **all**

  Displays all of the platforms, including those for which the Agent software is not available.

- **noheader**

  Displays tabular output without column headers.

- **script**

  This option is equivalent to -format="name:script".

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Platform ID, Platform Name

Examples

Example 1

This example displays the platforms for which the agent software is available so that the Add Host operation can be performed.

```
emcli list_add_host_platforms
```

Example 2

This example displays all of the platforms, including those for which the Agent software is not available.

```
emcli list_add_host_platforms -all
```

list_add_host_sessions

Lists all of the Add Host sessions.

Format

```
emcli list_add_host_sessions
        [-host_name="Host name"]
```

```
                    [-session_name="Session name"]
                    [-match_all]
                    [-noheader]
                    [-script | -format=
                               [name:<pretty|script|csv>];
                               [column_separator:"column_sep_string"];
                               [row_separator:"row_sep_string"];
                    ]
```

```
[ ] indicates that the parameter is optional
```

Options

- **host_name**

  Displays all of the Add Host sessions that the provided host is a part of.

- **session_name**

  Displays all of the sessions that match the session name provided.

- **match_all**

  Displays results that match all of the provided query criteria. By default, the results that match any of the provided query criteria are displayed.

- **noheader**

  Displays tabular output without column headers.

- **script**

  This option is equivalent to -format="name:script".

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  – `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Session Name, Deployment Type, Host, Initialization, Remote Prerequisite, Agent Deployment

Examples

Example 1

This example displays all of the Add Host sessions.

```
emcli list_add_host_sessions
```

Example 2

This example displays all of the Add Host sessions that the host 'example.com' was part of, AND whose session name contains the string 'Jan_15'.

```
emcli list_add_host_sessions -host_name=example.com -session_name=Jan_15 -
match_all
```

list_adms

Lists the names, source target name, and application suites of existing Application Data Models.

Format

```
emcli list_adms
```

Output

List of Application Data Models.

Examples

This example lists all Application Data models.

```
emcli list_adms
```

list_agents_on_gold_image

Lists the Management Agents that were deployed or updated using a particular Management Agent gold image version, or the overall agent deployment report for the Management Agent gold image.

Format

```
emcli list_agents_on_gold_image
      -version_name|-image_name="gold_image_version_name|gold_image_name"
      [-agent_name="agent_name_pattern"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   version_name

    Lists the Management Agents that were deployed or updated using a particular Management Agent gold image version.

*   image_name

    Lists the number of Management Agents deployed for given Management Agent gold image.

*   agent_name

    Lists only the Management Agents that match the specified name pattern.

Examples

Example 1

The following example displays the Management Agents that were deployed or updated using the Management Agent gold image `OPC_AGI_DB_JUL_13`.

```
emcli list_agents_on_gold_image
      -version_name=OPC_AGI_DB_JUL_13
```

Example 2

The following example displays the number of Management Agents that were deployed or updated using any of the Management Agent gold image versions that are part of the gold image `OPC_DB_MONITORING`.

```
emcli list_agents_on_gold_image
      -image_name=OPC_DB_MONITORING
```

list_allowed_pairs

Lists allowed association types for the specified source and destination target types.

Format

Standard Mode

```
emcli list_allowed_pairs
       -source_type="source type"
       -dest_type="dest type"]
       [-noheader]
       [-script]
       [-format="[name:<pretty|script|csv>];[column_separator:
       "column_sep_   string"];[row separator:"row_sep_string"]"]
```

Interactive (Script) Mode

```
list_allowed_pairs(
      source_type="source type"
      [,dest_type="dest type"]
      [,noheader=True/False]
      [,script=True/False]
      [,format="[name:<pretty|script|csv>];[column_separator:
      "column_sep_string"];[row_separator:"row_sep_string"]"]
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   **source_type**

    Source target type.

*   **dest_type**

    Destination target type.

*   **noheader**

    Displays the output in tabular output without column headers.

*   **script**

    Prints the output in a format that can be used in scripting.

- **format**

  Specifies how the output is formatted. The default value is "name:pretty", which prints the output table in a readable format not intended to be parsed by scripts. Other format options include:

  – format="name:script" Sets the default column separator to a tab and the default row separator to a newline in the output. You can override the column and row separator strings with your own values.

  – format="name:script;column_separator:<*column_sep_string*>" Causes the verb output to be column-separated by <*column_sep_string*>. Rows are separated by the newline character.

  – format="name:script;row_separator:<*row_sep_string*>" Causes the verb output to be row-separated by <*row_sep_string*>.

  – format="name:script;column_separator:<*column_sep_string*>;row_separator:<*row_sep_string*>"

  – Causes the verb output to be column-separated by <*column_sep_string*> and row- separated by <*row_sep_string*>.

  – format="name:csv" Sets the default column separator to a comma and the default row separator to a newline in the output.

Exit Codes

0 indicates that the verb processing was successful.

Non-zero values indicate that the verb processing was not successful.

Example

This example lists allowed associations for the source target type "cluster" and the destination target type "host":

```
emcli list_allowed_pairs
      -source_target_type="cluster"
      -dest_target_type="host"
```

list_aru_languages

Lists ARU language information.

Format

```
emcli list_aru_languages
      [-name="language_name" | -id="language_id"]
      [-noheader]
      [-script | -format=
                [name:<pretty|script|csv>];
                [column_separator:"column_sep_string"];
                [row_separator:"row_sep_string"];

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Language name.

- **id**

  Language ID.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This option is equivalent to -format="name:script".

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli list_aru_languages
     emcli list_aru_languages -noheader
     emcli list_aru_languages -name="language name" -format="name:pretty"
     emcli list_aru_languages -id="language id" -format="name:script"
```

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode
get_patch_plan_data
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data
show_patch_plan
submit_patch_plan

upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

ORACLE®

list_aru_platforms

Lists ARU platform information.

Format

```
emcli list_aru_platforms
        [-name="platform_name" | -id="platform_id"]
        [-noheader]
        [-script | -format=
                        [name:<pretty|script|csv>];
                        [column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Platform name.

- **id**

  Platform ID.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This option is equivalent to -format="name:script".

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli list_aru_platforms
    emcli list_aru_platforms -noheader
    emcli list_aru_platforms -name="platform_name" -format="name:pretty"
    emcli list_aru_platforms -id="platform id" -noheader -format="name:script"
```

See Also

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

list_aru_products

Lists ARU product information.

Format

```
emcli list_aru_products
       [-name="product_name" | -id="product_id"]
       [-noheader]
       [-script | -format=
                          [name:<pretty|script|csv>];
                          [column_separator:"column_sep_string"];
                          [row_separator:"row_sep_string"];
       ]

[ ]  indicates that the parameter is optional
```

Options

• **name**

  Product name.

• **id**

  Product ID.

• **noheader**

  Displays tabular information without column headers.

• **script**

  This option is equivalent to -format="name:script".

• **format**

Format specification (default is `-format="name:pretty"`).

– `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

– `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

– `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

– `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

– row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli list_aru_products
     emcli list_aru_products -id="product id"
     emcli list_aru_products -name="product name"
     emcli list_aru_products -id="product id" -noheader
     emcli list_aru_products -id="product id" -noheader -script
     emcli list_aru_products -id="product id" -noheader  -format="name:pretty"
```

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data
show_patch_plan
submit_patch_plan

upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

list_aru_releases

Lists ARU release information.

Format

```
emcli list_aru_releases
       [-name="release_name" | -id="release_id" | -productId="product_id"]
       [-noheader]
```

```
            [-script | -format=
                        [name:<pretty|script|csv>];
                        [column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];
            ]
```

`[ ]  indicates that the parameter is optional`

Options

- **name**

  Release name.

- **id**

  Release ID.

- **productId**

  Product ID.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This option is equivalent to -format="name:script".

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli list_aru_releases
     emcli list_aru_releases -noheader
     emcli list_aru_releases -name="release_name" -format="name:pretty"
     emcli list_aru_releases -id="release id" -format="name:script"
     emcli list_aru_releases -productId="product id" -noheader -format="name:csv"
```

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

list_assoc

Lists associations between the specified source and destination targets.

Format

Standard Mode

```
emcli list_assoc
      -source="target_name:target_type"
      -dest="target_name:target_type"[-subseparator="subseparator:attribute_
            name:character"]
      [-noheader]
      [-script
      [-format="[name:<pretty|script|csv>];[column_separator:
               "column_sep_string"];[row_separator:"row_sep_string"]"]
```

Interactive (Script) Mode

```
list_assoc(
      source="target_name:target_type"
      ,dest="target_name:target_type"
      [,subseparator="subseparator:attribute_name:character"]
      [,noheader=True/False]
      [,script=True/False]
      [,format="[name:<pretty|script|csv>];[column_separator:
      "column_sep_string"];[row_separator:"row_sep_string"]"]
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

* **source**

  Source target.

* **dest**

  Destination target.

* **subseparator**

Chapter 5
EM CLI Verbs

By default, multi-value input attributes use a colon (:) as a subseparator. Specifying this option overrides the default subseparator value.

Example: `subseparator="<attribute_name=sep_char>"` where *attribute_name* is the name of the attribute for which you want to override the separator character, and *sep_char* is the new subseparator character.Example: `separator="att=#"`

- **noheader**

  Displays the output in tabular output without column headers.

- **script**

  Prints the output in a format that can be used in scripting.

- **format**

  Specifies how the output is formatted. The default value is "name:pretty", which prints the output table in a readable format not intended to be parsed by scripts. Other format options include:

  - format="name:script" Sets the default column separator to a tab and the default row separator to a newline in the output. You can override the column and row separator strings with your own values.

  - format="name:script;column_separator:<*column_sep_string*>" Causes the verb output to be column-separated by <*column_sep_string*>. Rows are separated by the newline character.

  - format="name:script;row_separator:<*row_sep_string*>" Causes the verb output to be row-separated by <*row_sep_string*>.

  - format="name:script;column_separator:<*column_sep_string*>;row_separator:<*row_sep_string*>"

  - Causes the verb output to be column-separated by <*column_sep_string*> and row-separated by <*row_sep_string*>.

  - format="name:csv" Sets the default column separator to a comma and the default row separator to a newline in the output.

Output

Exit Codes

0 indicates that the verb processing was successful.

Non-zero values indicate that the verb processing was unsuccessful.

Example

This example lists all associations between the source target "abc_cluster:cluster" and the destination target "def.oracle.com:host":

```
emcli list_assoc
      -source="abc_cluster:cluster"
      -dest="def.oracle.com:host"
```

list_bipublisher_roles

Lists the BI Publisher roles granted to an Enterprise Manager user as well as the BI Publisher roles granted to an external LDAP group.

ORACLE®

5-575

Format

```
emcli list_bipublisher_roles
      [-user="user"]
      [-external_role="group"]
```

[ ] indicates that the parameter is optional.

Options

* user

  Lists the BI Publisher roles of the Enterprise Manager user.

* external_role

  Lists the BI Publisher roles of the group.

Examples

Example 1

The following command lists the roles of a specific group (TESTGROUPNAME).

```
emcli list_bipublisher_roles -external_role="TESTGROUPNAME"
```

Example 2

The following command lists the roles of a specific user (TESTUSERNAME).

```
emcli list_bipublisher_roles -user="TESTUSERNAME"
```

list_charge_item_candidates

Lists the charge items that can be registered to Chargeback.

Format

```
emcli  list_charge_item_candidates
      -target_type=target_type
      -source_data_type=<metric|config|property>
      [-target_name=target_name]
      [-config_name=config_name]
      [-config_data_source=target_name]
      [-all]
```

[ ] indicates that the parameter is optional

Options

* **target_name**

  Name of a target type.

* **source_data_type**

  Type of source data. Valid values are metric, config, and property.

* **target_name**

If specified, metering and charge data are retrieved only for the named target. If you do not specify a valid target name, or if the specified target has not been enabled, then no data is generated. If this option is not specified, All targets for the specified target-type are included. Required if source_data_type=config.

- **config_name**

  Name of a configuration. Required if source_data_type=config.

- **config_data_source**

  Data source of the configuration. Required if source_data_type=config.

- **all**

  Applies only when source_date=metric. Displays all items, including out-of-box metrics of target type. Without this option, only metric extensions are displayed.

Examples

Example 1

This example lists the metric extensions created for the Oracle Database target type:

```
emcli list_charge_item_candidates
        -target_type="oracle_database"
        -source_data_type="metric"
```

Example 2

This example lists the configuration items of the myCustomCCS configuration for an Oracle Database target named myDatabase:

```
emcli list_charge_item_candidates
        -target_type="oracle_database"
        -source_data_type="config"
        -target_name="myDatabase"
        -config_name="myCustomCCS"
        -config_data_source="CCSDataSource"
```

list_charge_plans

Lists the charge plans in Chargeback.

Format

```
list_charge_plans
      [[-entity_type="entity_type" [-all]]
      [-charge_plan="charge_plan_name" [-all]]
      [-all]

[ ]  indicates that the parameter is optional
```

Options

- **entity_type**

  Entity type for which the charge plans are to be listed.

- **charge_plan**

  Lists details about a specific charge plan.

- all

Lists all active and future plans.

Examples

Example 1

This example provides details about the active version of the charge plan.

```
list_charge_plans -charge_plan="chargePlanName"
```

Example 2

This example provides details about the active and future versions of the charge plan.

```
list_charge_plans -charge_plan="chargePlanName" -all
```

See Also

add_chargeback_entity
assign_charge_plan
assign_cost_center
list_chargeback_entities
list_chargeback_entity_types
list_cost_centers
remove_chargeback_entity
unassign_charge_plan
unassign_cost_center

list_chargeback_entities

List all of the entities added into Chargeback

Format

```
list_chargeback_entities
```

See Also

add_chargeback_entity
assign_charge_plan
assign_cost_center
list_chargeback_entity_types
list_charge_plans
list_cost_centers
remove_chargeback_entity
unassign_charge_plan
unassign_cost_center

list_chargeback_entity_types

Lists all of the entity types supported by Chargeback.

Format

```
list_chargeback_entity_types    -[usage_mode]    -[entity_type="eType"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **usage_mode**

  Lists all of the entity types supported by Chargeback and the corresponding usage modes.

- **entity_type**

  Lists all of the usage modes supported for the particular entity type "eType".

Examples

Example 1

This example lists all of the entity types supported by Chargeback.

```
list_chargeback_entity_types
```

Output:

```
Entity Type-------------------Entity Type Display Name
------------------------------------------------------------------
1. oracle_database--------Database Instance
2. host------------------------Host
```

Example 2

This example lists all of the entity types supported by Chargeback and the corresponding usage modes.

```
list_chargeback_entity_types -usage_mode
```

Output:

```
Entity Type----------------Entity Type Display Name-------------------Usage Mode
------------------------------------------------------------------
1. oracle_database------------------Database Instance----------------------dbMetered
2. oracle_database------------------Database Instance----------------------
dbByService
3. oracle_database------------------Database Instance----------------------cdbBypdb
4. host-------------------------------Host--------------------------hostMetered
```

See Also

add_chargeback_entity
assign_charge_plan
assign_cost_center
list_chargeback_entities
list_charge_plans
list_cost_centers
remove_chargeback_entity
unassign_charge_plan
unassign_cost_center

list_compliance_rules

Provides a list of compliance standard rules with specified values (attributes).

Format

```
emcli list_compliance_rules
      [-attrs="<attr_list>]"
      [-sep="<separator>"]
      [-target_type="<target_type>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- attrs

  Attributes of the compliance standard rule. The <attr_list> is a comma separated, case insensitive, quote enclosed list of the attributes of interest. If no attributes are specified, then all attributes are returned in the following order:

  – rule_guid: Unique identifier of the compliance standard rule.

  – rule_iname: Internal name of the compliance standard rule.

  – rule_dname: Display name of the compliance standard rule in English.

  – description: English description of the compliance standard rule.

  – target_type: Applicable target type of the compliance standard rule.

  – lifecycle_state: Lifecycle status of the compliance standard rule.

  – author: Author of the compliance standard rule.

  – owner: Owner of the compliance standard rule.

  – is_system: "1" if system, "0" if user.

  – rule_dname_nlsid: Rule display name NLSID for non-English users.

  – description_nlsid: Rule description NLSID for non-English users.

  – severity: Severity of the compliance standard rule.

  – rule_type_code: Code representing the type of compliance standard rule (1 - Repository), (2 - Agent), (3 - Monitoring).

  – severity_code: Rule severity code (18 - Minor Warning), (20 - Warning), (25 - Critical).

  – is_system_code: Code to represent whether the rule is system defined (0 - False), (1 - True).

- sep

  Output between column values. If no separator is specified, then a comma "," is used.

- target_type

  Specifies that only rules of that target_type should be returned. If no target_type is specified, then rules for all target types are returned.

Example

The following example lists the ID, internal name, target type, and severity for all compliance standard rules for all target types. Attributes are separated by a colon (:).

```
emcli list_compliance_rules
     -attsr="rule_guid:rule_iname:target_type:severity"
     -separator=":"
```

list_compliance_rules_ca

Provides one corrective action for a compliance standard rule. **Note**: There cannot be more than one corrective action per rule.

Format

```
emcli list_compliance_rules_ca
     [-attrs="<attribute_list>"]
     [-sep="<separator>"]
     [-target_type="<target_type>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- attrs

  Attributes of the corrective action for a compliance standard rule. The <attribute list> is a comma separated, case insensitive, quote enclosed list of the attributes of interest. If no attributes are specified, then all the attributes are returned in the following order:

  – rule_guid: Unique identifier of the compliance standard rule.

  – rule_iname: Internal name of the compliance standard rule.

  – rule_dname: Name of the compliance standard rule in English.

  – target_type: Applicable target type of the compliance standard rule.

  – ca_id: Unique identifier of the corrective action.

  – simultaneous_action: "0" allows more than one corrective action to run simultaneously, "1" allows only one corrective action.

- sep

  Separator character between column values. If no separator is specified, then a comma "," is used.

- target_type

  Specifies that only compliance standard rules of that target type are returned. If no target type is specified, then the compliance standard rules for all target types are returned.

Example

The following example lists the unique ID, internal name, and ID for all corrective actions for the compliance standard rule for all target types. The separator character used between attributes is a colon (:).

```
emcli list_compliance_rule_ca
     -attrs="rule_guid,rule_iname,ca_id"
     -separator=":"
```

list_cost_centers

Lists the cost centers in various formats depending on the options given.

Format

```
list_cost_centers
     -[[cost_center_name="cName"]
     -[parent]
     -[children]
     -[top]
     -[leaf]]

[ ]  indicates that the parameter is optional
```

Options

*   **cost_center_name**

    Name of the cost center for which further details like parent/children/top/leaf should be listed.

*   **parent**

    Provides the parent cost center of the given cost center.

*   **children**

    Provides the list of child cost centers of the given cost center.

*   **top**

    Provides the hierarchy of the given cost center from the top.

*   **leaf**

    Provides the leaf nodes of the given cost center.

Examples

Example 1

This example provides the parent of the given cost centers.

```
list_cost_centers -cost_center_name="c11" -parent
```

Output:

```
Parent Node
----------------
c1
```

Example 2

This example provides a list of all the child cost centers of the given cost center

```
list_cost_centers -cost_center_name="c1" -children
```

Output:

```
Children Nodes
--------------------
c11
c12
```

See Also

add_chargeback_entity
assign_charge_plan
assign_cost_center
list_chargeback_entities
list_chargeback_entity_types
list_charge_plans
remove_chargeback_entity
unassign_charge_plan
unassign_cost_center

list_cs_rules

Returns unique identifiers for the compliance standard rules referenced by the compliance standard. If the rule is part of the root standard, then the runtime identifier, the root standard identifier, and the compliance standard identifier will be the same. If the rule is part of the included standard (ics) hierarchy, then the runtime identifier and the compliance standard identifier will be the same but different from the root standard identifier.

Format

```
emcli list_cs_rules
      -cs_iname="<internal_name_of_standard>"
      -author="<author>"
      -version="<version>"
      [-attrs="<attribute_list>"]
      [-sep="<separator>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   cs_iname: Internal name of compliance standard.

*   author: Author of compliance standard.

*   version: Version of compliance standard.

*   attrs

    Attributes of the compliance standard rules. The <attribute_list> is a comma separated, case insensitive, quote enclosed list of the attributes of interest. If no attributes are specified, then all attributes are returned in the following order.

    –   root_cs_guid: Unique identifier of the root standard with which the rule is associated.

    –   rqs_guid: Unique runtime identifier of the rule referenced with the root standard.

    –   cs_guid: Compliance standard identifier.

    –   rule_iname: Internal name of the compliance standard rule.

    –   target_type: Target type associated with the compliance standard rule.

*   sep

Separator character between column values. If no separator is specified, then a comma "," is used.

Example

The following example lists compliance standard rules for the second version of the security standard and authored by Jones.

```
emcli list_cs_rules
      -cs_iname="security_standard"
      -author="Jones"
      -version="2"
```

list_custom_plugin_updates

Lists all of the custom plug-in updates imported to Enterprise Manager to date. Only one custom plug-in update can be imported for each plug-in version and revision combination.

Format

```
emcli list_custom_plugin_updates
```

Example

The following example lists all of the custom plug-in updates imported to Enterprise Manager to date.

```
emcli list_custom_plugin_updates
```

list_database_sizes

Lists all of the database sizes that have been created.

Format

```
emcli list_database_sizes
      [-name="<Existing size name>"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- name

  A complete or a partial string. If the name parameter is specified, only database sizes that include the specified string are returned.

Examples

Example 1

The following command finds all database sizes that have been created.

```
emcli list_database_sizes
```

Output:

```
Name:Extra-Small
Description:Extra-small database size
```

```
CPU(cores):4
Memory(GB):4
Storage(GB):Not Specified
Processes(Units):Not Specified

Name:Small
Description:Small database
CPU(cores):8
Memory(GB):8
Storage(GB):Not Specified
Processes(Units):Not Specified

Name:Medium
Description:Medium
CPU(cores):8
Memory(GB):16
Storage(GB):Not Specified
Processes(Units):Not Specified
```

Example 2

The following command finds all database sizes that include 'Extra' in the name string.

```
emcli list_database_sizes
-name="Extra*"
```

Output:

```
Name:Extra-Small
Description:Extra-small database size
CPU(cores):4
Memory(GB):4
Storage(GB):Not Specified
Processes(Units):Not Specified
```

list_dbprofiles

Lists all the database profiles.

Format

```
emcli list_dbprofiles
     [-details]
```

[ ] indicates that the parameter is optional.

Options

• details

  Shows the details for each database profile.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example lists all the existing database profiles in detail:

```
emcli list_dbprofiles -details


     Output:
          Name=RMAN Profile,Location=Database Provisioning Profiles/11.2.0.4.0/
linux_x64/,Type=RMAN,Status=Ready,Description=Database Reference Profile
04-11-2014 12:40 PM from database.mycompany.com
          Version : 11.2.0.4.0,contains=Structure and
Data,removalOverdue=0,sourceDatabaseName=database.mycompany.com.
          Name=DB Template,Location=Database Provisioning Profiles/11.2.0.4.0/
linux_x64/,Type=DBCA_TEMPLATE,Status=Ready,Description=Database Reference
Profile 03-11-2014 04:55 PM from database.mycompany.com
          Version : 11.2.0.4.0,contains=Structure
only,removalOverdue=0,sourceDatabaseName=database.mycompany.com.
          Name=Snapshot Profile,Location=Database Provisioning Profiles/11.2.0.4.0/
linux_x64/,Type=SNAPSHOT,Status=Ready,Description=Database Reference Profile
05-11-2014 03:09 PM from database.mycompany.com
          Version : 11.2.0.4.0,contains=Structure and
Data,removalOverdue=2,sourceDatabaseName=database.mycompany.com.
```

list_diagcheck_exclude_applies

Displays the list of targets using a diagcheck exclusion.

Format

```
emcli list_diagcheck_exclude_applies
    -target_type="target type"
    -exclude_name="name"
```

Options

- target_type

  The target type.

- exclude_name

  The exclusion name.

list_diagcheck_exclusions

Gets the list of diagnostic check exclusions defined for a target type.

Format

```
emcli list_diagcheck_exclusions
      -target_type="type"
```

Options

- **target_type**

  Type of target.

list_diagchecks

Gets the list of diagnostic check exclusions defined for a target type.

Format

```
emcli list_diagchecks
        -target_type="type"
        [-version="<diag_version>" ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **target_type**

  Type of target.

- **version**

  Diagnostic version. Defaults to the latest version.

list_fmw_profiles

Lists all available Fusion Middleware provisioning profiles in the software library.

Format

```
emcli list_fmw_profiles
     [-source_type="Profile Source"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- source_type

  Specify one source type to view only profiles of that type. Valid values are weblogic_domain, oracle_home, or install_media.

Example

The following example displays all available Weblogic domain provisioning profiles in the software library.

```
emcli list_fmw_profiles      -source_type="weblogic_domain"
```

list_gold_agent_image_activities

Lists the activities that the specified Management Agent gold image is a part of.

Format

```
emcli list_gold_agent_image_activities
     -version_name="gold_image_version_name"
     [-noheader]
     [-script | -format=
             [name:<pretty|script|csv>];
             [column_separator:"column_sep_string"];
             [row_separator:"row_sep_string"];
```

```
[ ]  indicates that the parameter is optional.
```

ORACLE®

Options

- version_name

  Management Agent gold image version whose activities you want to view.

- noheader

  Displays the output in tabular format, without any column headers.

- script

  Displays the output in script format. This option is equivalent to -format="name:script". -format="name:script" sets the default column separator to a tab character and the default row separator to a new line character. You can specify the column_separator and row_separator strings to change these default characters.

- format

  Output format you want. The default value of this option is -format="name:pretty".

  The following are the supported output formats.

  -format="name:pretty" displays the output table in a readable format that cannot be parsed by scripts.

  -format="name:script" sets the default column separator to a tab character and the default row separator to a new line character. You can specify the column_separator and row_separator strings to change these default characters.

  -format="name:csv" sets the column separator to a comma and the row separator to a new line character.

Example

The following example lists the activities that the Management Agent gold image `OPC_AGI_DB_JUL_13` is part of.

```
emcli list_gold_agent_image_activities      -version_name=OPC_AGI_DB_JUL_13
```

list_gold_agent_images

Lists the various Management Agent gold images that have been created.

Format

```
emcli list_gold_agent_images
      [-noheader]
      [-script | -format=
            [name:<pretty|script|csv>];
            [column_separator:"column_sep_string"];
            [row_separator:"row_sep_string"];
```

```
[ ]  indicates that the parameter is optional.
```

Options

- noheader

  Displays the output in tabular format, without any column headers.

- script

Displays the output in script format. This option is equivalent to -format="name:script".

- format

  Output format you want. The default value of this option is -format="name:pretty".

  The following are the supported output formats:

  -format="name:pretty" displays the output table in a readable format that cannot be parsed by scripts.

  -format="name:script" sets the default column separator to a tab character and the default row separator to a newline character. You can specify the column_separator and row_separator strings to change these default characters.

  -format="name:csv" sets the column separator to a comma and the row separator to a newline character.

Example

The following example displays all the Management Agent gold image that have been created.

```
emcli list_gold_agent_images
```

list_gold_agent_imageversions

Lists the Management Agent gold image versions that have been promoted to the Current status by default.

Format

```
emcli list_gold_agent_imageversions
      [-image_name="gold_image_name"]
      [-all]
      [-noheader]
      [-script | -format=
                [name:<pretty|script|csv>];
                [column_separator:"column_sep_string"];
                [row_separator:"row_sep_string"];

[ ]  indicates that the parameter is optional.
```

Options

- image_name

  Name of a particular Management Agent gold image whose Management Agent gold image versions should be listed.

- all

  Lists all the Management Agent gold images.

- noheader

  Displays the output in tabular format, without any column headers.

- script

  Output format you want. The default value of this option is equivalent to -format="name:script".

  The following are the supported output formats

-format="name:pretty" displays the output table in a readable format that cannot be parsed by scripts.

-format="name:script" sets the default column separator to a tab character and the default row separator to a newline character. You can specify the column_separator and row_separator strings to change these default characters.

-format="name:csv" sets the column separator to a comma and the row separator to a newline character.

Examples

Example 1

The following example displays the Management Agent gold image versions that are promoted to the `Current` status.

```
emcli list_gold_agent_imageversions
```

Example 2

The following example displays the Management Agent gold image versions that are part of the `OPC_DB_MONITORING` and promoted to the `Current` status.

```
emcli list_gold_agent_imageversions
       -image_name=OPC_DB_MONITORING
```

emcli list_gold_image_subscribed_agent

Displays a list of Management Agents subscribed to a given Management Agent Gold Image.

Format

```
emcli list_gold_image_subscribed_agent
-image_name="gold_image_name"
[-noheader]
[-script | -format=
[name:<pretty|script|csv>];
[column_separator:"column_sep_string"];
[row_separator:"row_sep_string"];
]

[ ] indicates that the parameter is optional.
```

Parameters

- image_name

  Image name of a particular Management Agent that is subscribed to a Management Agent Gold Image.

- noheader

  A tabular form of the output without column headers.

- script

  This is equivalent to -format="name:script".

- format

Format of a particular Management Agent that is subscribed to a Management Agent gold image.

Examples

Example

The following example displays all the agents subscribed to OPC_DB_MONITORING image:

```
emcli list_gold_image_subscribed_agent
-image_name=OPC_DB_MONITORING
```

list_internal_metrics

Lists all available internal metrics in an OMS.

Format

```
emcli list_internal_metrics        [-oms_name=<specific oms name> ]
```

```
[ ]  indicates that the parameter is optional.
```

Options

* oms_name

  The name of the target OMS . The explicit OMS name can be found in the Cloud Control console Management Services page. To navigate to this page, from the Setup menu, select Manage Cloud Control and then Management Services. In the Servers area, look for the full name of the Management Service (<host name>:<port number>_Management_Service).

  Note: You only need to specify the oms_name option if you are attempting to access a specific OMS in a multi-OMS environment. If you omit the oms_name option, the list_internal_metric verb will access the OMS running the current instance of EMCLI..

Examples

Example 1

The following example generates a list of internal metrics from an Enterprise Manager repository named "myserver.myco.com:17999_Management_Service".

```
emcli list_internal_metrics -oms_name=myserver.myco.com:17999_Management_Service
```

Example 2

The following example generates a list of internal metrics from the OMS currently running EMCLI.

```
emcli list_internal_metrics
```

list_masking_definitions

Gets the list of masking definitions for an associated target and its script status.

Format

```
emcli list_masking_definitions
    [-definition_name=<masking_defn_name_filter>]
```

```
[-adm_name=<application_data_model_filter>]
[-target_type=<target_type_filter>]
[-target_name=<target_name_filter>]
[-string_match]
[-script | -format=[name:<pretty|script|csv>];
                    [column_separator:"column_sep_string"];
                    [row_separator:"row_sep_string"];
]
[-noheader]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **definition_name**

  Masking definition name filter. This can be either a full value or a pattern match (%).

- **adm_name**

  Application Data Model (ADM) name. This can be either a full value or a pattern match (%).

- **target_type**

  Database target type. This can be either 'oracle_database' or 'rac_database'.

- **target_name**

  Database target name. This can be either a full value or a pattern match (%).

- **string_match**

  Uses an exact string match for a target_name and definition_name match.

- **script**

  This option is equivalent to -format='name: script' .

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  – `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

- **noheader**

  Suppresses printing of column headers.

Output Columns

Masking Definition, Database, Status

Examples

Example 1

This example lists the masking definition named mask_hr_data created on a database named testdb.

```
emcli list_masking_definitions -definition_name=mask_hr_data -target_name=testdb
```

Example 2

This example lists all masking definitions with names starting with credit and created on databases with names starting with test.

```
emcli list_masking_definitions -definition_name=credit% -target_name=test%
```

list_mda_finding_types

Displays a list of all finding types registered with Middleware Diagnostic Advisor (MDA). The output may optionally be filtered by target types.

Format

```
emcli list_mda_finding_types      [-target_types="<list of target types>"]
[ ]  indicates that the parameter is optional.
```

Options

• target_types

    Provides a delimited list of target types. The default delimiter is ';'. If this option is provided, only finding types applicable to the provided target types are returned.

Examples

Example 1

The following example lists all of the MDA finding types:

```
emcli list_mda_finding_types
```

Example 2

The following example lists the MDA finding types applicable to specified target types:

```
emcli list_mda_finding_types
    -target_types="weblogic_domain;weblogic_server"
```

list_mda_properties

Lists Middleware Diagnostic Advisor (MDA) properties and their current values from the repository.

Format

```
emcli list_mda_properties
```

Example

The following example lists MDA properties and their current values.

```
emcli list_mda_properties
```

list_mw_profiles

Lists All Available non-Oracle Middleware Provisioning Profiles from software library.

Examples of non-Oracle Middleware include Apache Tomcat, JBoss, etc.

Format

```
emcli list_mw_profiles
```

Example

The following example lists all the generic Middleware Provisioning Profiles.

```
emcli list_mw_profiles
```

list_named_credentials

Lists the named credentials. You can list the credentials you own or have explicit access to.

Format

```
emcli list_named_credentials
      [-cred_name="cred_name"]
      [-cred_owner="cred_owner"]
      [-script | -format=[name:<pretty|script|csv>];
                        [column_separator:column_sep_string];
                        [row_separator:row_sep_string];
      [-separator="separator:attname:charseq"]
      [-noheader]

[ ]  indicates that the parameter is optional
```

Options

- **cred_name**

  Credential name to filter the list of credentials displayed.

- **cred_owner**

  Credential owner to filter the list of credentials displayed.

- **script**

  This is equivalent to -format='name: script'.

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

- – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

- – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

- – format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

- – format="name:script;row_separator:<row_sep_string>" row-separates the verb output by <row_sep_string>. Columns are separated by the tab character.

- **separator**

  Multi-value attributes use the semi-colon character as the separator. When data contains this character, you can override its value. For example:

  ```
  separator="<attributename=sep_char"
  ```

  ... where 'attributename' is the name of the attribute for which you want to override the separator character, and 'sepchar' is the new separator character. For example:

  ```
  separator="att=#"
  ```

- **noheader**

  Suppresses printing of column headers in tabular output.

Exit Codes

0 if successful. A non-zero value means that verb processing was unsuccessful.

Examples

This example lists credentials matching credential names containing 'NC'.

```
emcli list_named_credentials -cred_name="NC"
```

list_oms_config_properties

Lists the OMS configuration properties.

Format

```
emcli list_oms_config_properties
        [-oms_name="omsName"]
        [-details]

[ ]  indicates that the parameter is optional
```

Options

- **oms_name**

  Name of the OMS from where the properties have to be retrieved.

- **details**

  Displays the details about from where the property value has been derived, and also the global and default values for the property.

Examples

Example 1

This example lists the entire set of properties.

```
list_oms_config_properties
```

Example 2

This example lists all the properties set on the management server myhost:1159_Management_Service.

```
list_oms_config_properties -oms_name="myhost:1159_Management_Service"
```

list_oms_logging_properties

Lists the logging configuration properties.

Format

```
emcli list_oms_logging_properties
       [-oms_name="omsName"]
       [-details]

[ ]  indicates that the parameter is optional
```

Options

- **oms_name**

  Name of the OMS from where the logging properties have to be retrieved.

- **details**

  Displays the details about from where the property value has been derived, and also the global and default values for the logging property.

Examples

Example 1

This example lists the entire set of logging properties.

```
list_oms_logging_properties
```

Example 2

This example lists all the logging properties set on the management server myhost:1159_Management_Service.

```
list_oms_logging_properties -oms_name="myhost:1159_Management_Service"
```

list_patch_plans

Lists existing patch plans. You can list all the existing patch plans and can also list the existing patch plans whose names match the specified pattern.

Format

```
emcli list_patch_plans
        [-name="name"]
        [-noheader]
        [-script | -format=
                        [name:<pretty|script|csv>];
                        [column_separator:"column_sep_string"];
                        [row_separator:"row_sep_string"];
        ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Plan name used for searching patch plans. If you do not specify this option, the patch plan whose name is the same as the specified name, or contains the specified name string, will be listed. If you do not specify this option, all of the existing patch plans are listed.

- **noheader**

  Suppresses printing of column headers.

- **script**

  This option is equivalent to -format='name: script' .

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli list_patch_plans
    emcli list_patch_plans -name="plan name" -noheader
    emcli list_patch_plans -name="plan name" -noheader -script
    emcli list_patch_plans -name="plan name" -noheader -format="name:pretty"
    emcli list_patch_plans -name="plan name" -noheader
    -format="name:pretty";column_separator="separator"
```

See Also

create_patch_plan
delete_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

list_patches_in_custom_plugin_update

Lists all of the patches included in the custom plug-in update for a particular plug-in.

Format

```
emcli list_patches_in_custom_plugin_update
      -plugin="<plugin_id>:<plugin_version>:<plugin_revision>"
      [-discovery]

[ ]  indicates that the parameter is optional.
```

Options

- **plugin**

  ID, version, and revision of the plug-in. To view the version and revision of a plug-in, run the emcli list_custom_plugin_updates verb.

- **discovery**

  Lists even patches with the discovery component of the plug. If not passed, only patches with the monitoring component of the plug-in are listed.

Examples

Example 1

The following example lists all of the patches included in the custom plug-in update of the 12.1.0.2.0 version of the `oracle.sysman.db2` plug-in. The patch list includes patches that contain only the monitoring component of the plug-in.

```
emcli list_patches_in_custom_plugin_update
      -plugin="oracle.sysman.db2:12.1.0.2.0"
```

Example 2

The following example lists all the patches included in the custom plug-in update of the 12.1.0.2.0 version of the `oracle.sysman.db2` plug-in. The patch list includes patches

that contain not only the monitoring component but also the discovery component of the plug-in.

```
emcli list_patches_in_custom_plugin_update
      -plugin="oracle.sysman.db2:12.1.0.2.0"
      -discovery
```

list_plugins_on_agent

Lists all of the plug-ins deployed on the management Agents.

Format

```
emcli list_plugins_on_agent
        [-agent_names="agent1,agent2,agent3 "
        [-all]
        [-include_discovery]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **agent_names**

    All of the management Agents(host:port) whose deployed plug-ins need to be listed. If you do not provide any Agent names, plug-ins on all Agents are listed. String literals with a wildcard (*) expression are accepted. For example:

    ```
    emcli list_plugins_on_agent -agent_names='adc*,st*93'
    ```

*   **all**

    Lists plug-ins on all the management's Agents.

*   **include_discovery**

    Includes discovery components of the plug-ins. By default, discovery components of the plug-ins are ignored.

Examples

Example 1

This example lists plug-ins on the Agent abc.example.com.

```
emcli list_plugins_on_agent -agent_names=abc.example.com:3872
```

Example 2

This example lists plug-ins for both of the Agents as well as their discovery components.

```
emcli list_plugins_on_agent -agent_names=
abcd.example.com:3872,efgh.example.com:3872 -include_discovery
```

list_plugins_on_server

Lists all of the plug-ins that are deployed on the OMS instances.

Format

```
emcli list_plugins_on_server
      [-details]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **details**

  Displays the plug-in home location.

Examples

Example 1

The following example lists all the plug-ins that are deployed on the OMS instances.

```
emcli list_plugins_on_server
```

Example 2

The following example lists all of the plug-ins, with their plug-in home locations, which are deployed on the OMS instances.

```
 emcli list_plugins_on_server
      -details
```

list_Pluggable_Database_Profiles

Self Service Administrator can list the existing PDB Data Profiles using the below emcli command.

```
emcli list_dataprofiles -details
```

The PDB data profiles can be listed based on the criteria.

Format

```
emcli list_dataprofiles -owner=SSA1
emcli list_dataprofiles  -name=cdb:m
    emcli list_dataprofiles -owner=SSA1 -name=cdb:m
```

Options

- **owner**

  Owner of the dataprofile entity.

- **name**

  Name of the dataprofile entity. This value is not case-sensitive match. ":m" in the above emcli determines if a character string matches a pattern.

Create a PDB Data Profile on Deletion

Format

```
emcli cleanup_dbaas_requests -ids=<requested> -preserve_backup_of_instance -
save_as="profile
      name " -description="profile description"
```

Options

- **-save_as**

  Profile component name

- **-description**

  Profile description

Example

```
emcli cleanup_dbaas_requests -ids=<requested> -preserve_backup_of_instance -
save_as="profile name " -description="profile description"
```

list_prerequisites

Displays a list of Enterprise Manager repository-related prerequisites.

Format

```
emcli list_prerequisites
     -db_user=<database_user>
     -db_password=<database_password>
     -db_role=<database_role>
     -repos_user=<repository_user>
     [-prerequisite_xml_root_dir=<xml_root_directory_for_platform_prerequisites>]
     [-prerequisite_resource_locs="<xml_resource_location_for_platform/
     plug-in_prerequisites>"]
     [-log_loc=<location_for_log_files_of_EMPrereqKit_tool>]
     [-upgrade_version=<EM_version_to_which_upgrade_is_being>]
     [-configuration_type=<configuration/deployment_type>]

[ ]  indicates that the parameter is optional.
```

Options

- db_user

  Database user account with which a connection to the database can be established, for example SYS.

- db_password

  Database user account password. If you do not provide here, you will be prompted for the password.

- db_role

  Database role. For example, sysdba. Required only when the -db_user value is SYS.

- repos_user

  Repository user account with which the prerequisite checks can be run, for example, SYSMAN. Required only when the -db_user value is SYS.

- prerequisite_xml_root_dir

  Absolute path to the `requisites/list` directory where of the all prerequisite XMLs are located. This is an optional parameter and if not provided, the value is calculated internally. The XML files can be in a subdirectory within the `requisites/list` directory,

but make sure the path that you enter leads only up to the `list` directory. For example, `$<OMS_HOME>/install/requisites/list`.

- prerequisite_resource_locs

    Absolute path to the directory where the plug-in opar files or the platform/plug-in binaries, which contains XML files for platform or plug-in prerequisite checks, are located. This option is not mandatory. For plug-in opar files, use the format `plugin_id=<<absolute_path_.opar_file>>`. For the plug-in home directory use the format `plugin_id=<<plugin_home>>`.

- log_loc

    Absolute path to a directory where the logs of the execution of the Enterprise Manager prerequisite kit can be stored.

- upgrade_version

    The Enterprise Manager version to which the upgrade is being done. For example, 12.1.0.3. If you have downloaded the Enterprise Manager prerequisite resources for two future versions, for example v1 and v2 through Self-Update then with -upgrade_version, you can see or run the prerequisite of the specified version.

- configuration_type

    Configuration or deployment type. For example, MINI, SMALL, MEDIUM, LARGE. This is an optional parameter, and if not provided, it will be calculated internally.

Examples

Example 1

Displays a list of Enterprise Manager repository-related prerequisites with the configuration type MEDIUM.

```
emcli list_prerequisites
     -db_user=SYS
     -db_password=pwd
     -db_role=sysdba
     -repos_user=SYSMAN
     -prerequisite_xml_root_dir=$ORACLE_HOME/install/requisites/list
     -configuration_type=MEDIUM
```

Example 2

Displays a list of Enterprise Manager repository-related prerequisites with upgrade version 12.1.0.4.

```
emcli list_prerequisites
     -db_user=SYS
     -db_password=pwd
     -db_role=sysdba
     -repos_user=SYSMAN
     -prerequisite_xml_root_dir=$ORACLE_HOME/install/requisites/list
     -upgrade_version=12.1.0.4.0
```

list_privilege_delegation_settings

Lists privilege delegation setting templates available on the server that apply to targets.

Format

```
emcli list_privilege_delegation_settings        [-setting_type="SUDO/
POWERBROKER]"        [-noheader]        [-script | -format=                [name:<pretty|
script|csv>];                [column_separator:"column_sep_string"];
[row_separator:"row_sep_string"];        ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **setting_type**

  Setting type. All applicable settings are displayed if you do not specify this option.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This is equivalent to -format="name:script".

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  – `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli list_privilege_delegation_settings        -setting_type="SUDO"
```

list_target_properties_master_list

Lists the values of a property.

Format

Standard Mode

```
emcli list_target_properties_master_list
    -property_name="null"
    [-details]
```

```
[ ]  indicates that the parameter is optional.
```

Interactive or Script Mode

```
list_target_properties_master_list (
     property_name="null"
     [,details=True/False]
     )
```

Options

- property_name

  Name of the property that you want to view.

- details

  Details of the selected property

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example lists the values in the Master List for the orcl_gtp_lifecycle_status property:

```
emcli  list_target_properties_master_list
     -property_name="orcl_gtp_lifecycle_status"
```

list_prov_parameters

Lists parameters used by the actions to provision a Generic Middleware Provisioning Profile.

Format

```
emcli list_prov_parameters
      -profile="Profile Location"
      [-action="Provisioning Action"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- profile

  Complete software library location of the profile.

- action

  One of the provisioning actions as described in the profile properties. This option is not mandatory. If no value is provided then the default action mentioned in the profile properties will be used.

Example

The following example lists all the parameters used for cloning the profile MyProfile1.

```
emcli list_prov_parameters
            -profile="Middleware Provisioning/Generic Profile/MyProfile1"
            -action="clone"
```

list_proxies

Lists all HTTP(S) proxies with the following details:

- Proxy Name
- Protocol
- Hostname:Port
- Status (indicating if the proxy is up or down)

By default, the output is in tabular format.

> **Note:**
>
> These proxies mediate HTTP(S) traffic from the Oracle Management Server to the Management Agent.

Format

```
emcli list_proxies
        [-noheader]
        [-script
        | -format="name:<pretty | csv>"
        | -format="name:script[;column_separator:<string>][;row_separator:<string>]"

[ ]  indicates that the parameter is optional.
```

Options

- -noheader

  Displays the output without a header row.

- -script

  Enables output to be parsed by a script. Each proxy is displayed in a separate line with its fields separated by a tab. This option is equivalent to the -format="name:script" option

- -format

  Specifies the format of the output. Available options:

  – "name:pretty" - Output format in tabular fashion (default). Each row displays a specific proxy. This format is not intended to be parsed by a script.

  – "name:script" - Output format to be parsed by a script. By default, each proxy is displayed in a separate line with its fields separated by a tab. This format can be customized by using the following format specifiers:

    column_separator: String separating proxy fields

    row_separator: String separating different proxies

  – "name:csv" - Output format in CSV fashion. Each proxy is displayed in a separate line with its fields separated by a comma.

Examples

Example 1

The following command lists all proxies in a tabular format along with the column headers.

```
emcli list_proxies
```

Example 2

The following command lists all proxies with each proxy in a separate line and proxy fields are separated by '|'. The header row listing the proxy field names is not displayed.

```
emcli list_proxies
       -noheader
       -format="name:script;column_separator:|"
```

list_siebel_enterprises

Lists the Siebel enterprises currently discovered in Enterprise Manager.

Format

```
eemcli list_siebel_enterprises
```

Example

This example lists the Siebel enterprises that are discovered in Enterprise Manager.

```
emcli list_siebel_enterprises
```

For example, the listed Siebel enterprises that are displayed are:

```
siebel_enterprise: siebel_slc01nqr.us.example.com
siebel_enterprise: siebel_slc01qhn.us.example.com
```

list_siebel_servers

Lists the Siebel servers present in the specified Siebel enterprise.

Format

```
emcli list_siebel_servers -enterprise=<Siebel enterprise>
```

Options

*   **enterprise**

    Indicates the fully-qualified name of the Siebel enterprise.

    For example, to list servers under a Siebel enterprise `<Seibel enterprise>`, enter the option as: `-enterprise=<Siebel enterprise>`.

> **✎ Note:**
>
> The command `emcli list_siebel_enterprises` can be used to list the currently monitored Siebel enterprises in EM.

Example

This example lists the Siebel servers present in the `siebel_slc01nqr.us.example.com` Siebel enterprise in Enterprise Manager.

```
emcli list_siebel_servers -enterprise=siebel_slc01nqr.us.example.com
```

list_sla

Lists the SLA life-cycle status and version information for a target. If you specify the slaName, the command prints the summary information of the different versions. If you do not specify the slaName, the command prints all the available SLA version series for a target. When you specify the version, this commands prints only summary information for the specified version.

Format

```
emcli list_sla
    -targetName=<target_name>
    -targetType=<target_type>
    [-slaName=<SLA_name>]

[ ]  indicates that the parameter is optional
```

Options

- **targetName**

  Name of the target.

- **targetType**

  Type of target.

- **slaName**

  Name of the SLA.

Examples

Example 1

This example prints the SLA information for one SLA.

```
emcli list_sla
      -targetName='my_service' -targetType='generic_service'
      -slaName='gold_sla' -version=2
```

Example 2

This example prints the SLA information for all SLAs of a target.

```
emcli list_sla          -targetName='my_service' -targetType='generic_service'
```

list_standards

Returns the specified attributes optionally filtered by the target type.

Format

```
emcli list_standards
      [-attrs="<attribute_list>"]
      [-sep="<separator>"]
      [-target_type="<target_type>"]

[ ]  indicates that the parameter is optional.
```

Options

- attrs

  Attributes of the compliance standard rule. The <attribute_list> is a comma separated, case insensitive, quote enclosed list of the attributes of interest. If no attributes are specified, then all attributes are returned in the following order:

  – cs_guid: Unique identifier of the compliance standard.

  – cs_iname: Internal name of the compliance standard.

  – cs_dname: Name of the compliance standard name in English.

  – target_type: Applicable target type, for example, host.

  – cs_type: Type of compliance standard. Type the number associated with the desired standard:

    * "1" Repository

    * "2" WebLogic Server Signature

    * "3" Real-time Monitoring

    * "4" Agent side

    * "7" Configuration Drift

    * "8" Configuration Consistency

  – is_system: "0" user defined compliance standard, "1" system defined compliance standard.

  – author: Author of the compliance standard.

  – version: Version of the compliance standard.

  – owner: Owner of the compliance standard.

  – created_date: Date when compliance standard was created.

  – cs_dname_nlsid: National Language Support (NLS) identifier of cs_dname.

  – description_nlsid: National Language Support (NLS) identifier of description.

  – front_matter_nlsid: National Language Support (NLS) identifier of front matter.

  – rear_matter_nlsid: National Language Support (NLS) identifier of rear matter.

  – notice_nlsid: National Language Support (NLS) identifier of notice.

- sep

Separator character between column values. If no separator is specified, then a comma "," is used.

- target_type

    Specifies that only rules of that target type should be returned. If no target type is specified, then the rules for all target types are returned.

Example

The following example lists all the attributes for the host target.

```
emcli list_standards
     -target_type="host"
```

list_subset_definitions

Gets the list of subset definitions, Application Data Models, and target names.

Format

```
emcli list_subset_definitions
     [-subset_name=<subset_definition_name_filter>]
     [-adm_name=<application_data_model_filter>]
     [-target_name=<target_name_filter>]
     [-string_match]
     [-script | -format=[name:<pretty|script|csv>];
                     [column_separator:"column_sep_string"];
                     [row_separator:"row_sep_string"];
     ]
     [-noheader]

[ ]  indicates that the parameter is optional
```

Options

- **subset_name**

    Filter for the subset definition name. This can either be a full value or a pattern match(%).

- **adm_name**

    Filter for the Application Data Model (ADM) name. This can be either a full value or a pattern match(%).

- **target_name**

    Filter for the database target name. This can be either a full value or a pattern match (%).

- **string_match**

    Uses an exact string match for the subset definition name, target name, and ADM name.

- **script**

    This option is equivalent to -format='name: script'.

- **format**

    Format specification (default is -format="name: pretty").

    – -format="name:pretty" prints the output table in a readable format not intended to be parsed by scripts.

– -format="name:script" sets the default column separator to a tab and the
   default row separator to a newline.

– -format="name:csv" sets the column separator to a comma and the row
   separator to a newline.

- **noheader**

   Suppresses the printing of column headers.

Output Columns

Subset Definition, Adm Name, Target Name

Examples

Example 1

This example prints the SLA information for one SLA.

```
emcli list_sla
      -targetName='my_service' -targetType='generic_service'
      -slaName='gold_sla' -version=2
```

Example 2

This example prints the SLA information for all SLAs of a target.

```
emcli list_sla           -targetName='my_service' -
targetType='generic_service'
```

list_swlib_entities

Lists the entities in the software library based on the specified filter criteria. The results
are printed in the following order:

Display Name, Revision, Description, Status, Type, Subtype, Maturity, Owner, [Folder
Path, Folder Id, Entity Rev Id]

Format

```
emcli list_swlib_entities
        [-name="entity_name"]
        [-folder_id="folder_internal_id"]
        [-desc="entity_desc"]
        [-attr="<attr_name>:<attr_value>"]
        [-type]="type_internal_id"]
        [-subtype]="subtype_internal_id"]
        [-maturity]="maturity"]
        [-owner]="owner"]
        [-status]="status"]
        [-show_folder_path]
        [-show_folder_id]
        [-show_entity_rev_id]

[ ]  indicates that the parameter is optional
```

Options

- **name**

Name of the entity. The value specified for this option is considered for a case-insensitive match.

- **folder_id**

    Internal identifier of the parent folder. The value specified for this option is considered for an exact match.

- **desc**

    Description of the entity. The value specified for this option is considered for a case-insensitive match.

- **attr**

    An attribute and its value, separated by a colon (:). For specifying values for multiple attributes, repeat the option. The value specified for this option is considered for an exact match.

    You can only use this option with the type parameter.

- **type**

    Internal identifier of the entity type. Use the list_swlib_entity_types verb to identify the type.

- **subtype**

    Internal identifier of the entity sub-type. Use the list_swlib_entity_subtypes verb to identify the sub-type.

- maturity

    Maturity of the entity revision. Can be one of: MAT_Untested MAT_Beta MAT_Production

- owner

    Owner of the entity revision.

- status

    Status of the entity revision. Can be one of: STATE_Incomplete STATE_Ready STATE_Deleted

- **show_folder_path**

    Enables printing of the internal path of each entity's folder.

- **show_folder_id**

    Enables printing of the internal ID of each entity's folder. If specified, the value is printed after the value for show_folder_path.

- **show_entity_rev_id**

    Enables printing of the internal ID of each entity. If specified, the value is printed after the value for show_folder_id.

Examples

This example lists all folders under the specified parent folder, and also prints the internal identifier for each folder in the list.

```
emcli list_swlib_entities
    -name="myEntity"
    -type="COMP_Component"
    -attr="PRODUCT:Oracle Database"
    -show_folder_id
```

list_swlib_entity_subtypes

Lists the entity subtypes available in the software library for a specified entity type.

Format

```
emcli list_swlib_entity_subtypes
        [-entity_type_id="type_internal_name"]
        [-show_subtype_id]

[ ]  indicates that the parameter is optional
```

Options

- **entity_type_id**

  Internal identifier of the type.

- **show_subtype_id**

  Enables printing of the internal identifier for the subtype.

Examples

This example lists all subtypes available in the software library for the type 'COMP_Component.'

```
emcli list_swlib_entity_subtypes
        -entity_type_id="COMP_Component"
        -show_subtype_id
```

list_swlib_entity_types

Lists the entity types available in the software library.

Format

```
emcli list_swlib_entity_types
        [-show_type_id]

[ ]  indicates that the parameter is optional
```

Options

- **show_type_id**

  Enables printing of the internal identifier for the type.

Examples

This example lists all of the types available in the software library.

```
emcli list_swlib_entity_types
        -show_type_id
```

list_swlib_folders

Lists folders in the software library.

Format

```
emcli list_swlib_folders
        [-parent_id="parent_folder_id"]
        [-show_folder_path]
        [-show_folder_id]
```

`[ ]  indicates that the parameter is optional`

Options

- **parent_id**

  Internal identifier of the parent folder.

- **show_folder_path**

  Enables printing of the internal path for the folder.

- **show_folder_id**

  Enables printing of the internal identifier for the folder.

Examples

This example lists all folders under the specified parent folder, and prints the internal identifier for each folder in the list.

```
emcli list_swlib_folders
      -parent_id=
"oracle:defaultService:em:provisioning:1:cat:B13B3B7B086458CFE040E80A19AA560C"
      -show_folder_id
```

list_swlib_storage_locations

Lists storage locations configured in the software library.

Format

```
emcli list_swlib_storage_locations
        [-type="OmsShared|OmsAgent|Http|Nfs|ExtAgent"]
```

`[ ]  indicates that the parameter is optional`

Options

- **type**

  Type of the storage location. The default is OmsShared.

Examples

This example lists all locations configured for storage type 'OmsAgent.'

```
emcli +_locations
        -type="OmsAgent"
```

list_target_privilege_delegation_settings

Lists current privilege delegation settings for targets.

Format

```
emcli list_target_privilege_delegation_settings        -
target_names="name1;name2;name3"      [-input_file="FILE:file_path"]     [-
noheader]       [-script | -format=              [name:<pretty|script|
csv>];              [column_separator:"column_sep_string"];
[row_separator:"row_sep_string"];       ]

[ ]  indicates that the parameter is optional
```

Options

- **target_names**

  List of targets. All targets must be of the host type. Either target_names or input_file must be present.

- **input_file**

  Path of the file that has the list of targets. The file should have one target name per line.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **noheader**

  Display tabular information without column headers.

- **script**

  This option is equivalent to -format="name:script".

- **format**

  Format specification (default is -format="name:pretty").

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  – `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli list_target_privilege_delegation_settings         -
target_names="host.example.com;host2.example.com;emcli
list_target_privilege_delegation_settings          -input_file="FILE:/home/
nqureshi/targets.txt"

emcli list_target_privilege_delegation_settings
        -target_names="host.example.com;host2.example.com;
```

list_target_property_names

Lists property names for the global properties.

Format

```
emcli list_target_property_names
```

Options

None.

list_targets_having_property_value

Lists all targets with the specified property value for this specified property name.

Format

Standard Mode

```
emcli list_targets_having_property_value
    -property_name="null"
    -property_value="null"
```

Interactive or Script Mode

```
list_targets_having_property_value(
    property_name="null"
    ,property_value="null"
    )
```

Options

- -property_name

  Name of the property that you want to investigate.

- -property_value

  Value of the property that you are basing your search on.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following command provides the list of targets with the Production property value for orcl_gtp_lifecycle_status:

```
emcli list_targets_having_property_value
  -property_name="orcl_gtp_lifecycle_status"
  -property_value="Production"
```

list_templates

Lists monitoring templates and their display names.

Format

```
emcli list_templates
        [-target_type="target_type"]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **target_type**

Template's target type. If specified, all templates defined for this target type are displayed.

Examples

Example 1

This example lists all templates.

```
emcli list_templates
```

Example 2

This example lists all templates defined for the host target type.

```
emcli list_templates -target_type="host"
```

list_trace

Displays the list of OMS traces for the Oracle Management System.

Format

```
emcli list_trace
```

Options

None.

list_unconverted_udms

Retrieves the list of UDMs that are not yet in a migration session.

Format

```
emcli list_unconverted_udms
     [-templates_only]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **templates_only**

Only lists unconverted UDMs in templates.

Examples

Example 1

This example displays all the UDMs that are not part of a migration session.

```
emcli list_unconverted_udms
```

Exmaple 2

This example displays all the UDMs that are in a template and not part of a migration session.

```
emcli list_unconverted_udms -templates_only
```

lock_user_account

Locks or unlocks an Enterprise Manager administrator's account.

Format

Standard Mode

```
emcli lock_user_account
      -name="name"
      [-unlock]
```

Interactive or Script Mode

```
lock_user_account(
     name="name"
     [,unlock=True/False]    )[ ] indicates that this option is optional
```

Options

• name

The name of the administrator.

• unlock

Unlocks the administrator's account.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

This section contains the following examples.

Example 1

The following example locks the account of the administrator with the name `user`.

Standard Mode

```
emcli lock_user_account
      -name=user
```

Interactive or Script Mode

```
emcli lock_user_account(name-"user")
```

Example 2

The following example unlocks the account of the administrator with the name `user`.

Standard Mode

```
emcli lock_user_account
      -name=user -unlock
```

Interactive or Script Mode

```
emcli lock_user_account(name="user"),
      unlock="True"
```

login

Logs into Enterprise Manager with the given credentials and sets up a session with the OMS.

> **Note:**
>
> To avoid an uncommon occurrence in which multiple emcli sessions are created on the OMS, Oracle recommends that you enter the login command before running a script containing EM CLI commands.

Standard Mode

```
emcli login      -username=<EM_Console_Username>      [-
password=<EM_Console_Password>]      [-force]
```

```
[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
login      (username="<EM_Console_Username>"
[,password="<EM_Console_Password>"]      [,force=True|False])
```

```
[ ]  indicates that the parameter is optional
```

Options

- **username**

  Enterprise Manager user name to be used by all subsequent EM CLI commands when contacting the OMS.

- **password**

  Enterprise Manager user password. If you do not specify this , you are prompted for the password interactively.

  > **Note:**
  >
  > Providing a password on the command line is insecure and should be avoided.

- **force**

Force a login even if there is an existing session. The value must be set to either True or False for Interactive or Script Mode.

Examples

These examples show a login as a different user using newly specified credentials, then a subsequent login using the previous credentials.

Example 1 - Command-Line

```
emcli logout
emcli login -user=new_user -pass=new_user_pass
emcli <verb-name>
emcli logout
emcli login -user=old_user -pass=old_user_pass
```

Example 2 - Scripting and Interactive

```
logout()
login(username="new_user", password="new_user_pass")
<verb-name>
logout()
login(username="old_user", password="old_user_pass")
```

See Also

logout

logout

Terminates the existing session with the OMS. This verb and the login verb are useful when you need to run a particular verb as a different user. After a logout, you need to invoke either the setup verb or login verb before invoking any other emcli verb.

> **Note:**
>
> Verbs executed after 'emcli logout' may fail with the message "Error: Session expired. Run emcli login to establish a session." You need to run the login verb to log in to EM CLI after an 'emcli logout'.

Format

```
emcli logout
```

Options

None.

Examples

This example shows a login as a different user using newly specified credentials, then a subsequent login using the previous credentials.

```
emcli logut
emcli login -user=new_user -pass=new_user_pass
emcli <verb-name>
```

```
emcli logout
emcli login -user=old_user -pass=old_user_pass
```

See Also

login

manage_incident

Assigns an incident to an enterprise manager user. If the incident is already assigned, it will be reassigned to the specified user. If the incident is assigned to the current user, then it is automatically acknowledged.You can optionally add a comment to the incident.

Format

```
emcli manage_incident
      -incident_id
      -assign_to
      [-comment]
```

Options

*   incident_id

    ID of the incident to be updated.

*   assign_to

    Enterprise Manager user to whom the incident is to be assigned. .

*   comment

    Optional comment to be added to the incident.

Examples

This example assigns Incident 2 to the user with the ID JOHN, and includes the comment "This needs to be watched more closely":

```
emcli manage_incident
      -incident_id=2
      -assign_to=JOHN
      -comment="This needs to be watched more closely"
```

manage_storage_access

Grants, revokes, or changes the privilege on a storage.

Format

Standard Mode

```
emcli manage_storage_access
      -storage_name="<storage name>"
      -action="<change_owner|add_grant|revoke_grant>"
      -user="<user name>"
      [-privilege="<VIEW_STORAGE|MANAGE_STORAGE|FULL_STORAGE>"]
```

Interactive or Script Mode

```
manage_storage_access(
        storage_name="<storage name>"
        ,action="<change_owner|add_grant|revoke_grant>"
        ,user="<user name>"
        [,privilege="<VIEW_STORAGE|MANAGE_STORAGE|FULL_STORAGE>"]
        )
```

```
[ ]  indicates that the parameter is optional.
```

Options

- storage_name

  Name of the storage.

- action

  The manage access action to be performed on the storage.

- user

  The user to or from whom the privilege has to be granted or revoked.

- privilege

  The privilege to be granted/revoked.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example grants a Full Storage privilege to the user stradmin2:

```
emcli manage_storage_access
              -storage_name="sunzfs1"
              -action="add_grant"
              -user="stradmin2"
              [-privilege="FULL_STORAGE"]
```

manage_agent_partnership

Overrides Enterprise Manager's default behavior of automatically assigning partner agents to agents. A partner agent is an agent that, in addition to its other functions, is assigned to another agent as its **partner** in order to remotely monitor the availability of that agent and its host. A partner agent is typically in close network proximity, for example, in the same subnet, with the agent that it remotely monitors. An agent can be a partner (remote monitor) of multiple agents. An agent can only have one partner agent assigned to it.

This verb is not meant to be commonly used. It is provided to support special circumstances where an administrator might want to explicitly assign agent partnerships or exclude agents from being partners or exclude agents from being remotely monitored by other agents.

Format

Standard Mode

```
emcli manage_storage_access
        [-add_agent_partnership]
        [-remove_agent_partnership]
        [-enable_agent_partnership]
```

```
[-disable_agent_partnership]
[-partner_agent="partneragent"]
[-monitored_agent="monitoredagent"]
```

Interactive or Script Mode

```
manage_agent_partnership(
        [add_agent_partnership=True/False]
        [,remove_agent_partnership=True/False]
        [,enable_agent_partnership=True/False]
        [,disable_agent_partnership=True/False]
        [,partner_agent="partneragent"]
        [,monitored_agent="monitoredagent"]
        )
```

```
[ ]  indicates that the parameter is optional.
```

Parameters

*   add_agent_partnership

    Assigns a partner agent to an agent. You must also specify the monitored_agent and partner_agent parameters.

*   remove_agent_partnership

    Removes the partnership between a partner agent and the agent that it monitors. For a remotely monitored agent, to remove the relationship between itself and its partner agent, the monitored_agent parameter must be specified. The partner_agent can be optionally specified. For a partner agent to remove the relationships between itself and all agents that it remotely monitors, the partner_agent parameter must be specified. If the monitored_agent parameter is not specified, then all partnerships that the partner agent currently has will be deleted.

*   disable_agent_partnership

    Prevents an agent from being a partner agent or from being a monitored agent depending on the additional parameters used. If the partner_agent parameter is used, then it prevents the specified agent from being a partner agent (remotely monitoring other agents). If the monitored_agent parameter is used, it prevents the specified agent from being remotely monitored by any agent.

*   enable_agent_partnership

    Enables an agent to become a partner agent or a monitored agent based on the additional parameters used. If the partner_agent parameter is used, it enables the specified agent to be a partner agent (remotely monitor other agents). If the monitored_agent parameter is used, it enables the specified agent to be remotely monitored by another agent. Only one of these actions can be specified.

*   monitored_agent

    The name of the agent that is remotely monitored by another agent. It is typically in the form

    ```
    host:port
    ```

, for example

```
myhost.example.com:1830
```

.

- partner_agent

  The name of the agent that will remotely monitor the availability of another agent and its host. It is typically in the form

  ```
  host:port
  ```

  , for example m

  ```
  yhost.example.com:1830
  ```

  .

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Output

Example 1

This example assigns agent2 as the partner agent for agent1:

```
emcli manage_agent_partnership
        -add_agent_partnership
        -monitored_agent=agent1.example.com:1830
        -partner_agent=agent2.example.com:1833
```

Example 2

This example unassigns agent2 as the partner agent for agent1. If agent1 does not have a partner agent, then an exception is thrown.

```
emcli manage_agent_partnership
        -remove_agent_partnership
        -monitored_agent=agent1.example.com:1830
        -partner_agent=agent2.example.com:1833
```

Example 3

This example unassigns agent2 as the partner agent for all the agents that it remotely monitors. If agent1 is not a partner agent for any agent, then an exception is thrown.

```
emcli manage_agent_partnership
        -remove_agent_partnership
        -partner_agent=agent2.example.com:1833
```

Example 4

This example prevents agent3 from being assigned a partner agent. This means agent3 cannot be remotely monitored by another agent.

```
emcli manage_agent_partnership
        -disable_agent_partnership
        -monitored_agent=agent3.example.com:1830
```

Example 5

This example prevents agent4 from being a partner agent for any agent. This means agent4 cannot be used to remotely monitor other agents.

```
emcli manage_agent_partnership
        -disable_agent_partnership
        -partner_agent=agent4.example.com:1833
```

Example 6

This example allows agent3 to be assigned a partner agent to remotely monitor it.

```
emcli manage_agent_partnership
        -enable_agent_partnership
        -monitored_agent= agent3.example.com:1830
```

Example 7

This example allows agent4 to become a partner for other agents. This means agent4 can be used to remotely monitor other agents.

```
emcli manage_agent_partnership
        -enable_agent_partnership
        -partner_agent= agent4.example.com:1833
```

manage_ra

Used to perform Recovery Appliance administration. There are multiple forms of the command, each using a sub-command to perform different Recovery Appliance management operations. Each form of the command uses a specific set of parameters.

Format

```
emcli manage_ra -addProtectedDB
    -ra_target_name="<Recovery Appliance target name>"
    -ra_admin_cred ="<Named credential for Recovery Appliance administrator>"
    -ra_vpc_user_cred="<Named credential for Recovery Appliance recovery catalog
user> "
    (
      (-target_name="<database or group target name>" -
target_type="oracle_database | rac_database | composite")
      | -db_unique_name="<database unique name for a database that is not an
Enterprise Manager target>"
      | -input_file="target_list:<full path name of input file>"
    )
    -protection_policy = "<protection policy name>"
    [-reserved_space = "<reserved space for the protected database in G, T, P>"]
    [-grantee_name = "<comma separated list of the Enterprise Manager users>"]
    [-create_ramv_targets]
    [-schedule=
      {
        start_time:yyyy/MM/dd HH:mm;
        tz:{java timezone ID};
        frequency:interval/weekly/monthly/yearly;
        repeat:#m|#h|#d|#w;
```

```
        months:#,#,...;
        days:#,#,...;
        end_time:yyyy/MM/dd HH:mm;
    }]
```

This form of the command is used to enroll one or more databases for protection by one or more Recovery Appliances. A single database can be specified directly on the command line via either an Enterprise Manager target name or a database unique name (for databases not managed by Enterprise Manager). Multiple databases can be specified via an input file or via an Enterprise Manager group target.

Options

A description of all arguments for this command follows. For each parameter, it is noted whether the argument is required (either on the command line or in the input file), what the default is if it's not required, and whether it can be specified for individual targets in an input file (i.e., whether the parameter can be set on a per-database or per-group basis when the command is run against multiple databases and/or groups). Required arguments can be specified either on the command line or on a per-target basis in an input file. When an input file is used, command line argument values globally apply to all targets listed in the input file, while per-target parameter values specified in the input file override the corresponding command line argument values.

The following conventions are used for the attribute values in the argument descriptions:

- Required: Whether the argument must be specified, either on the command line or on a per-target basis in an input file.

- Default: For optional arguments, whether there is a default value.

- Scope:

  - Command Line Only: The argument can be specified only on the command line, not in an input file, and will apply globally to all database targets involved in the command.

  - Both: The argument can be specified on either the command line, in the input file, or both.

- **ra_target_name**

  The target name of the Recovery Appliance that the specified databases will be enrolled with.

  Required: Yes

  Scope: Both

- **ra_admin_cred**

  The named credential of the Recovery Appliance Administrator for the Recovery Appliance specified by -ra_target_name. If this argument is not specified, preferred credentials will be used.

  Required: Yes

  Default: Preferred credentials for the Recovery Appliance administrator

  Scope: Both

- **ra_vpc_user_cred**

The name of the Recovery Appliance recovery catalog user that will be used for all the databases being enrolled with the Recovery Appliance. This must be a virtual private catalog user, not the Recovery Appliance administrator user.

Required: Yes

Scope: Both

- **target_name**

   Enterprise Manager target name of a single-instance or cluster database to be enrolled with the Recovery Appliance. Alternatively, a group target can be specified, in which case all the database members of the group will be enrolled.

   Required: Yes, unless -database_unique_name or -input_file are specified. (Either -target_name, -database_unique_name, or -input_file must be specified.)

   Scope: Both

- **target_type**

   Target type corresponding to the target specified by -target_name. This can be oracle_database, rac_database or composite.

   Required: Yes if -target_name is specified. Not applicable if -target_name not specified.

   Scope: Both

- **db_unique_name**

   Database unique name of the database to be enrolled as a protected database. This argument is used if the database being enrolled has not been discovered as an Enterprise Manager target. If the database is a target, -target_name/-target_type should be used.

   Required: Yes, unless -target_name or -input_file are specified. (Either -target_name, -database_unique_name, or -input_file must be specified.)

   Scope: Both

- **input_file**

   A file containing information for multiple databases and/or group targets. This is an alternative to the -target_name and -database_unique_name parameters that can be used when there are multiple databases to be enrolled with one or more Recovery Appliances. The entries in the file mirror the command-line parameters.

   Required: Yes, unless -target_name or -db_unique_name are specified. (Either -target_name, -database_unique_name, or -input_file must be specified.)

   Scope: Command Line Only

   The format is as follows:

   – Either target_name and target_type or db_unique_name entries are required for each database.

   – The following parameters are optional (conditionally if noted, otherwise entirely). They can be specified for some or all of the targets. If an option is not specified for a particular target, values specified on the command line for that option will be used for that target. If an option is present in both the input file and command line, the input file value overrides the command-line value.

      * ra_target_name (optional only if corresponding command line argument is specified)

* ra_admin_cred

* ra_vpc_user_cred (optional only if corresponding command line argument is specified)

* protection_policy (optional only if corresponding command line argument is specified)

* reserved_space

* grantee_name

– Input file format:

```
target.0.target_name="<database #1 target name or group target name>"
target.0.target_type="oracle_database|rac_database|composite"
target.0.ra_target_name="<target name of Recovery Appliance with
which database #1 is to be enrolled (or multiple databases if
target_name is group)>"
target.0.ra_admin_cred="<named credential for administrator for
Recovery Appliance specified for database #1>"
target.0.ra_vpc_user_cred="<named credential for Recovery Appliance
virtual private catalog user that will be used to backup database #1>"
target.0.reserved_space="<reserved space on Recovery Appliance for
database #1>"
target.0.protection_policy="<Recovery Appliance protection policy to
be used for database #1>"
target.0.grantee_name="<comma separated list of EM users that will be
performing backups for database #1 and will be granted access to the
above virtual private catalog credentials>"
target.1.target_name="<database #2 target name or group target name>"
target.1.target_type="oracle_database|rac_database|composite"
target.1.ra_target_name="<target name of Recovery Appliance with
which database #2 is to be enrolled (or multiple databases if
target_name is group)>"
target.1.ra_admin_cred="<named credential for administrator for
Recovery Appliance specified for database #2>"
target.1.ra_vpc_user_cred="<named credential for Recovery Appliance
virtual private catalog user that will be used to backup database #2>"
target.1.reserved_space="<reserved space on Recovery Appliance for
database #2>"
target.1.protection_policy="<Recovery Appliance protection policy to
be used for database #2>"
target.1.grantee_name="<comma separated list of EM users that will be
performing backups for database #2 and will be granted access to the
above virtual private catalog credentials>"
```

• **protection_policy**

The name of the Recovery Appliance protection policy to be used for the databases being enrolled.

Required: Yes

Scope: Both

• **reserved_space**

Reserved space to be allocated on the Recovery Appliance for the protected databases. This can be specified in gigabytes, terabytes, or petabytes.

Required: Yes if -database_unique_name is specified, no otherwise.

Default: For any databases for which a value is not specified (either via the command line or an input file), reserved space will be set according to the following formula: ((2 x <size of database in GB>) + 100) GB.

Scope: Both

- **grantee_name**

  A list of Enterprise Manager database administrator users that need access to the Recovery Appliance virtual private catalog user credentials specified by -ra_vpc_user_cred, in order to configure databases to send backups to the Recovery Appliance. (In addition, access is granted to the Recovery Appliance Monitoring View target associated with the virtual private catalog user, if the target has been created.) After being granted access, these users will be able to select the credentials on the Enterprise Manager Backup Settings page or specify them in the EMCLI configure_db_ha -configureRABackup command.

  Required: No

  Default: None

  Scope: Both

- **create_ramv_targets**

  Create Recovery Appliance Monitoring View targets for every unique combination of Recovery Appliance and virtual private catalog user specified on the command line or in the input file.

  Required: No

  Default: Do not create Recovery Appliance Monitoring View targets.

  Scope: Command Line Only

- **schedule**

  Schedule the deployment procedure. If this argument is not provided, the procedure will run immediately.

  Required: No

  Default: Schedule procedure for immediate execution.

  Scope: Command Line Only

  Sub-arguments:

  - start_time - Time when the procedure has to start execution.
    Format should be "yyyy/MM/dd HH:mm"

  - tz - The timezone ID (optional)

  - frequency - Valid values are once/interval/weekly/monthly/yearly. (optional)
    If frequency is set to interval then repeat has to be specified.

    If frequency is set to weekly or monthly, days has to specified.

    If frequency is set to yearly, both days and months have to specified.

  - repeat - Frequency with which the procedure has to be repeated. (Required only if frequency is set to interval)

  - days - Comma separated list of days. (Required only if frequency is weekly, monthly, or yearly)
    If frequency is weekly, then valid range is 1 to 7

If frequency is monthly or yearly, then valid range is 1 to 30

– months - Comma separated list of months. (Required only if frequency is yearly)
Valid range is 1 to 12.

– end_time - End time for procedure executions. (optional)
If it is not specified, procedure will run indefinitely.

Format should be "yyyy/MM/dd HH:mm"

– grace_period - Grace period in minutes(optional)

Examples

Example 01: Enroll one single-instance database ("finance_db") that is an Enterprise Manager target with Recovery Appliance target "Montreal ZDLRA", using protection policy "MY_POLICY" and 50 GB of reserved space.

```
emcli manage_ra -addProtectedDB
   -ra_target_name="Montreal ZDLRA"
   -ra_admin_cred="NC_RASYS"
   -ra_vpc_user_cred="NC_VPC_USER"
   -target_name="finance_db"
   -target_type="oracle_database"
   -protection_policy="MY_POLICY"
   -reserved_space="50G"
```

Example 02: Enroll multiple databases that are members of an Enterprise Manager group target with Recovery Appliance target "Montreal ZDLRA", using protection policy "MY_POLICY". In addition, grant access to the virtual private catalog user credentials to EM users EMUSER_ADMIN and EM_CLOUD_ADMIN. (Note that the reserved space values for each database in the group will be determined using the formula described in the -reserved_space description above.)

```
emcli manage_ra -addProtectedDB
  -ra_target_name="Montreal ZDLRA"
  -ra_admin_cred="NC_RASYS"
  -ra_vpc_user_cred="NC_VPC_USER"
  -target_name="finance_group"
  -target_type="composite"
  -protection_policy="MY_POLICY"
  -grantee_name="EMUSER_ADMIN,EM_CLOUD_ADMIN"
  -create_ramv_targets
```

Example 03: Enroll one single-instance database that is not an Enterprise Manager target with Recovery Appliance target "Montreal ZDLRA", using protection policy "MY_POLICY" and 50 GB of reserved space.

```
emcli manage_ra -addProtectedDB
   -ra_target_name="Montreal ZDLRA"
   -ra_admin_cred="NC_RASYS"
   -ra_vpc_user_cred="NC_VPC_USER"
   -db_unique_name="FINDB"
   -protection_policy="MY_POLICY"
   -reserved_space="50G"
```

Example 04: Enroll multiple databases with multiple Recovery Appliances using an input file.

```
emcli manage_ra -addProtectedDB
 -input_file="target_list:/tmp/dblist"
```

/tmp/dblist input file

The input file used in this example illustrates the ability to specify target (via database and group target types) and non-target databases, and how to specify different Recovery Appliances and different parameter values (credentials, protection policy, reserved space, etc.) for different databases. In this example, individual target and non-target databases are to be enrolled with Recovery Appliance "Montreal ZDLRA", with different virtual private catalog users, protection policies, and reserved space amount. An additional set of databases that are members of group "finance_group" are to be enrolled with Recovery Appliance "Boston ZDLRA", with specific virtual private catalog user, protection policy, and reserved space values to be used for all databases in the group.

```
target.0.ra_target_name="Montreal ZDLRA"
target.0.ra_admin_cred="NC_RASYS"
target.0.ra_vpc_user_cred="VPC_USER1"
target.0.target_name="hr_db"
target.0.target_type="oracle_database"
target.0.reserved_space="500G"
target.0.grantee_name="EMUSER_ADMIN,EM_CLOUD_ADMIN"
target.0.protection_policy="GOLD"
target.1.ra_target_name="Montreal ZDLRA"
target.1.ra_admin_cred="NC_RASYS"
target.1.ra_vpc_user_cred="VPC_USER2"
target.1.db_unique_name="FINDB"
target.1.grantee_name="EMUSER_ADMIN"
target.1.reserved_space="200G"
target.1.protection_policy="SILVER"
target.2.ra_target_name="Boston ZDLRA"
target.2.ra_admin_cred="BC_RASYS"
target.2.ra_vpc_user_cred="BC_VPC_USER"
target.2.target_name="finance_group"
target.2.target_type="composite"
target.2.reserved_space="100G"
target.2.grantee_name="EMUSER_ADMIN,EM_CLOUD_ADMIN"
target.2.protection_policy="BRONZE"
```

Example 05: Enroll multiple databases with multiple Recovery Appliances using an input file that contains multiple group targets. Schedule the procedure to execute on a daily recurring schedule. By default, since all the targets in the input file are group targets, only databases that have joined the groups since the last procedure execution are processed.

```
emcli manage_ra -addProtectedDB
  -input_file="target_list:/tmp/dblist"
  -schedule="start_time:2020/2/10 01:00;tz:PST;frequency:interval;repeat:1d"
```

/tmp/dblist input file

The input file used in this example specifies three groups, two associated with one Recovery Appliance and one associated with a different Recovery Appliance. Each group is also associated with a specific set of parameter values. (Note that a single reserved space value is specified for group "Montreal Group Gold", meaning all databases in that group will be enrolled with that amount of reserved space. No reserved space value is specified for the other groups, meaning reserved space for each database in those groups will be determined using the formula above.) On each recurring execution, the procedure will dynamically distill the list of database members of each group and process that list, automatically picking up any new group members.

```
target.0.ra_target_name="Montreal ZDLRA"
target.0.ra_admin_cred="NC_RASYS"
```

```
target.0.ra_vpc_user_cred="NC_VPC_USER1"
target.0.target_name="Montreal Group Gold"
target.0.target_type="composite"
target.0.reserved_space="500G"
target.0.grantee_name="EMUSER_ADMIN,EM_CLOUD_ADMIN"
target.0.protection_policy="GOLD"
target.1.ra_target_name="Montreal ZDLRA"
target.1.ra_admin_cred="NC_RASYS"
target.1.ra_vpc_user_cred="NC_VPC_USER2"
target.1.target_name="Montreal Group Silver"
target.1.target_type="composite"
target.1.grantee_name="EMUSER_ADMIN"
target.1.protection_policy="SILVER"
target.2.ra_target_name="Boston ZDLRA"
target.2.ra_admin_cred="BC_RASYS"
target.2.ra_vpc_user_cred="BC_VPC_USER2"
target.2.target_name="Boston Group"
target.2.target_type="composite"
target.2.grantee_name="EMUSER_ADMIN"
target.2.protection_policy="GOLD"
```

Example 06: Enroll multiple databases with one Recovery Appliance using an input file. Provide command line values for Recovery Appliance, administrator credentials, virtual private catalog user, and protection policy that apply globally to all databases and groups listed in the input file. Provide per-database values in the input file for reserved space.

```
emcli manage_ra -addProtectedDB
  -input_file="target_list:/tmp/dblist"
  -ra_target_name="Montreal ZDLRA"
  -ra_admin_cred="NC_RASYS"
  -ra_vpc_user_cred="NC_VPC_USER"
  -protection_policy="GOLD"
```

The input file used in this example is as follows. Note the "SILVER" protection policy specified for database "finance_db" overrides the "GOLD" command line value.

/tmp/dblist input file

```
target.0.target_name="finance_db"
target.0.target_type="oracle_database"
target.0.protection_policy="SILVER"
target.0.reserved_space="500G"
target.0.grantee_name="EMUSER_ADMIN,EM_CLOUD_ADMIN"
target.1.target_name="hr_db"
target.1.target_type="rac_database"
target.1.reserved_space="600G"
target.1.grantee_name="EMUSER_ADMIN"
target.2.target_name="test_group"
target.2.target_type="composite"
target.2.reserved_space="400G"
target.2.grantee_name="EMUSER_ADMIN,EM_CLOUD_ADMIN"
```

merge_credentials

Merges all the references of named credentials provided in the source_credential_list into the destination_credential. The verb expects all the named credentials provided to be equivalent. You can list equivalently named credentials using the command emcli get_duplicate_credentials. All the matching duplicate credentials can be merged using the flag merge_all.

Format

```
emcli merge_credentials
        -destination_credential="destination_cred_name[:destination_cred_owner]"
        [-source_credential_list="source_credential_list"]
        [-merge_all]
        [-merge_without_testing]
```

```
[ ]  indicates that the parameter is optional.
```

Options

• **destination_credential**

  Destination credentials to merge the references.

• **source_credential_list**

  Source-named credential list.

• **merge_all**

  Finds all the duplicate credentials and merges.

• **merge_without_testing**

  Merges the credentials without testing the destination credential.

Examples

Example 1

This example merges the named credentials MyOracleCredential2 and MyOracleCredential3 into MyOracleCredential1. If MyOracleCredential1 is equivalent to MyOracleCredential2 and MyOracleCredential3, all the usages of MyOracleCredential2 and MyOracleCredential3 are replaced with MyOracleCredential1.

```
emcli merge_credentials
        -destination_credential="MyOracleCredential1:ADMIN1"
        -source_credential_list=
          "MyOracleCredential2:ADMIN1;MyOracleCredential3:ADMIN3"
```

Example 2

This example finds all the named credentials equivalent to MyOracleCredential1 and merges their usages with MyOracleCredential1.

```
emcli merge_credentials
        -destination_credential=MyOracleCredential1
        -merge_all
```

metric_control

For the specified target type, lists the metrics whose alerts are stateless and therefore can be manually cleared. Both the metric name and metric internal name are provided in the output of this command. To clear the stateless alerts associated with the specified metric, use the clear_stateless_alerts verb.

Format

```
emcli metric_control
        -command=command
        -target_type=type
        -metric_name=name
```

```
[ ]  indicates that the parameter is optional
```

Options

- **command**

  Can be one of the following:

  – disable_metric — Disables loading of the specified metric .

  – enable_metric — Reenables loading of the specified metric.

  – list_disabled_metrics — Lists the metrics currently disabled for loading.

  – flush_metadata_cache — Flushes the metric API metadata cache target_type.

- **target_type**

  Internal target type identifier (host, oracle_database, oc4j, oracle_emrep, oracle).

- **metric_name**

  Internal name of the metric (for example, load for the host target type).

Example

This example disables the loading of the Load metric on the host target type.

```
emcli metric_control -command=disable_metric -target_type=host -metric_name=Load
```

See Also

clear_stateless_alerts

migrate_db

Submits deployment procedures for performing a database consolidation, using source and destination mappings defined in the XML or JSON file passed as input to the command. The file also denotes which of the following migration methods to use:

- Data Guard Physical Standby (minimal downtime)

- RMAN Clone

- Data Pump (full or schema) Export and Import (cross-platform)

- Full Transportable Export and Import (minimal downtime, cross-platform)

- Cross Platform Incremental Backup (optimized, minimal downtime, cross-platform)

> **✎ Note:**
>
> Before attempting a migration using `migrate_db` on Oracle databases 13.3.2.0.0 and later, make sure patch 12.1.0.2.210119 is deployed.

Format

```
emcli migrate_db
      -file=<XML or JSON file path>
      [-exec_mode=<execution mode>]
      [-ignore_pre_req]
```

```
[ ]  indicates that the parameter is optional.
```

Options

* `file`
  Absolute path of the XML or JSON file containing source and destination database mapping.

* `exec_mode`
  Use only when migration method is `ONLINE_DATAGUARD`, as follows:

  – `FULL`: execute all migration steps in the same job, with no user control over downtime. For example, execute the database standby, convert to RAC, and standby switchover steps in the same job.

  – `PRE_DOWNTIME`: execute all steps that do not require the database to be down. For example, execute the database standby and convert to RAC steps in the same job.
    When you use this mode, you must run the migration job again after successful completion of the current job, with the same input file and `exec_mode` option `DOWNTIME`.

  – `DOWNTIME`: execute all steps that require downtime. For example, execute the switchover step in the job.
    Using this mode presupposes that all pre-downtime steps were successfully executed in a job that specified the `PRE_DOWNTIME exec_mode` option.

  If `exec_mode` is unspecified, defaults to `FULL`.

* `ignore_pre_req`
  Ignores all prerequisite XML validation.

Examples

Example 1

The following example deploys a procedure to perform a database migration, based on the mappings specified in the `dgpredt-migrate.xml` file. The job uses the Data Guard migration method in which it will execute all steps that do not require database downtime.

```
emcli migrate_db
      -file="/home/migrations/dgpredt-migrate.xml"
      -exec_mode="PRE_DOWNTIME"
```

Example 2

The following example deploys a procedure to perform a database migration, based on the mappings specified in the `dgdt-migrate.xml` file. The job uses the Data Guard

migration method in which it will execute all steps that require database downtime. In addition, there will be no prerequisite validation of the XML.

```
emcli migrate_db
        -file="/home/migrations/dgdt-migrate.xml"
        -exec_mode="DOWNTIME"
        -ignore_pre_req
```

Example 3

The following example uses a valid migration XML file.

> **Note:**
>
> When migrating multiple databases to the same target CDB a separate `storageDirectory` must be passed in the XML file for each PDB.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<moveDB type="ONLINE_TTS">
    <mapping>
        <source>
            <targetName>example.source.com</targetName>
            <targetType>oracle_database</targetType>
            <sysdbaCreds>MY_SYSDBA</sysdbaCreds>
            <dbHostCreds>MY_HOST</dbHostCreds>
            <preScript>/prescript.sql</preScript>
            <workingDirectory>/duser/predb</workingDirectory>
            <phase>prepare</phase>
            <pdbName>ORCL18C_PDB1</pdbName>
            <pdbAdminCreds>MY_PDB_CREDS</pdbAdminCreds>
        </source>

        <destination>
            <targetName>example.destination.com</targetName>
            <targetType>oracle_database</targetType>
            <sysdbaCreds>MY_SYS</sysdbaCreds>
            <dbHostCreds>MY_HOST</dbHostCreds>
            <postScript>/postscript.sql</postScript>
            <workingDirectory>/duser/postsql</workingDirectory>
            <storageDirectory>+DATA</storageDirectory>
            <storageType>ASM_STORAGE</storageType>
            <asmCreds>MY_ASMCREDS</asmCreds>
        </destination>
    </mapping>
</moveDB>
```

migrate_noncdb_to_pdb

Migrates a non-container database (non-CDB) as a PDB.

Format

```
emcli migrate_noncdb_to_pdb
 -cdbTargetName="EM CDB target into which the database will be added as PDB"
 -cdbTargetType="EM CDB target type (oracle_database|rac_database)"
 -cdbDBCreds="Named DB credentials of CDB user having sysdba privileges"
 -cdbHostCreds="Named host credentials for Oracle Home owner of CDB"
 -migrationMethod="Migration method to be used (DATAPUMP|PLUG_AS_PDB)"
 -noncdbTargetName="EM non-CDB target to be migrated"
 -noncdbTargetType="EM non-CDB target type (oracle_database|rac_database)"
 -noncdbDBCreds="Named DB credentials for non-CDB user having sysdba privileges"
 -noncdbHostCreds="Named host credentials for Oracle Home owner of non-CDB"
 -pdbName="Name of the PDB to be created on the CDB"
 -pdbAdminName="Username of the PDB administrator to be created"
 -pdbAdminPassword="Password for the PDB administrator"
 [-exportDir="Temporary file system location on the non-CDB host where the
exported files will be stored"]
 [-importDir="Temporary file system location on the CDB host used to stage the
migration metadata and/or datafiles"]
 [-useOMF="Use OMF for datafile location if CDB is OMF enabled (Y|N)"]
 [-createAsClone="If -migrationMethod is 'PLUG_AS_PDB' and if 'createAsClone'
is 'Y', the PDB will be created as clone. (Y|N)"]
 [-dataFilesLoc="Location on the CDB host where datafiles for the newly created
PDB will be stored. Disk Group name in case of ASM"]
 [-encryptionPwd="Password to decrypt/encrypt datapump dump file. Mandatory if
non-CDB contains encrypted tablespaces"]
 [-cdbWalletPwd="Wallet password of the CDB. Mandatory if non-CDB contains
encrypted tablespaces"]
 [-objectExistsAction="Action to be taken when the exported object with same
name is found on the newly created PDB (SKIP|REPLACE). Defaulted to SKIP"]
 [-precheck="Perform pre-requisite checks (YES|NO|ONLY). Defaulted to YES"]
 [-ignoreWarnings="Ignore the warnings from precheck (Y|N)"]
 [-locationSharedAcrossTargets="Export/Import location is shared across Source
and Destination targets)"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- **cdbTargetName**

  Name of the Container Database (CDB) target in Enterprise Manager where the migration will take place.

- **cdbTargetType**

  Target type of the CDB. "oracle_database" for single instance database and "rac_database" for a cluster database.

- **cdbDBCreds**

  Named database credentials on the CDB of a DB user with sysdba privileges.

- **cdbHostCreds**

  Named host credentials of the user who owns the Oracle Home installation of the CDB.

- **migrationMethod**

  The method that will be used to migrate a database into the CDB. The valid values are:

- DATAPUMP (Oracle Data Pump Full Transportable Export and Import) : Uses Data Pump Full Transportable Export and Import to export data from a non-CDB and import into a newly created PDB. This option is supported for non-CDBs of version 11.2.0.3 or higher.

- PLUG_AS_PDB (Plug as PDB) : Uses the DBMS_PDB package to generate an XML metadata file. The XML metadata file describes the database files of the non-CDB that is used to plug it into a CDB. To use this option, the non-CDB must be an Oracle Database and the PDB must be a 12c database.

   Example case: Source non-CDB and destination PDB version 12.1

- **noncdbTargetName**

   Enterprise Manager target name of the non-CDB to be migrated..

- **noncdbTargetType**

   Target type of the non-CDB. "oracle_database" for single instance database and "rac_database" for a cluster database.

- **noncdbDBCreds**

   Named database credentials on the non-CDB of a user with sysdba privileges.

- **noncdbHostCreds**

   Named host credentials of the user who owns the Oracle Home installation of the non-CDB.

- **pdbName**

   The name of the Pluggable database that will be created on the CDB..

- **pdbAdminName**

   The username of the PDB administrator that will be created for the new PDB.

- **pdbAdminPassword**

   The password for the PDB administrator.

- **exportDir**

   The file system location on the non-CDB host where the exported datapump files (dump and data files) will be stored. This directory will be cleaned up after successful migration. Default export directory is the location pointed to by DATA_PUMP_DIR directory object on the non-CDB. Use a location on which non-CDB Oracle Home owner has read and write permissions.

- **importDir**

   File system location on the CDB host that will be used to temporarily stage the migration metadata and/or datafiles. This directory will be cleaned up after successful migration. Default import directory is the location pointed to by DATA_PUMP_DIR directory object on the CDB. Use a location on which CDB Oracle Home owner has read and write permissions. If '-locationSharedAcrossTargets' is specified, export directory will be used.

- **useOMF**

   Use OMF location as datafile location.

   Y: Supported only if the CDB uses OMF. Ignored otherwise.

   N (Default)

- **createAsClone**

This flag specifies whether the new pluggable database should be created as clone. Used only if the -migrationMethod is specified as 'PLUG_AS_PDB', ignored otherwise. The valid values are: Y | N (default).

- **dataFilesLoc**

  The file system location on the CDB host where the datafiles for the newly created PDB will be stored. If the CDB uses ASM then a disk group name can also be used as the datafile location. Default datafile location is the location pointed to by DATA_FILE_DIR directory object on the CDB. Use a location on which CDB Oracle Home owner has read and write permissions. Ignored if useOMF is Y.

- **encryptionPwd**

  Password to decrypt/encrypt datapump dump files. This is mandatory if the non-CDB contains encrypted tablespaces.

- **cdbWalletPwd**

  Wallet password to open the wallet on the CDB. This is mandatory if the non-CDB contains encrypted tablespaces.

- **objectExistsAction**

  Action to be taken when the exported object with same name is found on the newly created PDB.

  – SKIP: Default value

  – REPLACE

- **precheck**

  Option to run pre-requisite checks during the migration job.

  – YES (Default): Runs pre-requisite checks and proceeds to database migration if there are no errors during the pre-requisite checks.

  – NO: Proceeds to the database migration directly. Does not run the pre-requisite checks.

  – ONLY: Runs pre-requisite checks only. Does not migrate the database.

- **ignoreWarnings**

  Option to ignore the warnings if any, during pre-requisite checking and proceed with migration. Used only when precheck is set to YES, ignored otherwise. The valid values are:

  – YES (Default) : Ignores warnings and proceeds to migration.

  – NO : Does not proceed to migration if warnings are found.

- **locationSharedAcrossTargets**

  This parameter indicates that the exported location is shared across source and destination targets. This implies that import location is same as exported location in turn skipping the file transfer.

Examples

Example 1

The following example migrates the non-CDB target NON_CDB_1 as a PDB named NEW_PDB, using the datapump method, the non-CDB target credentials NON_CDB_DB_CREDS, and the non-CDB host credentials

NON_CDB_HOST_CREDS, specifying the administrator user name of the newly created PDB as pdbAdmin, and the administrator password as welcome:

```
emcli migrate_noncdb_to_pdb
  -migrationMethod=datapump
  -noncdbTargetName=NON_CDB_NAME
  -noncdbTargetType=oracle_database
  -noncdbHostCreds=NON_CDB_HOST_CREDS
  -noncdbDBCreds=NON_CDB_DB_CREDS
  -cdbTargetName=CDB_NAME
  -cdbTargetType=oracle_database
  -cdbHostCreds=CDB_HOST_CREDS
  -cdbDBCreds=CDB_DB_CREDS
  -pdbName=NEW_PDB
  -pdbAdminName=pdbAdmin
  -pdbAdminPassword=welcome
  -precheck=ONLY
  -ignoreWarnings
```

migrate_to_lifecycle_status

Migrates to the lifecycle state from the deployment type.

Format

```
emcli migrate_to_lifecycle_status
        -deployment_values="value1;value2;value3
        -lifecycle_stage_values="Stage;Stage;Production
```

Options

- **deployment_values**

  Deployment type values.

- **lifecycle_stage_values**

  Lifecycle stage values

modify_aggregate_service

Modifies an aggregate service instance.

Format

```
emcli modify_aggregate_service
      -name="name"
      -type="type"
      [-add_sub_services="name1:type1;name2:type2;..."]
      [-del_sub_services="name1:type1;name2:type2;..."]
      [-avail_eval_func="function_to_evaluate_availability."]
      [-timezone_region="timezone_region"]

[ ]  indicates that the parameter is optional
```

Options

- **name**

Aggregate service name.

- **type**

  Aggregate service type.

- **add_sub_services**

  Sub-services to be added.

- **del_sub_services**

  Sub-services to be deleted.

- **avail_eval_func**

  PL/SQL function to evaluate the availability of the aggregate service. Use [or|and] for the predefined evaluation helper function.

- **timezone_region**

  Time zone region of the service.

Examples

```
emcli modify_aggregate_service -name="My_Name"
     -type="aggregate_service"
     -add_sub_services="sub1:type1;sub2:type2"
     -del_sub_services="sub3:type3"
     -avail_eval_func="my_pkg.my_eval_func"
     -timezone_region="CST"
```

modify_collection_schedule

Modifies the collection schedule of a collection setup for metrics and policies for the specified set of targets. Combining all the metrics, running a script, and collecting the data is referred to as a collection. The collection has various attributes associated with it, such as the collection schedule, upload frequency, and so forth.

Format

```
emcli modify_collection_schedule
     -targetType=ttype
     -targetNames=tname1;tname2;tname3...
     -collectionName=collname
     [-collectionStatus=Enabled or Disabled]
     [-freqType={Minute}{Hour}{Day}{Week}{Weekly}{Month}
     [-freqValue={any integer value for Minute/Hour/Day/Week}{One or more from
      Mon...Sun for Weekly}{One or more from 1;2..31 or Last for Month}
     [-preview=Y or N]

[ ]  indicates that the parameter is optional
{ } indicates that you can select one of the s in the series shown

Note: All of the parameters and choices are case-insensitive
```

Options

- **targetType**

  You must specify a single target type value, and it should be the same as specified in the repository.

> **Note:**
>
> Only individual target types are currently supported.

- **targetNames**

  The target name should be the same as exists in the repository. All of the targets should be the same target type you specified in the targetType parameter. Use a semicolon (;) to separate the names. Changes to the collection schedule will be executed for only valid target name and target type combinations. For example:

  host1;host2;host3

- **collectionName**

  The collection name should be exactly the same as exists in the repository or the corresponding collections .xml file present on the Management Agent.

  Access files from the following locations to determine the collection to be modified. Select the desired collection and provide it as input to the EM CLI utility.

  - $AGENT_HOME/sysman/admin/metadata/<targetType>.xml

    This file is shipped as a part of the setup and contains information regarding the metrics for this target type.

  - $AGENT_HOME/sysman/admin/default_collection/ <targetType>.xml

    This file is shipped as a part of the setup and contains the collections shipped by default.

  - $AGENT_HOME/sysman/emd/collection/ <targetType_targetName>.xml

    Whenever changes have occurred for any particular target, this file is automatically generated. Collections for user-defined metrics are available in this file.

- **collectionStatus**

  Enables or disables the collection. The default is Enabled. If Disabled, freqType and freqValue are ignored.

- **freqType**

  You can specify one of the following values:

  Minute (default) Hour Day Week Weekly Month

  For Week, you must specify an integer value as the frequency value. For instance, if you specify freqType='WEEK' and freqValue='2', the collection occurs every two weeks.

  For Weekly, the possible values are Mon, Tue, Wed, Thu, Fri, Sat, Sun. For instance, if you specify freqType='Weekly' and freqValue='Tue;Thu;Sun', the collection occurs every Tuesday, Thursday, and Sunday of a week.

  The schedule is modified based on your selection. You do not need to specify a value (and the value will be ignored) if the collectionStatus parameter is set to Disabled.

  If you use this option, you must also use the freqValue parameter.

- **freqValue**

  You can specify one of the following values:

  - You must specify an integer value if the freqType is any one of Minute, Hour, Day, or Week. The default value is 5.

     – For Weekly, specify one or more choices from Mon, Tue, Wed, Thu, Fri, Sat, and Sun. If the collection occurs on any particular day(s) of the week, you must specify the corresponding value(s) against the Weekly option.

     – For Monthly, specify one or more choices from 1...31 or Last. If the collection occurs on any particular date(s) in a month, you must specify the corresponding value(s) against the Monthly option.

You do not need to specify a value (and the value will be ignored) if the collectionStatus parameter is set to Disabled.

If you use this option, you must also use the freqType parameter.

- **preview**

Provides a preview of the changes that would occur if this verb is executed. The default value for this option is Y (Yes), whether you specify the option or not. If you specify N, the changes to the collection schedule are executed for both the repository and Management Agent.

Examples

Example 1

This example changes the collection schedule to collect once every 5 minutes for hosts host1, host2, and host3. DiskActivity is a collection item associated with a host target type. The preview flag is set to Y, so the changes are not executed, but you can see the metrics affected if the changes were implemented.

```
emcli modify_collection_schedule -targetType="host"
      -targetNames="host1;host2;host3" -collectionName="DiskActivity"
      -freqType="Minute" -freqValue="5" -preview="Y"
```

Example 2

This example changes the collection schedule to collect once every 15 hours for host host1. Inventory is a collection item associated with a host target type. The preview flag is set to N, so the changes are executed for the associated metrics for both the repository and Management Agent.

```
emcli modify_collection_schedule -targetType="host"
      -targetNames="host1" -collectionName="Inventory"
      -freqType="Hour" -freqValue="15" -preview="N"
```

modify_group

Adds or removes targets from an existing group.

An error is not generated when attempting to delete a non-existent target in the group or when attempting to add a target that already exists in the group.

Standard Mode

```
emcli modify_group
      -name="name"
      [-type=<group>]
      [-add_targets="name1:type1;name2:type2;..."]...
      [-delete_targets="name1:type1;name2:type2;..."]...
      [-privilege_propagation=true|false]
      [-drop_existing_grants=yes|no]
```

```
[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
modify_group
      (name="name"
      [,type="<group>"]
      [,add_targets="name1:type1;name2:type2;..."]...
      [,delete_targets="name1:type1;name2:type2;..."]...
      [,privilege_propagation="true|false"]
      [,drop_existing_grants="yes|no"])
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Target name of the group to modify.

- **type**

  Group type: group. Defaults to `group`.

- **add_targets**

  Targets to add, each specified as `target_name:target_type`. You can specify this option more than once for Standard Mode.

- **delete_targets**

  Targets to delete, each specified as `target_name:target_type`. You can specify this option more than once for Standard Mode.

- **privilege_propagation**

  Enables or disables the privilege propagation flag for the group. Converts the normal group to a privilege propagating group and vice versa.

- **drop_existing_grants**

  Drops the existing grants on a group during privilege propagation conversion. This option is only applicable with the privilege_propagation parameter. The default value is yes.

Examples

These examples modify group `db2_group` by adding database `database:oracle_database` and deleting database `database2:oracle_database` from the group.

Example 1 - Command-Line

```
emcli modify_group
      -name=db2_group
      -add_targets=database:oracle_database
      -delete_targets=database2:oracle_database
```

Example 2 - Scripting and Interactive

```
modify_group
      (name="db2_group",
       add_targets="database:oracle_database",
       delete_targets="database2:oracle_database")
```

These examples modify group `my_hosts` by adding host `yourhost.example.com:host` to the group.

Example 3 - Command-Line

```
emcli modify_group
      -name=my_hosts
      -add_targets=yourhost.example.com:host
```

Example 4 - Scripting and Interactive

```
modify_group
      (name="my_hosts",
       add_targets="yourhost.example.com:host")
```

modify_incident_rule

Enables or disables a specific incident rule or rule set. (Updates all rules in the rule set.)

Format

```
emcli modify_incident_rule
      -action=enable|disable
      -type=ruleset|rule
      -rule_set_name=<name_of_rule_set>
      [-owner=<owner_of_rule_set>]
      [-rule_name=<name_of_rule>]

[ ]  indicates that the parameter is optional
```

Options

- **action**

  Action to be performed. Supported actions are enable and disable.

- **type**

  Disables a specific rule or the entire rule set.

- **rule_set_name**

  Name of the rule set to which you would like to apply the action.

- **owner**

  Owner of the rule set. If multiple rule sets exist with same name, the rule set owner is used to identify the rule set.

- **rule_name**

  Name of the specific rule to which the action will apply.

Examples

Example 1

This example enables 'rule set 1' and all child rules.

```
emcli modify_incident_rule -action='enable' -type='ruleset' -rule_set_name='rule
set 1'
```

Example 2

This example disables a single rule named 'rule 1' within 'rule set 1'.

```
emcli modify_incident_rule -action='disable' -type='rule' -rule_set_name='rule set 1' -
rule_name='rule 1'
```

modify_lifecycle_stage_name

Changes the life-cycle stage name. Only super users can run this command.

The Lifecycle Status property of the target has special semantics. The property does priority processing of events related to the target. Therefore, events from mission-critical targets have a higher priority than events from development targets. If you change the name, make sure to use a name that reflects its corresponding priority, because the same priority continues to be maintained regardless of the name change.

Format

```
emcli modify_lifecycle_stage_name
      -name="current_name"
      -new_name="new_name"
```

Options

- **name**

  Current life-cycle stage name. The available list in the order of decreasing priority is:

  – MissionCritical

  – Production

  – Stage

  – Test

  – Development

- **new_name**

  New life-cycle stage name. The new name is not translated into your locale and will be displayed as is. The new name should only contain alpha characters.

  When you change the existing name to a new name, all existing targets are updated with the new property value. For instance, if name=MissionCritical and new_name=Production, all existing targets are updated with Production.

Examples

```
emcli modify_lifecycle_stage_name
      -name="Test"
      -new_name="Test_staging"
```

modify_metric_data_load_limits

Modifies the metric data loading limits for a specified set of inputs.

This verb supports bulk operation for the following cases:

- All metrics of all targets

- All metrics of all targets for a specified target type

- All metrics of specified targets for a specified target type

- Specified metrics of all targets for a specified target type
- All metrics of targets with a specified property name and property values

Format

```
emcli modify_metric_data_load_limits
        [-num_rows="1000"]
        [-num_bytes="1024"]
        [-num_collections="10"]
        [-target_type="host"]
        [-prop_name="lifecycle"]
        [-prop_values="production;staging"]
        [-targets="host1;host2;"]
        [-metrics="Load;Filesystems;"]

[ ]  indicates that the parameter is optional.
```

Options

- -num_rows

  Limit set for the number of data rows. If you want to retain the existing value, then skip this option.

- -num_bytes

  Limit set for the number of bytes. If you want to retain the existing value, then skip this option.

- -num_collections

  Limit set for the number of collections. If you want to retain the existing value, then skip this option.

- -target_type

  Target types that you want to specify. If you want to perform the operation on all targets, then skip this option.

- -prop_name

  Global target property name.

- -prop_values

  Semicolon separated list of property values for a specified property name. Unless you want to set the data loading limits of targets based on target properties, skip this option.

- -targets

  Semicolon separated list of targets of a specified target type. You must use the -target_type option with this option. If you want to perform the operation on all targets for a specified target type, then skip this option.

- -metrics

  Semicolon separated list of metrics of a specified target type. You must use the -target_type option with this option. If you want to perform the operation on all metrics for a specified target type, then skip this option.

Examples

Example 1

The following command modifies the number of rows, number of bytes, and metric data loading limits for all metrics of "myhost1.oracle.com" and "myhost2.oracle.com" host targets.

```
emcli modify_metric_data_load_limits
       -num_rows="1000"
       -num_bytes="1024"
       -target_type="host"
       -prop_name="lifecycle"
       -prop_values="production;staging"
```

Example 2

The following command modifies the number of rows, number of bytes, and metric data loading limits for the Load and Filesystems metrics of all host targets.

```
emcli modify_metric_data_load_limits
       -num_rows="1000"
       -num_bytes="1024"
       -target_type="host"
       -metrics="Load;Filesystems;"
```

modify_metric_data_load_whitelist

Excludes or includes a specified set of metrics or targets from the metric data loading control mechanism.

This verb supports bulk operation for the following cases:

- All metrics of all targets for a specified target type

- All metrics of specified targets for a specified target type

- All metrics of targets with a specified property name and property values

- Specified metrics of all targets for a specified target type

Format

```
emcli modify_metric_data_load_whitelist
       -remove
       -target_type="host"
       -prop_name="lifecycle"
       [-prop_values="production;staging"]
       [-targets="myhost1.oracle.com;myhost2.oracle.com;"]
       [-metrics="Load;Filesystems;"]

[ ]  indicates that the parameter is optional.
```

Options

- -target_type

  Target type that you want to specify.

- -prop_name

  Global target property name.

- -prop_values

  Semicolon separated list of property values for a specified property name. Unless you want to add targets to a white list based on target properties, skip this option.

- -targets

Semicolon separated list of targets for a specified target type. If you want to perform the operation on all targets for a specified target type, then skip this option.

- -metrics

  Semicolon separated list of metrics for a specified target type. If you want to perform the operation on all metrics for a specified target type, then skip this option.

- -remove

  Use this option to remove the metric or target from the white listed targets or metrics. After removal, these metrics and targets are included in the metric data load quarantining process.

Examples

Example 1

The following command exempts host targets with the lifecycle status property set to production or staging from the data load quarantining process.

```
emcli modify_metric_data_load_whitelist
     -target_type="host"
     -prop_name="lifecycle"
     -prop_values="production;staging"
```

Example 2

The following command removes the myhost1.example.com and myhost2.example.com hosts from the white listed targets.

```
 emcli modify_metric_data_load_whitelist
         -remove
         -target_type="host"
         -targets="myhost1.example.com;myhost2.example.com;"
```

modify_proxy

Modifies a proxy which mediates the HTTP(S) traffic from the Oracle Management Server to the Management Agent.

> **Note:**
>
> This proxy is modeled as an 'oracle_em_proxy' target type.

Management Agents associated with the proxy can be modified in two ways:

- Using Management Agent names.

- Using patterns for Management Agent names.

Currently, 'oracle_em_proxy' target type proxies are supported only for traffic from the Oracle Management Server to the Management Agent. For traffic in the reverse direction (that is, from the Management Agent to the Oracle Management Server), proxy settings can be specified in the emd.properties file of the corresponding Management Agent.

Format

```
emcli modify_proxy
  -name="<name>"
   [-port=<port>]
   [-protocol=<http | https>]
   [-named_credential="<credential name>"]
   [
    [-agents="<name1>,<name2>,..."]
      |
      [
       [-add_agents="<name1>,<name2>,..."]
       [-delete_agents="<name1>,<name2>,..."]
      ]
   ]
   [
    [-agent_patterns="<name pattern1>,<name pattern2>,..."]
      |
      [
       [-add_agent_patterns="<name pattern1>,<name pattern2>,..."]
       [-delete_agent_patterns="<name pattern1>,<name pattern2>,..."]
      ]
   ]
   [
    [-excluded_agent_patterns="<name pattern1>,<name pattern2>,..."]

      |
      [
       [-add_excluded_agent_patterns="<name pattern1>,<name pattern2>,..."]
       [-delete_excluded_agent_patterns="<name pattern1>,<name pattern2>,..."]]
      ]
   ]
```

[ ] indicates that the parameter is optional.

Options

- -name

  Name that identifies a proxy.

- -port

  Port on the proxy host offering the proxy service.

- -protocol

  Protocol used by the traffic which the proxy mediates. Valid values are:

  – http

  – https

- -named_credential

  Name of the Named Credential to be used for authentication with the proxy.

- -agents

  Comma separated list of the names of the Management Agents which the proxy mediates for. You can use the backslash character (\) as an escape character. If the Management Agent with the specified name does not exist, then it is ignored.

- -add_agents

Comma separated list of the names of "additional" Management Agents which the proxy should mediate for. If the Management Agent with the specified name does not exist, then it is ignored. Note that you can use the backslash character (\) as an escape character.

- -delete_agents

  Comma separated list of the names of Management Agents which the proxy should not mediate for. This list, in addition to the list of Management Agents already associated with the proxy via their names, is applicable to the list of Management Agents specified in -add_agents option (if specified) also. Note that you can use the backslash character (\) as an escape character.

- -agent_patterns

  Comma separated list of patterns for the names of Management Agents which the proxy mediates for. The pattern can use two wildcard characters:

  – Asterisk character (*) for one or more characters

  – Question mark character (?) for a single character.

  – Backslash character (\) as an escape character.

- -add_agent_patterns

  Comma separated list of "additional" patterns for the names of Management Agents which the proxy should mediate for. These patterns also use the same wildcard characters as applicable for the -agent_patterns option.

- -delete_agent_patterns

  Comma separated list of patterns which should be removed from the list of patterns for the names of Management Agents which the proxy should mediate for. This list, in addition to the list of Management Agents name patterns already associated with the proxy, is applicable to the list of patterns specified in -add_agent_patterns option (if specified).

- -excluded_agent_patterns

  Comma separated list of patterns for the names of Management Agents which must be excluded from the names of the Management Agents identified by Management Agent name patterns associated with the proxy. These patterns can also use same wildcard characters as applicable for the -agent_patterns option.

- -delete_excluded_agent_patterns

  Comma separated list of patterns which should be removed from the list of patterns for the names of Management Agents which must be excluded from the names of Management Agents identified by the Management Agent name patterns associated with the proxy. This list, in addition to the list of excluded Management Agents name patterns already associated with the proxy, is applicable to the list specified in the -add_excluded_agent_patterns option (if specified) as well.

Examples

Example 1

The following command modifies the proxy identified by the " us-proxy-1" name to allow HTTPS traffic only.

```
emcli modify_proxy
        -name="us-proxy-1"
        -protocol=https
```

Example 2

The following command modifies the proxy identified by the "us-proxy-1" name to be associated with the "myhost001.us.example.com:3535 and myhost002.us.example.com:3535" Management Agents. If the proxy was associated with any other Management Agents using patterns previously, those patterns and the list of Management Agents they derive remains the same.

```
emcli modify_proxy
      -name="us-proxy-1"
      -format="name:script;column_separator:|"
      -agents="myhost001.us.example.com:3535,myhost002.us.example.com:3535
```

modify_monitoring_agent

Changes the Agents configured to monitor targets in a WebLogic Domain.

Format

```
emcli modify_monitoring_agent        -target_name=<target_name>        [-
target_type=weblogic_domain]        [-assign_local_agent]        -debug
[ ]  indicates that the parameter is optional
```

Options

*   **target_name**

    Complete target name of domain to be modified.

*   **target_type**

    Default value is weblogic_domain, and is the only valid target type.

*   **assign_local_agent**

    Globally assigns each target in the WebLogic Domain, such as WebLogic Server, to be monitored by the Agent installed on each target's host. That is, after running the verb with this option, each target in the domain is monitored by its local Agent. The local Agent is assigned if a local Agent is found. Otherwise, the monitoring Agent of the target is not changed.

*   **debug**

    Runs the verb in verbose mode for debugging purposes.

Examples

This example changes the Agents configured to monitor targets in a WebLogic Domain.

```
emcli modify_monitoring_agent
      -target_name=/prod_my_domain/my_domain
      -assign_local_agent
```

modify_named_credential

Updates an existing named credential. You can provide input parameters using command line arguments or an input properties file. It also supports the input_file tag for passwords and parameter values.

Format

```
 emcli modify_named_credential          -cred_name=<name>          -
new_cred_name<name>          -cred_type=<credential_type>          -
cred_scope=<credential_scope>          -cred_desc=<credential_description>          -
target_name=<target_name>          -target_type=<target_type>
        -test
        -test_target_name=<test_target_name>
        -test_target_type=<test_target_type>
        -input_file=<tag|value>          -properties_file=<filename>          -
attributes=<p1:v1;p2:v2;...>
        -remove_old_attributes
```

Options

- **cred_name**

  Credential name, such as MyBackUpCreds. This is required if you do not use the properties_file option.

- **new_cred_name**

  New credential name.

- **cred_type**

  Credential type.

- **cred_scope**

  Possible values are global instance. The default is global.

- **cred_desc**

  Credential description.

- **target_name**

  This is required when cred_scope is instance.

- **target_type**

  This is required when cred_scope is instance.

- **test**

  Use this option to test the credential before saving.

- **test_target_name**

  Use this option to supply the target name to test a global credential. This is mandatory when the scope is global and the test option is used.

- **test_target_type**

  Use this option to supply the target type to test a global credential. This is mandatory when the scope is global and the test option is used.

- **input_file**

  Use this option to supply sensitive property values from the file.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **properties_file**

Use this option to pass all parameters from the file. Values given on the command line take precedence.

- **attributes**

  Specify credential columns as follows:

  ```
  colname:colvalue;colname:colvalue
  ```

  You can change the separator value using -separator=attributes=<newvalue>, and you can change the subseperator value using -subseparator=attributes=<newvalue>.

- **remove_old_attributes**

  Unsets all existing credential column values.

Examples

Example 1

This example updates credentials to foo and bar:

```
emcli modify_named_credential
      -cred_name=NC1
      -attributes="HostUserName:foo;HostPassword:bar"
```

Example 2

This example reads the password from the mypasswordfile.txt file.

```
emcli modify_named_credential
      -cred_name=NC1
      -attributes="HostUserName:foo;HostPassword:tag"
      -input_file="tag:mypasswordfile.txt"
```

modify_red_group

Adds or removes targets from an existing redundancy group. An error is not generated when attempting to delete a non-existent target in the redundancy group.

Format

```
emcli modify_red_group
      -name="name"
      -type=<generic_redundancy_group>
      [-add_targets="name1:type1;name2:type2;..."]...
      [-delete_targets="name1:type1;name2:type2;..."]...
      [-owner=<redundancy_group_owner>]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Target name of the group to modify.

- **type**

  Redundancy Group type: generic_redundancy_group. Defaults to `generic_redundancy_group`.

- **add_targets**

Targets to add, each specified as `target_name:target_type`. You can specify this option more than once.

- **delete_targets**

  Targets to delete, each specified as `target_name:target_type`. You can specify this option more than once.

- **owner**

  Owner of the redundancy group.

Examples

This example modifies redundancy group servers by adding `Server1:generic_apache` and deleting `Server5:generic_apache` from the redundancy group.

```
emcli modify_red_group -name=Servers
     -add_targets=HTTP_Server1:generic_apache
     -delete_targets=Server5:generic_apache
```

modify_redundancy_group

Modifies a redundancy group.

Format

```
emcli modify_redundancy_group
       -redundancyGroupName="redGrpName"
       [-owner="new_owner"]
       [-memberTargetType="tType"]
       [-add_targets="tName1;tName2"]
       [-delete_targets="tName3;tName4"]
       [-group_status_criterion="NUMBER" or "PERCENTAGE"]         [-
group_status_tracked="UP" or "DOWN"]         [-group_status_value=<status_value>]
       [-privilege_propagation=true|false]        [-drop_existing_grants=yes|no]

[ ]  indicates that the parameter is optional
```

Options

- **redundancyGroupName**

  Name of the redundancy group.

- **owner**

  Valid owner to be specified.

- **memberTargetType**

  Target type of the constituent member targets. You need to specify this option if you specify either add_targets or delete_targets.

- **add_targets**

  Member targets to be added to this redundancy group.

- **delete_targets**

  Member targets to be deleted from this redundancy group.

- **group_status_criterion**

This option and the next two calculate the status of the Redundancy Group. Consequently, you need to specify all three options together. If this is not to be a capacity group, you need to specify the following combination:

```
-group_status_criterion='NUMBER' -group_status_tracked='UP'
-group_status_value='1']
```

- **group_status_tracked**

  See the option above.

- **group_status_value**

  See the group_status_criterion .

  You can specify any value between 1 and 100 if -group_status_criterion= "PERCENTAGE", or any value between 1 and the number of targets present if -group_status_criterion="NUMBER".

- **privilege_propagation**

  Enables or disables the privilege propagation flag for the group. Converts the normal group to a privilege-propagating group and vice versa.

- **drop_existing_grants**

  Drops the existing grants on a group during privilege propagation conversion. this option is only applicable with the privilege_propagation parameter. The default value is yes.

Examples

This example changes the configuration of the 'redGrp1' redundancy group to add listener, listener2, and listener3 to its existing members, and delete listener4 and listener5 from its existing members.

```
emcli modify_redundancy_group -redundancyGroupName='redGrp1'        -
memberTargetType='oracle_listener'        -
add_targets='listener;listener2;listener3'        -
delete_targets='listener4;listener5'        -group_status_criterion='NUMBER'       -
group_status_tracked='UP'        -group_status_value='2'
```

modify_resolution_state

Modifies an existing resolution state that describes the state of incidents or problems. Only super administrators can execute this command. You need to specify the updated label as well as the updated position. The position can be between 2 and 98, and cannot be in use by another resolution state.

You can also optionally indicate that the state should apply to both incidents and problems. A success message is reported if the command is successful. An error message is reported if the change fails.

Format

```
emcli modify_resolution_state
       -label="old_label_of_state"
       -new_label="new_label_for_display"
       -position="new_display_position"
       [-applies_to=BOTH]

[ ]  indicates that the parameter is optional
```

Options

- **label**

  Old label of the state to be modified.

- **new_label**

  End-user visible label of the state. The label cannot exceed 32 characters.

- **position**

  Position of this state within the overall list of states. This is used when displaying the list of states in the user interface. The position can be between 2 and 98.

  It is recommended that you set the position with sufficient gaps to facilitate moving states around. For example, if you set the positions to 5, 10, and 15 instead of 2, 3, and 4, it is easier to move a state from position 15 to 9, for instance, in contrast to the latter scheme, in which you would have to move all states to provide space for the reordering.

- **applies_to**

  Indicates that the state is applicable for incidents and problems. The only supported value is "BOTH."

Examples

Example 1

This example updates the resolution state with the old label "Waiting for TT" with the new label "Waiting for Ticket," and if necessary, changes the position to 25.

```
emcli modify_resolution_state  -label="Waiting for TT" -new_label="Waiting for
Ticket" -position=25
```

Example 2

This example updates the resolution state with the old label "SR Waiting" with the new label "Waiting for SR," and if necessary, changes the position to 35. It also makes the state applicable to incidents and problems.

```
emcli modify_resolution_state  -label="SR Waiting" -new_label="Waiting for SR" -
position=35 -applies_to=BOTH
```

modify_role

Modifies an existing Enterprise Manager administrator role.

> **Note:**
>
> To leave a current value unchanged, do not include the corresponding argument. For example, if you are not modifying the role's description, do not include `-description`.
>
> To update a role and add targets to the role, use the grant_privs verb.

Format

```
emcli modify_role
     -name="role_name"
     [-description="description"]
     [-roles="role1;role2;..."]
     [-privilege="name[;secure-resource-details]]"
     [-separator=privilege="sep_string"]
     [-subseparator=privilege="subsep_string"]
     [-users="user1;user2;..."]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  The name of the role to modify.

- **description**

  The updated description of the role.

- **roles**

  A list of roles to assign to the specified role. Currently, "PUBLIC" is the only built-in role.

- **privilege**

  A privilege to grant to this role. This option may be specified more than once. Note that privilege names are case-insensitive. Specify <secure_resource_details> as follows:

  ```
  resource_guid|
  [resource_column_name1=resource_column_value1[:resource_column_name2=resource_colum
  n_value2]..]"
  ```

  Note: Privileges are case-insensitive.

  To retrieve the list SYSTEM privileges, which do not require resource information, execute the following emcli command:

  ```
  emcli get_supported_privileges -type=SYSTEM
  ```

  To retrieve the complete list of privileges and resource column names, execute the following emcli command:

  ```
  emcli get_supported_privileges
  ```

  To retrieve the list of target type privileges, execute the following emcli command

  ```
  emcli get_supported_privileges -type=TARGET
  ```

  To get the list of job privileges, execute the following emcli command

  ```
  emcli get_supported_privileges -type=JOB
  ```

- **separator**

  Specify a string delimiter to use between name-value pairs for the value of the -privilege option. The default separator delimiter is a semi-colon (;).

- **subseparator**

  Specify a string delimiter to use between name and value in each name-value pair for the value of the -privilege option. The default subseparator delimiter is a colon (:).

- **users**

A list of users to whom this role is assigned. <subseparator:>WITH_ADMIN should be suffixed if the role needs to granted by WITH_ADMIN option.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

This example modifies a role named `existing_role` with the one-sentence description `"This role was changed."` The role combines three existing roles: `role1`, `role2`, and `role3`. The role also has two added privileges: to view the job with ID `923470234ABCDFE23018494753091111` and to view the target `host1.example.com:host`. The role is granted to `johndoe` and `janedoe`.

```
emcli modify_role
      -name="existing_role"
      -desc="This role was changed"
      -roles="role1;role2;role3"
      -privilege="view_job;923470234ABCDFE23018494753091111"
      -privilege="view_target;host1.example.com:host"
      -users="johndoe;janedoe"
```

Example 2

This example modifies a role named `existing_role` by assigning `role4`, `role5`, and `role6` to it. The description, privileges, and users associated with this role remain unchanged.

```
emcli modify_role
      -name="existing_role"
      -roles="role4;role5;role6"
```

modify_system

Adds or removes targets from an existing system. An error is not generated when attempting to delete a non-existent target in the system or when attempting to add a target that already exists in the system.

If you specify both the `-add_members` and `-delete_members` options in the same command, the members specified by `-delete_members` are deleted first, then the members specified by `-add_members` are added.

Format

```
emcli modify_system
        -name="name"
        [-type=<generic_system>]
        [-add_members="name1:type1:key_member|
non_key_member;name2:type2;..."]...
            [-separator=add_members="sep_value"]
            [-subseparator=add_members="subsep_value"]
        [-delete_members="name1:type1;name2:type2;..."]...
            [-separator=delete_members="sep_value"]
            [-subseparator=delete_members="subsep_value"]
        [-owner="new_owner"]
        [-privilege_propagation=true|false]
```

```
[-drop_existing_grants=yes|no]
[-availability_type="ALL/ANY"]
```

`[ ]` indicates that the parameter is optional

Options

- **name**

  Target name of the system to modify.

- **type**

  System type: generic_system. Defaults to `generic_system`.

- **add_members**

  Targets to add, each specified as target_name:target_type. You can specify this more than once. key_member specifies that this target is a part of the systems availability calculation. non_key_member specifies that this target is not a part of the systems availability calculation.

- **delete_members**

  Member targets to be removed from the system, each specified as target_name:target_type. You can specify this option more than once.

- **owner**

  New owner of the system.

- **privilege_propagation**

  Enables or disables the privilege propagation flag for the group. Converts the normal group to a privilege propagating group and vice versa.

- **drop_existing_grants**

  Drops existing grants on a group when conversion occurs in privilege propagation nature. This option is only applicable with the privilege_propagation parameter. The default value is yes.

- **availability_type**

  Availability calculation method of the system. Defining this is required if key_member is defined. ALL denotes that all key members must be up in order to establish the system as UP. ANY denotes that at least one of the key members must be up in order to establish the system as UP.

Examples

Example 1

This example modifies system `my_system` by adding targets `system_a:generic_system` and `database:oracle_database`, and deleting the nonexistent target `nosystem:generic_system` from the system.

```
emcli modify_system -name=my_system
    -add_members=system_a:generic_system
    -add_members=database:oracle_database
    -delete_members=nosystem:generic_system
```

Example 2

This example modifies system db2_system by adding database database1 as a key member, adding databases database2 and database3 as non-key members, and deleting database4 and database5. The availability computation is impacted, since database1 is now part of the availability computation for the db2_system. If database4 and database5 were key members, they are no longer part of the availability computation for the db2_system.

Specifying separator and subseparator is optional. Separator defaults to ; and subseparator defaults to : .

```
emcli modify_system -name=db2_system -type=generic_system
  [add_members=database1:oracle_database:key_member,database2:oracle_database]
    [separator=add_members=","]
    [subseparator=add_members=":"]
  [add_members=database3:oracle_database:non_key_member]
  [delete_members=database4:oracle_database,database5:oracle_database]
    [separator=delete_members=","]
    [subsrparator=delete_members=":]"
```

modify_target

Modifies a target instance definition.

> **Note:**
>
> To change the monitoring password of a database target, either use update_db_password (at the RAC level), or use modify_target with the following options:
>
> ```
> -credentials="UserName:newuser;password:PWD_FILE;Role:SYSDBA"
> -input_file="PWD_FILE:at_pwd_file"
> ```

Format

```
emcli modify_target
      -name="name"
      -type="type"
      [-properties="pname1:pval1;pname2:pval2;..."]...
      [-separator=properties="sep_string"]
      [-subseparator=properties="subsep_string"]
      [-credentials="userpropname:username;pwdpropname:password;..."]
      [-input_file="parameter_tag:file_path"]
      [-display_name="display name"]
      [-on_agent]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Target name.

- **type**

  Target type.

- **properties**

Name-value pair list of properties for the target instance. The "name"(s) are identified in the target-type metadata definition. They must appear exactly as they are defined in that file. Metadata files are located in `$AGENT_ORACLE_HOME/sysman/admin/metadata`.

> **✎ Note:**
>
> This verb does not support setting global target properties. It is recommended that you use set_target_property_values to set target properties.

- **separator=properties**

  Specifies a string delimiter to use between name-value pairs for the value of the `-properties` option. The default separator delimiter is ";".

- **subseparator=properties**

  Specifies a string delimiter to use between name and value in each name-value pair for the value of the `-properties` option. The default subseparator delimiter is ":".

- **credentials**

  Monitoring credentials (name-value pairs) for the target instance. The "name"(s) are identified in the target-type metadata definition as credential properties. They must appear exactly as they are defined in that file. Metadata files are located in `$AGENT_ORACLE_HOME/sysman/admin/metadata`.

- **input_file**

  Used in conjunction with the `-credentials` option, this option enables you to store specific target monitoring credential values, such as passwords, in a separate file. The `-input_file` option specifies a mapping between a tag and a local file path. The tag is specified in lieu of specific monitoring credentials of the `-credentials` option. The tag must not contain colons (:) or semi-colons (;).

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **display_name**

  Sets the target display name.

- **on_agent**

  Propagates changes to the Management Agent collecting this target's metrics.

Examples

Example 1

This example modifies the display name and properties for the `oracle_database` target with the name `database`. The `on_agent` flag ensures that the changes are propagated to the Management Agent collecting for this target.

```
emcli modify_target
     -name="database"
     -type="oracle_database"
     -display_name="New Name DB"
     -properties="SID=newsid|Port=15091|OracleHome=/oracle"
     -properties="MachineName=smpamp-sun1.example.com"
     -separator=properties="|"
```

```
-subseparator=properties="="
-on_agent
```

Example 2

This example modifies an `oracle_database` target type with the name `payroll_db`. In this example, the display name for this database (target name that is displayed in the Enterprise Manager UI) is being changed to `payroll`. The port number is being changed to `15067`, and the Oracle Home is being changed to `/oradb`. The administrator (`dbsnmp`), whose previous default role was `normal`, is being changed to `sysdba`. This example also illustrates the use of the `input_file` to camouflage the credentials. The password is actually in a file named `at_pwd_file`. The `-input_file` argument replaces `PWD_FILE` with the contents of `at_pwd_file` in the `-credentials` option.

```
emcli modify_target
      -name="payroll_db"
      -type="oracle_database"
      -credentials="UserName:Fred;password:PWD_FILE;Role:sysdba"
      -properties="Port:15067;OracleHome:/oradb"
      -input_file="PWD_FILE:at_pwd_file"
      -display_name=payroll
      -on_agent
```

Example 3

This example modifies an existing Apache Tomcat target named TARGET_NAME. Here, the SSL Trust Store location for the target is changed.

```
emcli modify_target
        -name="TARGET_NAME"
        -type="tomcat"
        -properties="SSLTrustStore:AGENT_HOME/agent_inst/sysman/config/montrust/
AgentTrust.jks"
        -separator=properties=";"
        -subseparator=properties=":"
        -on_agent
```

Example 4

This example modifies an existing Apache Tomcat target named TARGET_NAME. Here, the SSL Trust Store location and the SSL Trust Store password for the target are changed.

```
emcli modify_target
        -name="TARGET_NAME"
        -type="tomcat"
        -properties="SSLTrustStore:AGENT_HOME/agent_inst/sysman/config/montrust/
AgentTrust.jks;SSLTrustStorePassword:welcome"
        -separator=properties=";"
        -subseparator=properties=":"
        -on_agent
```

Bug Fix 1

This example updates the `ORACLE_HOME` property in RAC and Oracle Database targets. A total of four statements are required in order to update RAC targets. This command must be run in Oracle Databases within the RAC target.

```
emcli
modify_target(name="<TARGET_NAME>",type="rac_database",properties="OracleHome:/opt/
oracle/product/11.2.0/racdb11204")

modify_target(name="<TARGET_NAME>",type="oracle_database",properties="OracleHome:/opt/
oracle/product/11.2.0/racdb11204",on_agent=True)

modify_target(name="<TARGET_NAME>",type="oracle_database",properties="OracleHome:/opt/
oracle/product/11.2.0/racdb11204",on_agent=True)

modify_target(name="<TARGET_NAME>",type="oracle_database",properties="OracleHome:/opt/
oracle/product/11.2.0/racdb11204",on_agent=True)
```

modify_threshold

Edits threshold settings for a given target and metric

Format

```
emcli modify_threshold
        -target_name="tname"
        -target_type="ttype"
        [-metric="met"]
        [-column="col"]
        [-key_columns="val1;val2;..."]
        [-warning_threshold="warn"]
        [-critical_threshold="crit"]
        [-occurrences="occur"]
        [-prevent_override="0 or 1"]
        [-force]
        [-input_file="FILE:cli_input.txt"]

[ ]  indicates that the parameter is optional
```

Options

- **target_name**

  Name of the target associated with the threshold.

- **target_type**

  Type of target associated with the threshold.

- **metric**

  Metric category associated with the threshold.

- **column**

  Metric column associated with the threshold.

- **key_columns**

  Values of the key columns associated with the threshold. If you do not specify this option for a key-based metric, an EM CLI occurs.

- **warning_threshold**

  New warning threshold value. Specify " " for no warning threshold. If warning and critical thresholds are incoherent depending on the comparison operator, an EM CLI error occurs .Use -force to save the provided thresholds.To keep the previous value (if any), omit this option.

- **critical_threshold**

  New critical threshold value. Specify " " for no warning threshold. If warning and critical thresholds are incoherent depending on the comparison operator, an EM CLI error occurs .Use -force to save the provided thresholds.To keep the previous value (if any), omit this option.

- **occurrences**

  Number of times a threshold can be violated before causing an alert. To keep the previous value (if any), omit this option.

- **prevent_override**

  Prevents thresholds modification of this metric from future Apply Template operations on this target. Periodic Apply Template operations are submitted on targets managed by Administration Groups, which can override the metric thresholds you set if the prevent_override flag is not set.

  An error occurs if prevent_override is not set in database, you have not provided prevent_override, and the target is managed by Administration Groups. To continue without using prevent_override, use -force.To keep the previous value (if any), omit this option.

- **force**

  Saves the provided thresholds incase recommended in previous error messages.

- **input_file**

  Provides threshold details for multiple metrics in a text file.Do not provide metric, column, key_columns, warning_threshold, critical_threshold, occurrences and prevent_override in this command when using the input_file option.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

  You can provide the details for multiple metrics in the input file as shown:

  ```
  START_RECORD 1
  metric , Filesystems
  column , available
  key_columns , ab;cd;
  warning_threshold , 15
  critical_threshold , 50
  occurrences , 3
  prevent_override , 1
  END_RECORD 1

  START_RECORD 2
  metric , Load
  column , cpuUtil
  warning_threshold , 15
  critical_threshold , 50
  occurrences , 3
  prevent_override , 1
  END_RECORD 2
  ```

  To set the thresholds for the "Al Others" key, provide the details as shown:

  ```
  START_RECORD 1
  metric , Filesystems
  column , available
  ```

```
             key_columns , ;
             warning_threshold , 15
             critical_threshold , 50
             occurrences , 1
             END_RECORD 1
```

Examples

Example 1

This example sets the critical threshold value to "0" for the Load metric, and the cpuUtil column on the host "myhost.example.com". The warning threshold value and response action (if any) remain unchanged.

```
emcli modify_threshold
        -target_name="myhost.example.com"
        -target_type="host"
        -metric="Load"
        -column="cpuUtil"
        -critical_threshold="0"
        -prevent_override="0"
        -force
```

Example 2

This example sets the DiskActivitybusy threshold for the DiskActivitydevice called sd0 on the host myhost.example.com.

```
emcli modify_threshold
        -target_name="myhost.example.com"
        -target_type="host"
        -metric="DiskActivity"
        -column="DiskActivitybusy"
        -key_columns="sd0;"
        -warning_threshold="55"
        -critical_threshold="65"
        -occurrences="3"
```

modify_user

Modifies an existing Enterprise Manager administrator.

Format

Standard Mode

```
emcli modify_user
      -name="name"
[-password="password"]
[-type="type of user"]
[-roles="role1;role2;..."
[-email="email1;email2;..."]
[-privilege="name[;secure-resource-details]]"]
[-separator="privilege="sep_string""]
[-subseparator="privilege="subsep_string""]
[-profile="profile_name"]
[-desc="user_description"]
[-expired="true/false"]
[-prevent_change_password="true/false"]
[-department="department_name"]
[-cost_center="cost_center"]
```

```
[-line_of_business="line_of_business"]
[-contact="contact"]
[-location="location"]
```

Interactive or Script Mode

```
modify_user(
        name="name"
[,password="password"]
[,type="type of user"]
[,roles="role1;role2;..."]
[,email="email1;email2;..."]
[,privilege="name[;secure-resource-details]]"]
[,separator="privilege="sep_string""]
[,subseparator="privilege="subsep_string""]
[,profile="profile_name"]
[,desc="user_description"]
[,expired="true/false"]
[,prevent_change_password="true/false"]
[,department="department_name"]
[,cost_center="cost_center"]
[,line_of_business="line_of_business"]
[,contact="contact"]
[,location="location"]
)
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **name**

    Administrator name.

*   **password**

    Replaces the administrator password with the specified password.

*   **type**

    Converts to the specified type of user. Possible values for this parameter are EM_USER, EXTERNAL_USER, and DB_EXTERNAL_USER. The Default value of this parameter is EM_USER.

*   **roles**

    Replace current roles with the specified list of Enterprise Manager roles to grant to this administrator. Currently, the built-in roles include PUBLIC.

*   **email**

    Replaces current email addresses for this administrator with the specified list. To delete all email addresses for this administrator, specify an empty string.

*   **privilege**

    Privilege to grant to this administrator. You can specify this option more than once. Specify <secure_resource_details> as:

    ```
    resource_guid|
    [resource_column_name1=resource_column_value1[:resource_column_name2=resource
    _column_value2]..]"
    ```

To retrieve the list of SYSTEM privileges, which do not require resource information, execute the following emcli command:

```
emcli get_supported_privileges -type=SYSTEM
```

To retrieve the complete list of privileges and resource column names, execute the following command:

```
emcli get_supported_privileges
```

To retrieve the list of TARGET privileges, execute the following emcli command:

```
emcli retrieve -type=TARGET
```

To retrieve the list of job privileges, execute the following emcli command:

```
emcli get_supported_privileges -type=JOB
```

- **separator**

  Specify a string delimiter to use between name-value pairs for the value of the -privilege option. The default separator delimiter is ";".

- **subseparator**

  Specify a string delimiter to use between name and value in each name-value pair for the value of the -privilege option. The default subseparator delimiter is ":".

- **profile**

  Database profile name. It uses DEFAULT as the default profile name.

- **desc**

  User description for the user being modified.

- **expired**

  Valid values are true or false. Use this option to expire the password immediately. The default is false.

- **prevent_change_password**

  Valid values are true or false. When set to true, you cannot change your own password. The default is false.

- **department**

  Name of the department of the administrator.

- **cost_center**

  Cost center of the administrator in the organization.

- **line_of_business**

  Line of business of the administrator.

- **contact**

  Contact information for the administrator.

- **location**

  Location of the administrator.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

This example modifies the `new_admin` administrator. The user will have two privileges: to view the job with ID `923470234ABCDFE230184947530911111` and to view the target `host1.example.com:host`. The user will also be granted role `PUBLIC`. The user email addresses will be set to `first.last@example.com` and `joe.shmoe@shmoeshop.com`.

```
emcli modify_user
      -name="new_admin"
      -password="oracle"
      -email="first.last@example.com;joe.shmoe@shmoeshop.com"
      -roles="public"
      -privilege="view_job;923470234ABCDFE230184947530911111"
      -privilege="view_target;host1.example.com:host"
```

Example 2

This example deletes all the email addresses and privileges for administrator `new_admin`. Note that `-privilege=""` and `-privilege` are equivalent if specified at the command line in a UNIX shell.

```
emcli modify_user
      -name="new_admin"
      -email="
      -privilege=""
```

modify_user_profile

Modifies the user profile.

Format

Standard Mode

```
emcli modify_user_profile
      -name="profile name"
      [-description="profile desc"]
      [-users="users to be associated"]
      [-included_profiles="profile to be included"]
```

Interactive or Script Mode

```
emcli modify_user_profile(
      name="profile name"
      [,description="profile desc"]
      [,users="users to be associated"]
      [,included_profiles="profile to be included"]      )
```

Options

* name

  The name of the user profile to be modified.

* description

  Description of the user profile.

* users

The users to be associated with the user profile.

* included_profiles

  Profiles to be included in the profile

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

This example modifies the `profile1` user profile.

```
emcli modify_user_profile
    -name=profile1
    [-description]=test profile
    [-users]=user1;user2[-included_profiles=profile3;profile4]
```

modify_virtual_platform

Modifies the Oracle Virtual Platform target's monitoring agent, fail-over agent, or the monitoring credentials. Only the properties of the target needing modification must be specified when modifying a target of that type. For all of the parameters not passed, the existing values are retained.

Format

```
emcli modify_virtual_platform
    -name="target_name"
    -agent="agent_target_name"
    [-failover_agent="failover_agent_target_name"]
    -credentials="property_name1:property_value1;property_name2:
        property_value2;..."
        [-wait_for_completion=true|false]
        [-wait_for_completion_timeout=<time_in_minutes>]
    [-separator=credentials="separator_for_key_value_pairs"]
        [-subseparator=credentials="separator_for_key_value_pair"]
    [-input_file="FILE:file_path"]
```

```
[ ]  indicates that the parameter is optional
```

Options

* **name**

  Target name of the Oracle Virtual Platform to modify.

* **agent**

  Target name of the primary agent used to monitor the Oracle Virtual Platform and related targets.

* **failover_agent**

  Target name of the failover agent used to monitor the Oracle Virtual Platform and related targets.

* **credentials**

Monitoring credentials (name-value pairs) for the target instance. The "names" are defined in the target type metadata definition as credential properties. Metadata files are located at $AGENT_HOME/sysman/admin/metadata.

See the examples for details on various options.

- **wait_for_completion**

  Flag to indicate if the CLI is going to wait for the submitted job to finish. The default value is false. If the value is true, the progress of the job is printed on the command line as and when the addition of Oracle Virtual Platform(s) Succeeds/ Fails.

- **wait_for_completion**

  Flag to indicate if the CLI is going to wait for the submitted job to finish. The default value is false. If the value is true, the CLI waits and prints the job output on the command line when the modification of Oracle Virtual Platform(s) Succeeds/Fails.

- **wait_for_completion_timeout**

  Time in minutes after which CLI stops waiting for the job to finish. This option is honored only if the value for parameter wait_for_completion is true. A negative or zero value does not wait for the job to finish.

  See the examples for details.

- separator=credentials

  Custom separator for the credential key value pairs. Specify a string delimiter to use between name-value pairs for the values of the -credentials option. The default separator delimiter is ";".

  For more information about the separator parameter, see -input_file Syntax Guidelines.

- subseparator=credentials

  Custom separator for a key value pair. Specify a string delimiter to use between name and value in each name-value pair for the values of the -credentials option. The default subseparator delimiter is ":".

  For more information about the subseparator parameters see -input_file Syntax Guidelines.

- **input_file**

  File path with a credential secret value. Optionally use in conjunction with the -credentials option. You can use this option to set specific target monitoring credential values, such as passwords or SSH keys, in a separate file.

  This option specifies a mapping between a tag and a local file path. The tag is specified in lieu of specific -credentials property values.

Examples

Example 1

This example modifies the Oracle Virtual Platform's credentials with root user host credentials. The value of the property "OVSUsername" is used for the user name and "OVSPassword" for the password. The value of the property "privilegedUser" indicates if the virtualization-specific metrics are collected (true) or not (false) when monitoring. The password is passed at the command line.

```
emcli modify_virtual_platform
     -credentials='type:DMOvsBasicCreds;PrivilegeType:none;
             privilegedUser:true;OVSUsername:root;OVSPassword:mypassword'
```

Example 2

This example modifies the Oracle Virtual Platform's credentials with root user host credentials. The value of the property "OVSUsername" is used for the user name and "OVSPassword" for the password. The value of the property "privilegedUser" indicates if the virtualization-specific metrics are collected (true) or not (false) when monitoring. The password of the root user is read from the input file "password.txt".

```
emcli modify_virtual_platform
     -name=exampletarget
     -credentials='type:DMOvsBasicCreds;PrivilegeType:none;
         privilegedUser:true;OVSUsername:root;OVSPassword:PWD_FILE'
     -input_file='PWD_FILE:password.txt
```

package_fa_problem

This verb accomplishes the following tasks:

- Packages a Fusion Applications problem by reading details from a pre-written input file.

- Optionally attaches metrics, custom dumps, and reports by reading details from pre-written heap dumps and database AWR (Automatic Workload Repository) files.

- Uploads the finalized package to Oracle Support and reports the number of the draft Service Request created for the package if no SR is supplied.

Format

```
emcli package_fa_problem
     -input_file=incident_packaging_file:file_path
     [-input_file=heap_dumps_file:file_path]
     [-input_file=db_awr_file:file_path]

[ ]  indicates that the parameter is optional
```

Options

- **input_file=incident_packaging_file**

  Fully-qualified path to a CSV formatted file containing one line of details for the Fusion Applications problem to be packaged.

  The structure of the CSV file is as follows:

```
<Full target name>,
<Target type>,
<Problem key>,
<Host credential name - for using named credentials only>,
<Host username - for using new credentials only>,
<Host password - for using new credentials only>,
<Target credential name - for using named credentials only>,
<Target username - for using new credentials only>,
<Target password - for using new credentials only>,
<Boolean for adding host metrics - optional - default is true>,
<Boolean for adding WebLogic metrics - optional - default is true>,
<Boolean for adding JVM dump - optional - default is true>,
<Boolean for adding heap dumps - optional - default is false>,
<Boolean for adding Automatic Workload Repository (AWR) reports - optional -
```

```
default is false>,
<My Oracle Support username>,
<My Oracle Support password>,
<Service Request (SR) number - required if no CSI given>,
<Customer Support Identifier (CSI) - required if no SR number given>
```

For example:

```
/HCMDomain/Server_1/
SetupApp,fusion_apps_j2ee_app,Other-1,,username,mypassword,,FAadmin,fusionfa1
,,,,,,GENERIC@oracle.com,,3-6586541801
/HCMDomain/Server_1/
SetupApp,fusion_apps_j2ee_app,Other-1,HOST_CREDS,,,WLS_CREDS,,,false,false,fa
lse,true,true,GENERIC@oracle.com,,,15427437
/HCMDomain/Server_1/
SetupApp,fusion_apps_j2ee_app,Other-1,,,,,,,false,,,true,GENERIC@oracle.com,
,3-6586541801
```

Note the following points about the format of incident_packaging_file:

– The delimiter used is a comma (,).

– The order of parameters is fixed. You must provide the parameters in the same order as specified above in the sample file structure.

– Delimiters must be present even if the corresponding parameter is not provided.

– If you want to use a comma in one of the parameters provided, you must escape the comma with a backslash, as shown in This example in which the password has a comma:

```
/HCMDomain/Server_1/
SetupApp,fusion_apps_j2ee_app,Other-1,,username,mypassword,,FAadmin,fusio
n\,fa1,,,,,,GENERIC@oracle.com,,3-6586541801
```

– If you want to use a backslash in one of the parameters provided, you must escape the backslash with a backslash, as shown in This example in which the password has a comma:

```
/HCMDomain/Server_1/
SetupApp,fusion_apps_j2ee_app,Other-1,,username,mypassword,,FAadmin,fusio
n\ \fa1,,,,,,GENERIC@oracle.com,,3-6586541801
```

For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **input_file=heap_dumps_file**

  Fully qualified path to a CSV formatted file containing multiple lines of fully qualified paths to heap dump files to be included in the package. The files whose locations are provided in the file are added as heap dumps to the package.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **input_file=db_awr_file**

  Fully-qualified path to a CSV formatted file containing multiple lines of databases and the credentials used to generate reports for the package. The AWR reports generated by the databases provided in the file are added to the package, assuming that the credentials, if needed, are provided and valid.

  The structure of the CSV file is as follows:

```
<Database name as used in EM>,
<credential name - for using named credential only>,
<username - for using new credential only>,
<password - for using new credential only>,
<role - optional, for using new credential only>
```

For example:

```
Oemrep_database (preferred credentials set in Enterprise Manager)
Oemrep_database,MY_DB_CREDS
Oemrep_database,,sysman,sysman
Oemrep_database,,sysman,sysman,normal
```

Note the following points about the format of db_awr_file:

– The delimiter used is a comma (,).

– The order of parameters is fixed. You must provide the parameters in the same order as specified above in the sample file structure.

For more information about the input_file parameter, see -input_file Syntax Guidelines.

Examples

Example 1

This example shows a fully-qualified path to a CSV formatted file containing one line of details for the Fusion Applications problem to be packaged.

```
/HCMDomain/Server_1/
SetupApp,fusion_apps_j2ee_app,Other-1,,username,mypassword,,FAadmin,fusionfa1,,,,,,GENE
RIC@oracle.com,,
3-6586541801
/HCMDomain/Server_1/SetupApp,fusion_apps_j2ee_app,Other-1,HOST_CREDS,,,
WLS_CREDS,,,false,false,false,true,true,GENERIC@oracle.com,,,15427437
/HCMDomain/Server_1/
SetupApp,fusion_apps_j2ee_app,Other-1,,,,,,,false,,,true,GENERIC@oracle.com,,3-6586541
801
```

Example 2

This example shows a fully-qualified path to a CSV formatted file containing multiple lines of databases and the credentials used to generate reports for the package.

```
Oemrep_database (preferred credentials set in Enterprise Manager)
Oemrep_database,MY_DB_CREDS
Oemrep_database,,sysman,sysman
Oemrep_database,,sysman,sysman,normal
```

pdb_backup

Takes a backup of the data files and metadata xml of a given pluggable database (PDB).

Format

```
emcli pdb_backup     -inputFile="File containing properties required for taking
backup of PDB"
```

Options

• inputFile

Location of the file containing properties required for taking a backup of a PDB.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example takes a back of the PDB contained in the `pdb_backup.props` file:

```
emcli pdb_backup
      -input_file=data:/u01/files/pdb_backup.props
```

Contents of pdb_backup.props:

```
TARGET_HOST_LIST=xyz.abccorp.com
HOST_NORMAL_NAMED_CRED=XYZ_CRED:CRED_OWNER
SRC_CDB_NAMED_CRED=CDB1_CRED:CRED_OWNER
SRC_CDB_TARGET_NAME=CDB1
SRC_CDB_TARGET_TYPE=oracle_database
SRC_PDB_TARGET_NAME=CDB1_PDB1
BACKUP_LOCATION=/scratch/pdbBackup
WORK_DIR_LOCATION=/tmp
ORACLE_HOME_LOC=/scratch/d121hmcasm/product/12.1.0/dbhome_1
```

pdb_clone_management

Creates a new cloned PDB.

Format

```
emcli pdb_clone_management
     [-cloneToOracleCloud = Clone PDB to Container database (CDB) on Oracle
Cloud]     -input_file = pdb_input_file

[ ]  indicates that the parameter is optional.
```

Options

* cloneToOracleCloud

  Specifies if the destination CDB is on Oracle Cloud.

* input_file

  Location of the file containing properties required for cloning a PDB. The allowed properties for this job are:

  SRC_CDB_TARGET = Enterprise Manager target name of the CDB containing the source PDB.

  SRC_CDB_TYPE = Enterprise Manager target type of the CDB containing the source PDB.

  SRC_CDB_CREDS = Named credentials for the source CDB.

  SRC_HOST = Enterprise Manager target name of host containing source CDB. If not provided, will be defaulted from CDB.

  SRC_HOST_CREDS = Named credentials for the source target host.

  SRC_PDB_TARGET = Enterprise Manager target name of the source PDB.

SRC_WORK_DIR = Work directory at source host where files will be temporarily stored. If not provided, will be defaulted to agent work directory.

DEST_HOST = Enterprise Manager target name of host containing destination CDB. If not provided, will be defaulted from CDB.

DEST_HOST_CREDS = Named credentials for the destination target host. If destination host is on OPC, this should be Host SSH credentials.

DEST_LOCATION = Data file location at the destination where the new PDB will be hosted.

DEST_CDB_TARGET = Enterprise Manager target name of CDB where the new PDB should be cloned.

DEST_CDB_TYPE = Enterprise Manager target type of destination CDB.

DEST_CDB_CREDS = Named credentials for the destination CDB.

DEST_PDB_NAME = Name of the new PDB.

EXISTING_BACKUP = Absolute location of the existing backup in the file system, if it should be used to clone new PDB.

EXISTING_BACKUP_METADATA = Absolute location of the metadata template of the backup. Required, if EXISTING_BACKUP is provided.

BACKUP_TYPE = [TAR || OSIMAGE || RMAN]

If existing backup is provided, this represents the type of the backup. If not, this represents the type of backup that should be taken during job execution. If both, EXISTING_BACKUP and BACKUP_TYPE are not provided, source PDB will be unplugged and copied over to destination for creating new clone. After the data files are copied, the source PDB will be plugged back.

Mandatory properties:

SRC_PDB_TARGET, SRC_HOST_CREDS, SRC_CDB_CREDS, SRC_WORK_DIR, DEST_HOST_CREDS, DEST_LOCATION, DEST_CDB_TARGET, DEST_CDB_TYPE, DEST_CDB_CREDS, DEST_PDB_NAME

Clone Types:

Full Clone - Live backup: Takes a backup of the source PDB and creates a new PDB. BACKUP_TYPE specifies the type of backup.

Full Clone - Existing Backup: Uses an existing backup of the source PDB and creates a new PDB. BACKUP_TYPE specifies the type of backup.

EXISTING_BACKUP: Specifies the backup name and EXISTING_BACKUP.

METADATA: Specifies the metadata for the backup.Full Clone - Unplug/Plug: Unplugs the source PDB, creates a new PDB at the destination using the unplugged source, and plugs the source back.

Example

The following example creates a new cloned PDB from the information contained in the `pdb_clone.props` file.

```
emcli pdb_clone_management
      -input_file=data:/u01/files/pdb_clone.props
```

promote_gold_agent_image

Promotes a gold agent image version to the given maturity level.

Format

```
emcli promote_gold_agent_image
      -version_name="gold_image_version_name"
      -maturity="Current/Restricted/Draft"
```

Options

* version_name

  Management Agent gold image that you want to promote.

* maturity

  Gold image maturity level to which the Management Agent gold image should be promoted.

Example

The following example promotes the Management Agent gold image `OPC_AGI_DB_JUL_13` to the `Current` maturity level.

```
emcli promote_gold_agent_image
      -version_name=OPC_AGI_DB_JUL_13
      -maturity=Current
```

provision

Provisions a hardware server using configuration properties from the input file. The configuration properties required for a component can be viewed from the Cloud Control console. After you make a provisioning request, you can view the status of the request from the Enterprise Manager Cloud Control console by using the assignment name (specified by you or the automatically generated name returned to you).

Format

```
emcli provision
      -image="path_to_image"
      -network="network_profile_path"
      -bootserver="boot_server_name"
      -stageserver="stage_server_name"
      -stgcredentials="username"
      -schedule="type:immediate/onetime;timezone:zone;
       startdt:startdate;starttm:time"
      -resettimeout="time"
      -target="hardware_server_label"
      -input_file="config_properties:file_path"
      -assignment="assignment_name"
      [-desc="assignment_description"]

[ ]  indicates that the parameter is optional
```

Options

* **image**

Path to the image (includes the image name). This is the image used for provisioning.

- **network**

  Path name of the network profile.

- **bootserver**

  Name of the boot server.

  Format: hostName:Directory Path

- **stageserver**

  Name of the stage server. hostName:Directory Path.

- **Stgcredentials**

  User name of the stage server.

- **schedule**

  Time when provisioning should be scheduled. This is a string argument that contains multiple name-value pairs separated by `;`. This is used to schedule the provisioning operation. "type" can be `immediate' or `onetime'. If "type" is not immediate, the other values are expected in the Time Zone: string, which is a timezone ID of the format:

  zone Sign TwoDigitHours:Minutes

  zone: Time zone ID (GMT, PDT, and so forth)

  Sign: one of "+ -"

  TwoDigitHours: Digit Digit

  Minutes: Digit Digit

  Digit: One of 0 1 2 3 4 5 6 7 8 9

  Startdt: Date string of the format: MM/DD/YY

  Starttm: Time string of the format: HH:MM

- **resettimeout**

  Reset timeout for the hardware server in minutes.

- **target**

  Target hardware server is specified using the hardware label type.

- **input_file**

  File containing configuration properties.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **assignment**

  Name of the assignment.

- **desc**

  Assignment description. The description is automatically generated if not specified.

Examples

This example submits a job to provision `myimage` on a target with the label of `mylabel`. The job runs immediately with a reset timeout of 100 minutes. Image properties are picked from

properties.txt that overrides the default image. properties.stageserver is used as the staging server, and /private/share as the staging storage with joe as the user name.

```
emcli provision
      -image="Images/myimage"
      -network="Networks/networkprofile"
      -bootserver="booservername.example.com"
      -stageserver="stageserver.example.com:/private/share"
      -stgcredentials="joe"
      -schedule="type:immediate"
      -resettimeout="100"
      -target="mylabel"
      -input_file="config_properties:properties.txt"
      -assignment="provision mylabel"
```

provision_mw_profile

Provisions a non-Oracle middleware provisioning profile.

Examples of non-Oracle middleware include Apache Tomcat, JBoss, etc.

Format

```
emcli provision_mw_profile
      -profile="Profile Location"
      -hosts="List of Hosts"
      -credentials="List of Credentials"
      [-input_file=parameters:"Provisioning Options"]
      [-input_file=host_qualifiers:"Host Qualifiers"]
      [-action="Provisioning Action"]
      [-work_dir="Working Directory"]
      [-analyze]
      [-schedule=
          start_time:yyyy/MM/dd HH:mm;
          [tz:{java timezone ID}];
          [grace_period:xxx];
       ]
```

[ ] indicates that the parameter is optional.

Options

* profile

  Complete software library location of the profile.

* hosts

  Comma separated list of hosts where the profile will be provisioned.

* credentials

  Comma separated list of named credentials used to access the hosts. To pass one credential parameter, enter a name:value pair in the following format:

  credential_name:credential_owner

  Where:

  credential_name is the name of the named credential.

credential_owner is the credentials of the Oracle home owner on the administration server host.

Either a single credential should be provided or the number of credentials should match the number of hosts. If a single credential is provided then it will be used for all of the hosts.

- input_file:properties

  A properties file listing values for all of the parameters required by the commands listed in the profile properties. This is an optional parameter if the commands listed in the profile properties do not require any external parameters.

- input_file:host_qualifiers

  A plain text file containing details about the hosts that are provided for provisioning. This is an optional parameter if the commands listed in the profile properties do not require any host qualifiers.

- action

  One of the provisioning actions as described in the profile properties. This option is not mandatory. If no value is provided then the default action mentioned in the profile properties will be used.

- work_dir

  A temporary working directory for the provisioning process. It will be cleaned up by the procedure at the end. This option is not mandatory. If no value is provided then the agent's working directory will be used.

- analyze

  Use this flag to specify whether to run the procedure in analyze mode. If this option is passed, then the procedure will pause after the prerequisites for manual intervention.

- schedule

  Specify when to run the deployment procedure. If no value is entered, by default the procedure runs immediately. To schedule a procedure, provide:

  – start_time: when the procedure should start.

  – tz: the timezone ID.

  – grace_period: grace period in minutes

Examples

Example 1

The following example provisions a profile named MyProfile1 on two hosts using a single credential at the specified schedule.

```
emcli provision_mw_profile
    -profile="Middleware Provisioning/Generic Profile/MyProfile1"
    -hosts="myhost1.mycompany.com,myhost2.mycompany.com"
    -credentials="MYHOSTCRED:SYSMAN"
    -input_file=parameters:"/tmp/MyProfile1Input.properties"
    -input_file=host_qualifiers:"/tmp/host_details.txt"
```

Example 2

The following example provisions a profile named MyProfile2 on two hosts using individual credentials. The procedure runs the steps specified in the action named 'clone' in the profile properties The provided working directory is used by procedure and it runs in analyze mode.

```
emcli provision_mw_profile
      -profile="Middleware Provisioning/Generic Profile/MyProfile2"
      -hosts="myhost1.mycompany.com,myhost2.mycompany.com"
      -credentials="MYHOST1CRED:SYSMAN,MYHOST2CRED:SYSMAN"
      -input_file=parameters:"/tmp/MyProfile2Input.properties"
      -input_file=host_qualifiers:"/tmp/host_details.xml"
      -action="clone"
      -work_dir="/tmp/mytmpdir"
      -analyze
```

publish_change_request_ccc

Sends change request data to the Change Management Connector, and data processed into the Configuration Change Console. Some of the properties (such as connector_guid, target, and facet) are to be specified as part of customization. All of the data should be able to be mapped to the data required in publishChangeRequest.xsd after XSLT.

Format

```
emcli publish_change_request_ccc
      -connector_guid="ConnectorGUID"
      -change_id="change_ID"
      -last_modified_date="last_modified_date"
      -properties_list="list_of_Change_Management_specific_properties"
      -date_format="Date_format_in_Change_Management_System"
```

Options

- **connector_guid**

- **change_id**

- **last_modified_date**

- **properties_list**

  Specify all relevant properties of the Change Management System required for CCC to process a change request.

  The properties are name,value pairs to be specified as prop_name1=value1;prop_name2=value2 with no quotes for values.

  prop_name and values cannot contain the equals sign (=) or semi-colons (;).

- **date_format**

  Specify a date format in the Change Management System:

  MM/dd/yyyy hh:mm:ss if the date field in change management is "09/14/2011 5:38:24 AM"

publish_event

Publishes a user-reported event to Enterprise Manager. This event is published as an event of the "User-reported event" class. Only users with Manage Target privilege can publish these events for a target. An error message is reported if the publish fails.

After an event is published with a severity other than CLEAR (see below), end-users with appropriate privileges can manually clear the event from the user interface, or you can publish a new event using a severity level of CLEAR and the same details to report clearing of the underlying situation.

Format

```
emcli publish_event
        -target_name="target_name"
        -target_type="target_type_internal_name"
        -message="message_for_event"
        -severity="severity_level"
        -name="event_name"
        [-key="sub_component_name"
        [-context="name1=value1;name2=value2;.."]
        [-separator=context="alt._pair_separator"]
        [-subseparator=context="alt._name-value_separator"]

[ ]  indicates that the parameter is optional
```

Options

*   **target_name**

    Target name.

*   target_type

    Target type name.

*   message

    Message to associate for the event. The message cannot exceed 4000 characters.

*   **severity**

    Numeric severity level to associate for the event. The supported values for severity level are as follows:

    > "CLEAR"
    > "MINOR_WARNING"
    > "WARNING"
    > "CRITICAL"
    > "FATAL"

*   **name**

    Name of the event to publish. The event name cannot exceed 128 characters.

    This is indicative of the nature of the event. Examples include "Disk Used Percentage," "Process Down," "Number of Queues," and so on. The name must be repeated and identical when reporting different severities for the same sequence of events. This should not have any identifying information about a specific event; for example, "Process xyz is

down." To identify any specific components within a target that the event is about, see the key below.

- **key**

  Name of the sub-component within a target this event is related to. Examples include a disk name on a host, name of a tablespace, and so forth. The key cannot exceed 256 characters.

- **context**

  Additional context that can be published for a given event. This is a series of strings of format name:value separated by a semi-colon. For example, it might be useful to report the percentage size of a disk when reporting space issues on the disk. You can override the default separator ":" by using the subseperator , and the pair separator ";" by using the separator .

  The context names cannot exceed 256 characters, and the values cannot exceed 4000 characters.

- **separator**

  Set to override the default ";" separator. You typically use this option when the name or the value contains ";". Using "=" is not supported for this option.

- **subseparator**

  Set to override the default ":" separator between the name-value pairs. You typically use this option when the name or value contains ":". Using "=" is not supported for this .

Examples

Example 1

This example publishes a warning event for "my acme target" indicating that a HDD restore failed, and the failure related to a component called the "Finance DB machine" on this target.

```
emcli publish_event  -target_name="my acme target" -target_type="oracle_acme"
-name="HDD restore failed" -key="Finance DB machine" -message="HDD restoration
failed due to corrupt disk" -severity=WARNING
```

Example 2

This example publishes a minor warning event for "my acme target" indicating that a HDD restore failed, and the failure related to a component called the "Finance DB machine" on this target. It specifies additional context indicating the related disk size and name using the default separators. Note the escaping of the \ in the disk name using an additional "\".

```
emcli publish_event  -target_name="my acme target" -target_type="oracle_acme"
-name="HDD restore failed" -key="Finance DB machine" -message="HDD restoration
failed due to corrupt disk" -severity=MINOR_WARNING -context="disk
size":800GB\;"disk name":\\uddo0111245
```

publish_metric_extension

Publishes a metric extension for use by all administrators. The metric extension must currently be a deployable draft.

Format

```
emcli publish_metric_extension
      -target_type=<metric_extension_target_type>
      -name=<metric_extension_name>
      -version=<metric_extension_version>
```

Options

- **target_type**

  Target type of the metric extension.

- **name**

  Name of the metric extension.

- **version**

  Version of the metric extension to be published.

Example

This example publishes a metric extension of a given target type, name, and version.

```
emcli publish_metric_extension -target_type=<target type of the metric extension> -
name=<name of the metric extension -version=<version of the metric extension>
```

reassoc_masking_definition

Reassociates an existing masking definition with another database target.

Format

```
emcli reassoc_masking_definition
   -definition_name=masking definition name
   -target_name=database target name
   -target_type=database target type
   [-parameters=name1:value1;name2:value2;...]
   [-credential_name=credential_name]
   [-input_file=parameter_tag:file_path]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **definition_name**

  Masking definition name.

- **target_name**

  New database target name with which to associate the masking definition.

- **target_type**

  New database target type with which to associate the masking definition.

- **parameters**

  List of name-value pairs that represent the credentials required for connecting to the database instance. The supported parameters are db_username, db_password, and db_role.

- **credential_name**

  Name of the database credential. This option is mandatory when the db_username and db_password parameters are not specified.

- **input_file**

  Used in conjunction with the parameters option, this option enables you to store parameter values, such as username and password, in a separate file. This option specifies a mapping between a tag and a local file path. The tag is specified in lieu of specific parameter values for the parameters . The tag must not contain colons (:) or semi-colons (;).

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

Output

Success or failure message along with the details.

Examples

Example 1

This example reassociates the masking definition mask_hr_data with the new database target testdb2 :

```
emcli reassoc_masking_definition
    -definition_name=mask_hr_data
    -target_name=testdb2
    -parameters="db_username:system;db_password:password;db_role:NORMAL"
```

Example 2

This example reassociates the masking definition mask_hr_data with the new database target testdb2. The database password is read from the pwd.txt file.

```
emcli reassoc_masking_definition
    -definition_name=mask_hr_data
    -target_name=testdb2
    -parameters="db_username:system;db_password:PWD_FILE;db_role=SYSDBA"
    -input_file="PWD_FILE:pwd.txt"
```

redeploy_plugin_on_agent

Redeploys an existing plug-in on the Management Agents.

Format

```
emcli redeploy_plugin_on_agent
    {-agent_names="agent1[;agent2...]" | -group_name="group1"}
    -plugin="plug-in_id:version"
    [-redeploy_noprompt]
    [-include_dependent_agents]

[ ]  indicates that the parameter is optional.
```

Options

- **agent_names**

  List of Management Agents (host:port) on which the plug-in should be redeployed.

- **plugin**

  ID and version of the plug-in that should be redeployed on the Management Agents.

- **redeploy_noprompt**

  Redeploys the same plug-in that is already available in the Plug-in manager inventory, without prompting you to confirm the redeployment.

- **include_dependent_agents**

  Includes all of the dependent Management Agents and proceeds with plug-in redeployment.

Examples

Example 1

The following example redeploys 12.1.0.2.0 version of the `oracle.sysman.db2` plug-in on the Management Agent named host.example.com:

```
emcli redeploy_plugin_on_agent
     -agent_names="host.example.com:1838"
     -plugin="oracle.sysman.db2:12.1.0.2.0"
```

Example 2

The following example redeploys 12.1.0.2.0 version of the `oracle.sysman.db2` plug-in on the Management Agent named `host.example.com` without prompting you for any confirmation:

```
emcli redeploy_plugin_on_agent
     -agent_names="host.example.com:1838"
     -plugin="oracle.sysman.db2:12.1.0.2.0"
     -redeploy_noprompt
```

refer_swlib_entity_files

Refers one or more files from an entity revision in the software library.

Format

```
emcli refer_swlib_entity_files
       -entity_rev_id="entity_rev_id"
       -file="<relative_file_path>[;<new_file_name>]" | [-removefile="<existing_
          file_name>"]
       -refer_storage="<storage_location_name>;<storage_type>"
       [-use_latest_revision]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **entity_rev_id**

  Identifier of the entity revision. The Software Library home page exposes the identifier for folders and entities as a custom column (Internal ID) and is hidden by default.

- **file**

  Relative path of the file to be referred from the specified storage location. The file name stored in the software library is defaulted to the name of the file being referred. You can optionally specify a different file name, separated by a semi-colon (;).

- **removefile**

  Name of the file to be removed. This is an existing file carried forward from the specified entity revision.Alternatively, you can specify the following values:

  > ALL — Remove all existing files (no carry forward).
  > NONE — Retain all carried forward files.

  The default is NONE.

- **refer_storage**

  The storage location and type for referring to files, separated by a semi-colon (;). The location specified must be in 'active' status. The storage type can be Http, Nfs, or ExtAgent.

- **use_latest_revision**

  Indicates that the latest revision of the entity be used instead of the revision identified by entity_rev_id.

Example

This example refers the file 'scripts/perl/script1.pl' in the HTTP reference file location 'myScripts' from the entity revision identified. The file name associated will be 'new_script.pl'. The identifier of the updated revision is output.

```
emcli refer_swlib_entity_files
        -entity_rev_id="oracle:defaultService:em:provisioning:1:cmp:
         COMP_Component:SUB_Generic:B1B1880C6A8C62AAE040548C42832D14:0.1"
        -file="scripts/perl/script1.pl;new_script.pl"
        -refer_storage="myScripts;Http"
        -use_latest_revision
```

refresh_coherence

Refreshes one or more Coherence clusters.

Format

```
emcli refresh_coherence        -
input_file=coherence_refresh_file:file_path        [-debug]

[ ]  indicates that the parameter is optional
```

Options

- **input_file**

  Fully-qualified path to a CSV-formatted file listing Coherence cluster target per line. For example:

  ```
  ClusterA
  ClusterB
  ```

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **debug**

  Runs the verb in verbose mode for debugging purposes.

Examples

This example reads the my_clusters_name.csv file to determine the clusters to be refreshed to Cloud Control, and then refreshes them.

```
emcli refresh_coherence
  -input_file=coherence_refresh_file:c:\emcli\my_clusters_names.csv
```

refresh_database

Refreshes the database from the latest data in the source database. This command places the database target under blackout and the database is deleted from the Oracle Home. The database is then recreated from the latest data in the source database and the target is removed from blackout.

> **Note:**
>
> This command only applies to full clone test master databases created using the Database Cloning wizard. It does not apply to thin clone databases.

Format

```
emcli refresh_database
      -target_name="database target name"
      -target_type="database target type"
      -input_file=data:"file:path"
```

Options

- target_name

  The target name of the database to be refreshed.

- target_type

  The target type of the database to be refreshed.

- input_file

  The input file containing parameters for the temporary staging location and all passwords to be set:

  – DB_TEMPLATE_STAGE

    The staging area used to store files transferred from the source host.

  – COMMON_DB_SYSTEM_PASSWORD

    The password to be set for SYSTEM user.

  – COMMON_DB_DBSNMP_PASSWORD

    The password to be set for the DBSNMP user.

  – COMMON_DB_SYS_PASSWORD

    The password to be set for SYS user.

  – ASMSYSPWD

The ASM SYS password required to sign in to ASM. This password is only required if the database files are on ASM.

Example

The following example refreshes the Oracle database with the name 'database' using the parameters contained in the `/tmp/a.txt` file:

```
emcli refresh_database        -target_name="database"
       -target_type="oracle_database"
       -input_file=data:"/tmp/a.txt"
```

In this example, the `/tmp/a.txt` has the following content:

```
DB_TEMPLATE_STAGE=/tmp
COMMON_DB_SYSTEM_PASSWORD=welcome
COMMON_DB_DBSNMP_PASSWORD=welcome
COMMON_DB_SYS_PASSWORD=welcome
ASMSYSPWD=welcome
```

refresh_dbprofile

Creates a new snapshot under the specified database profile.

Format

```
emcli refresh_dbprofile
       -comp_loc="Database Profile component location in software library"
```

Options

* comp_loc

   A combination of the database profile location and name.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example creates a new snapshot of the database profile RMAN_Profile with the location `Database Provisioning Profiles/11.2.0.4.0/linux_x64`.

```
emcli refresh_dbprofile       -comp_loc="Database Provisioning Profiles/
11.2.0.4.0/linux_x64/RMAN_Profile"
```

refresh_fa

Refreshes a Fusion Application instance.

If the `-delete_targets` option is not passed, this verb submits a job to refresh all of the WebLogic Domains of the given Fusion Instance.

If the `-delete_targets` option is passed, this verb removes the targets that are not present.

If both `-add_targets` and `-delete_targets` options are passed, this verb adds, updates, and removes targets that are not present in the WebLogic Domains of the Fusion Instance.

Format

```
emcli refresh_fa
     -name=<Fusion_Instance_name>
     [-delete_targets]
     [-add_targets]

[ ]  indicates that the parameter is optional
```

Options

*   **name=<Name of the Fusion Instance>**

    Target name of the Fusion Application instance.

*   **delete_targets**

    Deletes the specified Fusion Application instance targets from the Enterprise Manager Cloud Control monitoring framework. Deleting a target removes it from the Management Repository and does not physically remove the target itself.

*   **add targets**

    Adds specified Fusion Application instance targets to be monitored by Enterprise Manager. The target type specified is checked on the Management Agent for existence and for required properties, such as user name and password for host target types, or log-in credentials for database target types. You must specify any required properties of a target type when adding a new target of this type.

Examples

This example refreshes the Fusion Application instance:

```
emcli refresh_fa -name=fa1
emcli refresh_fa -name=fa1 -delete_targets -add_targets
emcli refresh_fa -name=fa1 -delete_targets
```

refresh_jboss_domain

Refreshes a JBoss Domain target (target type is jboss_domain). For additional information on how to use the refresh_jboss_domain EM CLI verb, refer to Oracle Enterprise Manager Command Line Interface documentation available on the Oracle Technology Network.

Format

```
emcli refresh_jboss_domain     -domain_name="<name of the JBoss Domain target>"
[-debug]

[ ]  indicates that the parameter is optional.
```

Options

*   domain_name

    Fully qualified name of the JBoss domain to be refreshed.

*   debug

    When specified, additional debug information is displayed.

Example

The following example refreshes a JBoss domain:

```
emcli refresh_jboss_domain -domain_name=/host1.example.com/1234
```

refresh_jboss_partition

Refreshes a JBoss partition target (the target type is jboss_partition). For additional information on how to use the refresh_jboss_partition EM CLI verb, refer to the Oracle Enterprise Manager Command Line Interface documentation available on the Oracle Technology Network.

Format

```
emcli refresh_jboss_partition     -partition_name="<name of the JBoss Partition
target>"     [-debug]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- partition_name

  Fully qualified name of the JBoss Partition to be refreshed.

- debug

  When this option is specified, additional debug information is displayed.

Example

The following example refreshes a JBoss partition.

```
emcli refresh_jboss_partition
     -partition=DefaultPartition01
```

refresh_was

Refreshes one or more IBM WebSphere cells (the target type is websphere_cell). The EM CLI verb refresh_was can be used to refresh either a single WebSphere cell (specified using the -cell_name arg) or all existing WebSphere Cells (specified using the -all flag).

Format

```
emcli refresh_was     [-cell_name="<name of the WebSphere Cell target"] | [-
all]     [-debug]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- cell_name

  Fully qualified name of the WebSphere cell to be refreshed. This option cannot be specified if the -all argument is specified.

- all

When specified refreshes all existing WebSphere cell targets. This option cannot be used if -cell_name argument is specified.

- debug

  When this option is specified, additional debug information is displayed.

Examples

Example 1

The following example refreshes the specified WebSphere cell:

```
emcli refresh_was
     -cell_name=host1.wasCell01
```

Example 2

The following example refreshes all the existing WebSphere cells.

```
emcli refresh_was
     -all
```

refresh_wls

Enables/disables a refresh for one or more Oracle WebLogic Server Domains (target type --> weblogic_domain). This verb reads a file labeled domain_refresh_file in order to refresh the WebLogic Server. The domain_refresh_file is required; refresh cannot occur without it. You must create the file prior to performing refresh.

Format

```
emcli refresh_wls
        -input_file=domain_refresh_file:file_path
        [-debug]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **input_file**

  Fully-qualified path of the CSV(Comma-Separated Values) file that contains multiple lines of the Target name and Refresh action (Enable/Disable refresh of the WLS domains/farms to be refreshed).

  Note the following advisory information about the format of domain_refresh_file:

  – The target name should be the fully-qualified name of the domain target.

  – Every target is treated as type weblogic_domain.

  – Valid values of the refresh option are "E", "D", and "R". "E" enables a refresh for the WLS Domain, "D" disables the refresh for the WLS Domain, and "R" removes targets that are deleted from the WebLogic Domain.

  – A comma (,) is used as the delimiter.

  – The total number of tokens in each line is fixed, and should be equal to 2.

  – The order of parameters is fixed. You must provide the parameters in the same order as specified below in the sample file structure for domain_refresh_file:

```
        /Farm01_base_domain/base_domain,D
            /Farm02_base_domain/base_domain,E
            /Farm03_base_domain/base_domain,R
```

The first entry disables the refresh for target /Farm01_base_domain/ base_domain, the second entry enables a refresh for target / Farm02_base_domain/base_domain, and the third entry removes targets from Enterprise Manager that are deleted from /Farm03_base_domain/ base_domain.

For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **debug**

  Runs the verb in verbose mode for debugging purposes.

Example

```
$emcli refresh_wls
        -input_file=domain_refresh_file:/tmp/refresh/emcli/
            domain_refresh_file.csv -debug
```

refresh_wmq

Refreshes one or more IBM WebSphere MQ clusters (where the target type is wmq_cluster). The verb can be used to refresh either a single WebSphere MQ cluster (specified using the -cluster_name argument) or all existing WebSphere MQ clusters (specified using the -all flag).

Format

```
emcli refresh_wmq
        [-cluster_name="<name of the WebSphere MQ Cluster target"] | [-all]
        [-debug]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- cluster_name

  Fully qualified name of the WebSphere MQ cluster to be refreshed. This argument cannot be specified if -all flag is used.

- all

  When specified refreshes all existing WebSphere MQ cluster targets. This flag cannot be used if -cluster_name arg is specified.

- debug

  If specified, runs the verb in verbose mode for debugging purposes.

Examples

Example 1

The following example refreshes the specified IBM Webshpere MQ cluster:

```
emcli refresh_wmq -cluster_name=wmqCluster01
```

Example 2

The following example refreshes all IBM Webshpere MQ clusters:

```
emcli refresh_wmq -all
```

register_forwarder_agents

Takes a list of agents and registers each agent as a forwarding agent.

Format

```
emcli register_forwarder_agents
     -agent_list="agent_list"
```

Options

*   agent_list

    List of agents that need to be registered as forwarders. The agents must be separated by space.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example registers agent1 and agent2 as forwarding agents.

```
emcli register_forwarder_agents
     -agent_list="agent1 agent2..."
```

register_hybridgateway_agents

Takes a list of agents and marks each agent as a hybrid gateway.

Format

```
emcli register_hybridgateway_agent
     -hybridgateway_agent_list="hybridgateway_agent_list"
```

Options

*   hybridgateway_agent_list

    List of agents to be registered as hybrid gateway agents. The agents must be separated by a space.

*   named_credential

    Named credential used to make SSH connection to cloud host, it is used for the network check.

*   named_credential_owner

    Owner of named credential.

*   cloud_hostname

    Cloud hostname where you want to install hybrid agent.

- ignore_central_agent_check

  Flag used to skip the central agent check for the specified list of agents.

- ignore_network_check

  Flag used to skip the network check for the specified list of agents. This flag makes the following parameters optional:

  - -named_credential

  - -named_credential_owner

  - -cloud_hostname

- ssh_port

  Specified the SSH port used to check the network. Default: 22

- timeout

  Specifies the amount of time (in seconds) the network check process will wait for a connection. Default: 5

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example registers agent1 and agent2 as hybrid gateways.

```
emcli register_hybridgateway_agent
     -hybridgateway_agent_list="agent1 agent2"
```

Example

The following example registers agent1 and agent2 as hybrid gateways while using –ignore_network_check and -ignore_central_agent_check.

```
emcli register_hybridgateway_agent -hybridgateway_agent_list="agent1 agent2" -
ignore_network_check -ignore_central_agent_check
```

Example

The following example registers agent1 and agent2 as hybrid gateways while using –ignore_central_agent_check.

```
emcli register_hybridgateway_agent -hybridgateway_agent_list="agent1 agent2"
          -named_credential="named_credential" -
named_credential_owner="credential_owner"
          -cloud_hostname="cloud_hostname" -ignore_central_agent_check
```

register_storage

Registers a storage registered in Enterprise Manager.

Format

Standard Mode

```
emcli register_storage
     -storage_name="<storage name>"
     -vendor="<NetApp|Sun ZFS|Solaris ZFS|EMC>"
```

```
[-protocol="<http|https>"]
[-smis_url="http://host-name:port"]
-storage_cred="<credential_name>"
[-aliases="<alias1;alias2;alias3;...>"]
[-storage_agents="<host1:cred1;host2:cred2;...>"]
[-frequency="<Minutes:40|Hours:3|Days:2|Weeks:3>"]
```

Interactive or Script Mode

```
register_storage(
storage_name="<storage name>"
,vendor="<NetApp|Sun ZFS|Solaris ZFS|EMC>"
[,protocol="<http|https>"]
[,smis_url="http://host-name:port"]
,storage_cred="<credential_name>"
[,aliases="<alias1;alias2;alias3;...>"]
[,storage_agents="<host1:cred1;host2:cred2;...>"]
[,frequency="<Minutes:40|Hours:3|Days:2|Weeks:3>"]
)
```

```
[ ]  indicates that the parameter is optional.
```

Options

- storage_name

  Name of the storage.

- vendor

  Vendor of the storage hardware.

- protocol

  Protocol to be used to communicate with the storage hardware. Input required only for NetApp storage.

- smis_url

  The URL to be used to communicate with the EMC storage hardware. Input required only for EMC storage.

- storage_cred

  Credential to be used to communicate with the storage hardware. This has to be owned by the user.

- aliases

  The interfaces and aliases of the storage that will be used to mount the volumes on host in the aliases section. This should include fully qualified domain name, IP address, DNS alias or any other name that points to storage data interface.

- storage_agents

  The hosts that can be used to perform operations on this storage device.

- frequency

  The frequency at which the storage synchronize job should be run.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

The following command registers a Sun ZFS storage server:

```
emcli register_storage
              -storage_name="sunzfs1"
              -vendor="Sun ZFS"
              -storage_cred="sunzfs1_cred"
              -aliases="sunzfs1.example.com;sunzfs1_eg"
              -storage_agents="host1:cred1;host2:cred2"
              -frequency="Hours:3"
```

Example 2

The following command registers a NetApp storage server:

```
emcli register_storage
              -storage_name="napstr1"
              -vendor="NetApp"
              -protocol="https"
              -storage_cred="napstr1_cred"
              -aliases="netapp1.example.com;netapp1"
              -storage_agents="host1:cred1;host2:cred2"
              -frequency="Minutes:40"
```

reimport_swlib_metadata

Re-imports software library metadata from the OMS and deployed plug-in Oracle
Homes. Any Oracle-owned entity with missing files is restored to the corresponding
upload storage location.

Format

```
emcli reimport_swlib_metadata
```

Options

None.

relocate_bda_target

Relocates monitoring agents for BDA targets. Use it to relocate monitoring of a specific
target to another agent on a destination host, or use it to relocate monitoring of all
shared targets on a cluster to other agents on the same BDA rack.

Format

```
emcli relocate_bda_target
        -target="target_name" -dest_host="destination_host_name" | -all_shared
        -cluster="cluster_name">
```

Options

- **target**

  A target in the BDA network.

- **dest_host**

  Name of the host where monitoring of the specified target is to be relocated.

- **all_shared**

  Specifies to relocate monitoring of all shared targets in the named cluster.

- **cluster**

  Name of the cluster for whom monitoring is to be relocated.

Examples

Example 1

The following example relocates monitoring of the target `hdfs_USA_acme` to the agent on host `acme101.com`:

```
emcli relocate_bda_target
       -target="hdfs_USA_acme"
       -dest_host="acme101.com"
```

Example 2

The following example relocates monitoring of all shared targets on the cluster `acme101` to other valid agents on the same BDA rack:

```
emcli relocate_bda_target
       -all_shared
       -cluster="acme101"
```

relocate_pdb

Relocates a pluggable database from one container database to another.

Format

```
emcli relocate_pdb
       -pdb_target_name="pluggable database target name"
       -input_file="path of the input file"
```

`[ ]` indicates that the parameter is optional.

Options

- **pdb_target_name**

  Name of the pluggable database target.

- **input_file**

  Path of the file containing the following input properties:

  ```
      SRC_CDB_CRED = SYSDBA source container database credentials (format -
  CRED_NAME:OWNER)
      SRC_HOST_CRED = Source container database host credentials (format -
  CRED_NAME:OWNER)
      SRC_WORK_DIR = Staging location for pluggable database backup on the source
  host
      AVAILABILITY = Pluggable database relocation mode. Allowed values - <MAXIMUM|
  NORMAL>
      DEST_PDB_NAME = Name of the destination pluggable database
  ```

```
        DEST_PDB_DISPLAY_NAME = Display name of the destination pluggable
database
        DEST_CDB_TARGET_NAME = Destination container database target name
        DEST_CDB_TARGET_TYPE =  Destination container database target type
        DEST_CDB_CRED = SYSDBA destination container database credentials
(format - CRED_NAME:OWNER)
        DEST_HOST_CRED = Destination container database host credentials (format
- CRED_NAME:OWNER)
        DB_LINK_NAME = Name of an existing Database Link
        DEST_WORK_DIR = Staging location for pluggable database backup on the
destination host
        IS_BKP_LOC_SHARED = Flag to indicate if backup staging location is
shared across source and destination hosts. Allowed values - <Y|N>
        STORAGE_LOCATION = Destination datafiles location of the relocated
pluggable database
        USE_SAME_STORAGE_LOCATION = Flag to indicate if the pluggable database
should be relocated without moving its datafiles. Allowed values - <Y|N>
        STORAGE_MAX_SIZE = Amount of storage to be used by all tablespaces which
belong to the relocated pluggable database
        STORAGE_MAX_SHARED_TEMP_SIZE = Amount of storage in the default
temporary tablespace shared by all pluggable databases
        LOGGING_TYPE =  Logging attribute for the relocated pluggable database.
Allowed values - <LOGGING|NO_LOGGING>
        EXCLUDE_STANDBYS = Flag to indicate if the relocated pluggable database
should be excluded from all standby container databases. Allowed values -<Y|
N>
        CUSTOM_PRE_SCRIPT_URN = URN of the script in Software library to be run
before pluggable database relocation
        CUSTOM_POST_SCRIPT_URN = URN of the script in Software library to be run
after pluggable database relocation
        POST_SQL_SCRIPT_URN = URN of the SQL script in Software library to be
run after pluggable database relocation
        RUN_POST_SQL_USER = User name to run the post SQL script
        RUN_POST_SQL_PWD = Password of the user to run the post SQL script
```

Example

Example 1

This example submits the procedure to relocate a pluggable database
"PRODCDB_SALES".

```
 emcli relocate_pdb
        -pdb_target_name="PRODCDB_SALES"
        -input_file=data:/u01/relocate.props
```

relocate_targets

Moves all of the collections and blackouts for targets from the source Agent to the
destination Agent, and makes the destination Agent the monitoring Agent for these
targets in Enterprise Manager.

Format

```
emcli relocate_targets
        -src_agent=<source_agent_target_name>
        -dest_agent=<dest_agent_target_name>
        -target_name=<name_of_target_to_be_relocated>
        -target_type=<type_of_target_to_be_relocated>
        -copy_from_src
```

```
-changed_param=<propName>:<propValue>
-input_file:dupTargets=<targets_contents>
-input_file:moveTargets="complete path to file containing targets with
  overridden property values"
-copy_from_src [-changed_param=<propName>:<propValue>]*
[-ignoreRelatedTargets]
[-noHostColumnUpdate]
[-ignoreTimeSkew=yes]
[-changed_param=MachineName:mmmm ]
[-force=yes]
```

```
[ ]  indicates that the parameter is optional
```

> **Note:**
>
> To relocate a composite target, you must specify the input_file:dupTargets , and you
> cannot combine -target_type or -target_name.

> **Note:**
>
> For non-Sysman users, Full Any Target and Add Any Target privileges should be
> granted.

Modes

There are two modes for this verb:

- **Create Mode**

  This mode creates a list of targets on the destination Management Agent that already
  exists and is monitored by the source Management Agent in Enterprise Manager. It
  moves all the collections and blackouts for these targets from the source Management
  Agent to the destination Management Agent, and makes the destination Agent the
  monitoring Agent for these targets in Enterprise Manager.

  ```
  emcli relocate_targets -src_agent=<source_agent>
        -dest_agent=<destination_agent>
        -input_file=dupTarget:<complete_path_to_file>;
        [-ignoreTimeSkew=yes]
  ```

- **Exist Mode**

  In this mode, the target also exists at the destination.

  ```
  emcli relocate_targets
        -src_agent=<source_agent_target_name>
        -dest_agent=<destination_agent_target_name>
        -target_name=<target_name>
        -target_type=<target_type>
        [-ignoreTimeSkew=yes]
        [-force=yes]
  ```

In all cases, relocation moves all collections and blackouts for these targets from the source
Agent to destination Agent, and makes the destination Agent the monitoring Agent for these
targets in Enterprise Manager.

Options

- **src_agent**

  Management Agent currently monitoring the targets. If srcAgent is not known, enter currentOwner as the argument.

- **dest_agent**

  Management Agent that should monitor the targets.

- **target_name**

  Name of the target that needs to be moved.

- **target_type**

  Type of target that needs to be moved.

- **changed_param**

  The value of the propName property in the target should be changed to propValue.

- **input_file=dupTargets**

  Takes a file name that contains all the targets and its properties as seen in `targets.xml`. The contents of the file must have the same format as `targets.xml`.

  To relocate a composite target, you must specify the input_file:dupTargets, and you cannot combine -target_type or -target_name.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **input_file=moveTargets**

  Takes a file name that contains a list of targets, one per line, in the following format:

  ```
  <targetType>:<targetName>[;<propName>=<propValue>]*
  ;lkj;lkj;lkj
  ```

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **copy_from_src**

  Copies target properties from the source Agent.

- **ignoreTimeSkew**

  If specified, the target is relocated, ignoring the time skew between the source and destination Agent.

- **ignoreRelatedTargets**

  Moves related targets when not specified. Specified to move onlhy the targets on the command line.

- **noHostColumnUpdate**

  Preserves the host of the relocated tartet when specified. Otherwise, the host is updated to be the new Agent's host.

- **changed_param**

Specify the new MachineName as part of relocate operation, as it is different for each host.

- **force**

  If the command is executed with the `-force=yes` switch, the composite target is automatically relocated with its related targets. If the command is executed without this switch, an error message appears if it is a composite target.

Output

Output message of the command execution.

Examples

Example 1

The following Create Mode example creates a target on the destination Agent by copying the target property content from the source Agent, while allowing some property values to be changed.

```
emcli relocate_targets
      -src_agent=<source_agent>
      -dest_agent=<destination_agent>
      -target_name=<target_name>
      -target_type=<target_type>
      -copy_from_src
      [-ignoreTimeSkew=yes]
      [-changed_param=<Propname>:<Value>]*
```

Example 2

The following Create Mode example creates a list of targets on the destination Agent specified in the moveTargets file. You can specify property value overrides.

```
emcli relocate_targets
      -src_agent=<source_agent>
      -dest_agent=<destination_agent>
      -input_file=moveTargets:<complete_file_path>
      [-ignoreTimeSkew=yes]
```

relocate_wls

Automate the steps required when Weblogic Server is migrated from physical host to logical host.

Format

```
 emcli relocate_wls
  -target_name = <domain target name to relocate> -host = <new host of domain>
  -port = <new port of domain> -dest_agent = <new destination agent>
  [-input_file = old_to_new_host_mapping_file:"<fully qualified path of
old_to_new_host_mapping_file.csv>"]
  [-no_delete]
  [-debug]

[ ]  indicates that the parameter is optional.
```

Options

- **-target_name**

Complete Target name of Domain to be migrated

- **-host**

    Host Name of the Admin Server running on new host

- **-port**

    Port of the Admin Server running on new host

- **-dest_agent**

    New agent to relocate targets of WebLogic Domain

- **-input_file**

    Fully-qualified path of the CSV-formatted file that contains multiple lines of old host to new host mapping. This is used to update host property of targets from old host to the new host provided in the mapping file.

    Each line has the following format:

    ```
    host1.example.com,newhost1.example.com
    host2.example.com,newhost2.example.com
    host3.example.com,newhost3.example.com
    ```

- **-debug**

    Runs the verb in verbose mode for debugging purposes.

- **-no_delete**

    By default, node manager targets which contain old host will be deleted. If this option is provided, node manager targets will not be deleted.

Example

Example 1

The following example migrates a target from an old host to a new host.

```
 emcli relocate_wls -target_name=/EMGC_GCDomain/GCDomain
  -host=newhost1.example.com
  -port=7101
  -dest_agent=newhost1.example.com:3872
  -input_file=old_to_new_host_mapping_file:/tmp/hostmapping.txt
```

remove_association_cs_group_targets

Removes the associations for the specified compliance standard from the specified groups of targets.

Format

```
emcli remove_association_cs_group_targets
     -cs_iname="<internal_name_of_standard>"
     -author="<author>"
     -version="<version>"
     (-group_names="<group_name_list>" | -group_names_file="<file_name>")

[ ]  indicates that the parameter is optional.
```

Options

- cs_iname

  Internal name of the compliance standard.

- author

  Author of the compliance standard.

- version

  Version of the compliance standard.

- group_names

  Comma separated list of group names.

- group_names_file

  Name of the file that contains the group names. The group names can be either comma-separated values or in a file where the group names are listed on separate lines.

  Examples are:

  - group_names_file=group1,group2,group3

  - group_names_file="group.txt" where group.txt contains the following lines:

      group1
      group2
      group3

  Note: Use either group_names or group_names_file.

Example

The following example removes associations for the second version of the security standard, authored by Jones, for the groups named tgt_grp1 and tgt_grp2.

```
emcli remove_association_cs_group_targets
    -cs_iname="security_standard"
    -author="Jones"
    -version="2"
    -group_names="tgt_grp1,tgt_grp2"
```

remove_beacon

Removes a beacon from the monitoring set of beacons.

Format

```
emcli remove_beacon
      -name=<target_name>
      -type=<target_type>
      -bcnName=<beacon_name>

[ ]  indicates that the parameter is optional
```

Options

- **name**

Service target name.

- **type**

  Service target type.

- **bcnName**

  Beacon name to remove.

Examples

This example removes MyBeacon from the MyTarget service target of type generic_service.

```
emcli remove_beacon -name='MyTarget' -type='generic_service'
     -bcnName='MyBeacon'
```

remove_chargeback_entity

Removes the given entity from Chargeback.

Format

```
remove_chargeback_entity
     -entity_name="eName"
     -entity_type="eType"
     -[ entity_guid="entity guid" ]

[ ]  indicates that the parameter is optional
```

Options

- **entity_name**

  Name of the entity to be removed from Chargeback.

- **entity_type**

  Type of entity to be removed from Chargeback.

- **entity_guid**

  guid of the entity to be removed to Chargeback.

  When more than one entity is active in Chargeback with the given entity name and entity type, the command lists all such entities with additional details such as creation date, parent entity name, entity guid, and so forth to choose the correct entity. Select the correct entity from the given list and execute the command again with entity guid as the parameter instead of entity name and entity type.

Examples

This example removes "db1", an oracle_database entity, from Chargeback.

```
emcli remove_chargeback_entity -entity_name="db1" -entity_type="oracle_database"
```

See Also

add_chargeback_entity
assign_charge_plan
assign_cost_center
list_chargeback_entities

remove_cs_target_association

Removes the specified standard target associations.

Note: When the standard is provided by Oracle, the <std_name> is the standard internal name.

Format

```
remove_cs_target_association
 -name="<std_name>"
 -version="<std_version>"
 -author="<author_name>"
 -target_list="<target_name>[,<target_name>]*"
 -target_list_file="<file_name>"
```

Options

*   name

    Name of the standard.

*   version

    Version of the standard.

*   author

    Author of the standard.

*   target_list

    Name of the targets. Use this option when removing the compliance standard association from a small number of targets. Targets are separated by commas. When providing a group target, it should be appended with ":Group". Examples are:

    ```
    -target_list="slc0host"
    -target_list="slc0host,slc-host01"
    -target_list="slc0host,host_grps:Group"
    ```

*   target_list_file

    Name of the file that contains the list of targets. The targets can be either comma-separated values or in a file where the targets are listed on separate lines. Examples are:

    ```
    -target_list_file=slc0host,slc0host1,slc0host02
    -target_list_file="slc0host.txt" Where slc0host.txt contains the following lines:
        slc0host
        slc0host01
        slc0host02
    ```

    Note: Use either the target_list option or the target_list_file option.

Examples

Example 1

The following example removes the standard target association named "secure configuration for host" and uses the target_list option to remove the targets associated with the standard.

```
emcli remove_cs_target_association
    -name="secure configuration for host"
    -version="1"
    -author="sysman"
    -target_list="host1,host2"
```

Example 2

The following example removes the standard target association named "secure configuration for host" and uses the target_list_file option to remove the targets associated with the standard. The targets listed in the file are either comma separated values or each target is listed on a separate line.

```
emcli remove_cs_target_association
    -name="secure configuration for host"
    -version="1"
    -author="sysman"
    -target_list_file="file with target name list"
```

remove_metric_data_load_limits

Removes the customized data load limits for specified targets and metrics.

This verb supports bulk operation for the following cases:

- All metrics of all targets for a specified target type

- All metrics of specified targets for a specified target type

- Specified metrics of all targets for a specified target type

Format

```
emcli remove_metric_data_load_limits
        [-target_type="host"]
        [-targets="host1;host2;"]
        [-metrics="Load;Filesystems;"]

[ ]  indicates that the parameter is optional.
```

Options

- -target_type

  Target type that you are removing the customized data load limits from. If you want to perform the operation on all targets, then skip this option.

- -targets

  Semicolon separated list of targets for a specified target type. You must use the -target_type option with this option. If you want to perform the operation on all targets for a specified target type, then skip this option.

- -metrics

  Semicolon separated list of metrics of a specified target type. You must use the -target_type option with this option. If you want to perform the operation on all metrics for a selected target type, then skip this option.

Examples

Example 1

The following command removes the customized metric data loading limits for all metrics of all host targets.

```
emcli remove_metric_data_load_limits
        -target_type="host"
```

Example 2

The following command removes the customized metric data loading limits for all metrics of the "myhost1.example.com" and "myhost2.example.com" host targets.

```
emcli remove_metric_data_load_limits
    -target_type="host"
    -targets="myhost1.example.com;myhost2.example.com;"
```

remove_mos_credentials

Removes My Oracle Support preferred credentials from OMS.

Format

```
emcli remove_mos_credentials
```

remove_service_system_assoc

Removes the system for a given service.

Format

```
emcli remove_service_system_assoc
      -name='name'
      -type='type'
```

Options

- **name**

  Service name.

- **type**

  Service type.

Examples

This example removes the system for the generic service named my service.

```
emcli remove_service_system_assoc
      -name='my service' -type='generic_service'
```

remove_storage

Removes a storage registered in Enterprise Manager.

Format

Standard Mode

```
emcli remove_storage
        -storage_name="<storage name>"
```

Interactive or Script Mode

```
remove_storage(
        storage_name="<storage name>"
)
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   storage_name

    Name of the storage.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example removes the storage server:

```
emcli remove_storage
                -storage_name="sunzfs1"
```

remove_swlib_storage_location

Removes a storage location from the software library. The alternate storage location where the existing files need to be migrated should also be specified. For upload file storage types, OMS shared and the OMS Agent file system, a job is submitted to perform the migration of files, subsequent to which the location is removed. For these upload file storage types, the alternate location need not be of the same storage type, which is not the case for locations of referenced file storage types.

Format

```
emcli remove_swlib_storage_location
        -name="src_location_name"
        -type="OmsShared|OmsAgent|Http|Nfs|ExtAgent"
        -migrate_to_loc="dest_location_name"
        [-migrate_to_type="OmsShared|OmsAgent|Http|Nfs|ExtAgent"]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **name**

    Name of the storage location to be removed.

*   **type**

    Type of storage location, which can be one of:

```
OmsShared
OmsAgent
Http
Nfs
ExtAgent
```

- **migrate_to_loc**

  Name of the alternate storage location where existing files need to be migrated.

- **migrate_to_type**

  Type of the alternate storage location, which can be one of:

  ```
  OmsShared
  OmsAgent
  Http
  Nfs
  ExtAgent
  ```

  The default is the storage type of the location being removed.

  > **Note:**
  >
  > This option can be different from the type option specified only for OmsShared and OmsAgent storage types. For all other storage types, migrating files across storage types is not supported, and therefore, type and migrate_to_type (if specified) must be the same.

Examples

Example 1

This example removes an OMS shared file system storage location named 'myOMSSharedLocation' and migrates all of its files to another OMS shared file system storage location named 'myNewOMSSharedLocation'. A job is submitted for performing the file migration. The location being removed will be moved to 'Inactive' status during file migration and subsequently removed.

```
emcli remove_swlib_storage_location
        -name="myOMSSharedLocation"
        -type="OmsShared"
        -migrate_to_loc="myNewOMSSharedLocation"
```

Example 2

This example removes an OMS shared file system storage location named 'myOMSSharedLocation' and migrates all of its files to an OMS Agent file system storage location named 'myNewAGTLocation'. A job is submitted for performing the file migration. The location being removed will be moved to 'Inactive' status during file migration and subsequently removed.

```
emcli remove_swlib_storage_location          -name="myOMSSharedLocation"          -
type="OmsShared"          -migrate_to_loc="myNewAGTLocation"          -
migrate_to_type="OmsAgent"
```

remove_target_from_rule_set

Removes a target from an enterprise rule set.

Privilege Requirements: Super Administrators can add a target to any enterprise rule set except for predefined (out-of-box) rule sets supplied by Oracle.

Only the owner or co-author of a rule set can add a target to it.

Format

```
emcli remove_target_from_rule_set
      -rule_set_name="rule set name"
      -target_name="target name"
      -target_type="internal name for target type"
      [-rule_set_owner=<ruleset owner>]

[ ]  indicates that the parameter is optional
```

Options

*   **rule_set_name**

    Name of an enterprise rule set. This option only applies to rule sets associated with a list of targets.

*   **target_name**

    Name of the target to be removed.

*   **target_type**

    Type of the target to be removed. For example, *host*.

*   **rule_set_owner**

    Optionally, you can specify the owner of the rule set.

Examples

The following example removes the host target *myhost.com* from a rule set named *rules*. This rule set is owned by the administrator sysman.

```
emcli remove_target_from_rule_set -rule_set_name='rules' -
target_name='myhost.com' -target_type='host' -rule_set_owner='sysman'
```

remove_target_property

Removes the target property from all targets of the specified target type. This also removes all values associated with this target property.

Format

```
emcli remove_target_property
      -target_type="target_type"
      -property="property_name"
```

Options

*   **target_type**

Target type for which you want to remove this property. To remove this property from all target types for which it is defined, you can specify the "*" wildcard character.

- **property**

  Name of the property you want to remove. Property names are case-sensitive. You cannot remove the following Oracle-provided target properties:

  Comment, Deployment Type, Line of Business, Location, Contact

Examples

Example 1

This example removes the target property Owner from all targets of type oracle_database. This also removes all values associated with this target property.

```
emcli remove_target_property  -target_type="oracle_database" -property="Owner"
```

Example 2

This example removes the target property Owner from all targets. This also removes all values associated with this property for all target types.

```
emcli remove_target_property  -target_type="*" -property="Owner"
```

remove_update

Removes an update.

Format

```
emcli remove_update
        -id="internal id"
```

Options

- **id**

  Internal identification for the update to be removed.

Examples

This example submits a job to remove the update, and prints the job execution ID upon submission.

```
emcli remove_update
      -id="914E3E0F9DB98DECE040E80A2C5233EB"
```

rename_service_template

Renames a Service Template.

Format

```
emcli rename_service_template      -name_old="<Current_Name_of_Service_Template>"
-name_new="<New_Name_of_Service_Template>"      -
service_family="<Name_of_Service_Family>"
```

Options

- **name_old**

  Current name of the Service Template.

- **name_new**

  New name of the Service Template.

- **service_family**

  Name of the Service Family.

Examples

```
emcli rename_service_template     -name_old="Web_Logic"       -
name_new="Web_Logic_V1"      -service_family="MWAAS"
```

displays the following output:

```
Service Template renamed from "Web_Logic" to "Web_Logic_V1" successfully
```

rename_target

Renames the repository-side target.

Format

```
emcli rename_target
     -target_type=<type1>
     -target_name=<old_target1>
     -new_target_name=<new_target1>
```

Options

- **target_type**

  Target type of the target being renamed.

- **target_name**

  Existing name of the target.

- **new_target_name**

  New name of the target.

Examples

This example renames the repository-side target.

```
emcli rename_target
     -target_type="oracle_em_service"
     -target_name="TestService1"
     -new_target_name="NewTestService1"
```

rename_targets_property_value

Changes the value of a property for all targets.

Format

Standard Mode

```
 emcli rename_targets_property_value
   -property_name="null"
   -property_value="null"
   -new_property_value="null"

[ ]  indicates that the parameter is optional.
```

Interactive or Script Mode

```
rename_targets_property_value(
   property_name="null"
   ,property_value="null"
   ,new_property_value="null"
   )
```

Options

*   -property_name

    Name of the property to be changed.

*   -property_value

    Existing value of property.

*   -new_property_value

    New value of property

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following command changes the value for all targets with the orcl_gtp_lifecycle_status property from Product to Development:

```
emcli rename_targets_property_value
   -property_name="orcl_gtp_lifecycle_status"
   -property_value="Production"
   -new_property_value="Development"
```

reschedule_instance

Reschedules a submitted procedure instance. You can only reschedule scheduled instances.

Format

```
emcli reshedule_instance
     -instance=<instance_guid>
     [-exec=<execution_guid>]
     [-name=<execution_name>]
     [-owner=<execution_owner>]
     -schedule=
         start_time:yyyy/MM/dd HH:mm;
         [tz:<java_timezone_ID>];
```

```
          [grace_period:xxx]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **instance**

  GUID of the instance to execute.

- **exec**

  Execution GUID.

- **name**

  Execution name.

- **owner**

  Execution owner.

- **schedule**

  Schedule for the procedure instance:

  – **start_time** — When the procedure should start.

  – **tz —** Optional time zone ID.

  – **grace_period** — Optional grace period in minutes.

Examples

```
emcli reschedule_instance -instance=16B15CB29C3F9E6CE040578C96093F61 -
schedule="start_time:2011/8/21 21:23;tz:America/New_York;grace_period:60"
```

resecure_agent

Resecures a Management Agent already secured. This verb requires operator privilege or full privilege on the Management Agent.

Format

```
emcli resecure_agent
        -agent_name="agent_target_name"
        -registration_pwd="registration_password"
        [-host_username="agent_host_username" -host_pwd="agent_host_password"]
        [-credential_name="credential_name"]
        [-credential_setname="credential_setname_of_agent"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **agent_name**

  Name of the Management Agent target.

- **registration**

  Registration password to securely communicate with OMS.

- **host_username**

  User name of the OS user (on the host) who owns the Management Agent.

- **host_pwd**

  Password of the OS user (on the host) who owns the Management Agent.

- **credential_name**

  Name of the saved credential.

- **credential_setname**

  Name of the credential set of the Management Agent. Example: "HostCreds".

Examples

Example 1

```
emcli resecure_agent -agent_name="agent.example.com:1234"
                     -registration_pwd="test_pwd"
                     -host_username="test_user"
                     -host_pwd="test"
```

Example 2

```
emcli resecure_agent -agent_name="agent.example.com:1234"
                     -registration_pwd="test_pwd"
                     -credential_name="MyMachineCredential"
```

restart_agent

Restarts a Management Agent. This verb requires operator privilege or full privilege on the Management Agent.

Format

```
emcli restart_agent
      -agent_name="agent_target_name"
      [-host_username="agent_host_username" -host_pwd="agent_host_password"]
      [-credential_name="credential_name"]
      [-credential_setname="credential_setname_of_agent"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **agent_name**

  Name of the Management Agent target.

- **host_username**

  User name of the OS user (on the host) who owns the Management Agent.

- **host_pwd**

  Password of the OS user (on the host) who owns the Management Agent.

- **credential_name**

  Name of the saved credential.

- **credential_setname**

  Name of the credential set of the Management Agent. Example: "HostCreds".

ORACLE®

Examples

Example 1

```
emcli restart_agent -agent_name="agent.example.com:1234"
                         -host_username="test_user"
                         -host_pwd="test"
```

Example 2

```
emcli restart_agent -agent_name="agent.example.com:1234"
                         -credential_name="MyMachineCredential"
```

resume_instance

Resumes a suspended deployment instance.

Format

```
emcli resume_instance
     -instance=<instance_guid>
     [-exec=<execution_guid>]
     [-name=<execution_name>]
     [-owner=<execution_owner>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **instance**

  GUID of the instance.

- **exec**

  GUID of the execution.

- **name**

  Name of the execution.

- **owner**

  Owner of the execution.

Examples

```
emcli resume_instance -instance=16B15CB29C3F9E6CE040578C96093F61
```

resume_job

Resumes a job or set of jobs. Resumes job executions on any of the targets scheduled to start within the beginning and ending time window.

> **Note:**
>
> Suspend and resume operate either at the job or the execution level, but not both. If job executions were previously suspended, they must be resumed by execution matching. If a job was suspended, it must be resumed by job matching; it is not be possible to resume it by executions.

Format

```
emcli resume_job
    [-name="job_name_pattern"]
    [-owner="job_owner"]
    [-type="job_type"]
    [-targets="target_name:target_type"]
    [-input_file=property_file:"filename"]
    [-preview]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name or pattern of the job(s) to resume.

- **owner**

  Owner of the job(s).

- **type**

  Job type of the job(s).

- **targets**

  Target name and target type of the job(s).

- **input_file**

  Specify the filtering properties of the file in "filename."

  Any jobs matching all the specified filter criteria are resumed. You must specify at least one filter, and the logged in administrator must have the necessary privileges on the matching jobs.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **preview**

  Only lists the jobs that would be resumed.

Examples

Example 1

This example resumes all jobs of type Backup whose name starts with BK.

```
emcli resume_job -name=BK% -type=Backup
```

Example 2

This example resumes jobs or job executions matching search criteria in suspend_prop.txt.

```
resume_job -input_file=property_file:/tmp/suspend_prop.txt
```

If the same file is used for both suspend and for resume, the set of jobs or executions resumed should overlap, but might not be identical. The criteria may match more or fewer jobs or executions than previously.

resyncAgent

Performs an Agent recovery. A message is issued if the specified agent does not exist.

Format

```
emcli resyncAgent
     -agent="Agent Name"
     [-keep_blocked]
```

[ ] indicates that the parameter is optional.

Options

- agent

  Name of the Agent for which to perform Agent Recovery.

- keep_blocked

  Leaves the agent blocked even if the resync succeeds. By default the agent is unblocked after a successful resync.

Example

This example resyncs Agent XYZ.

```
emcli resyncAgent
-agent="XYZ"
```

resync_swlib_cache

Invokes resynchronization for one or all cache nodes.

Format

```
emcli resync_swlib_cache
     [ -cache_node_name="cache node name" | -all]
```

[ ] indicates that the parameter is optional.

Parameters

- cache_node_name

  The name of the specific cache node you want to resynchronize.

- all

  This is an option to indicate that all cache nodes should be resynchronized.

retry_add_host

Retries a failed add host session.

Format

```
emcli retry_add_host
        -session_name="session_name"
        -retry_using_same_inputs | -update_inputs_and_retry"
        [-host_names="host_names"]
        [-platform="platform_id"]
        [-installation_base_directory="installation_base_directory"]
        [-credential_name="credential_name"]
        [-credential_owner="credential_owner"]
        [-instance_directory="instance_directory"]
        [-port="agent_port"]
        [-deployment_type="type_of_agent_deployment"]
        [-privilege_delegation_setting="privilege_delegation_setting"]
        [-additional_parameters="parameter1 parameter2 ..."]
        [-source_agent="source_agent"]
        [-master_agent="master_agent"]
        [-preinstallation_script="preinstallation_script"]
        [-preinstallation_script_on_oms]
        [-preinstallation_script_run_as_root]
        [-postinstallation_script="postinstallation_script"]
        [-postinstallation_script_on_oms]
        [-postinstallation_script_run_as_root]
        [-wait_for_completion]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **session_name**

  Name of the session you want to retry.

- **retry_using_same_inputs**

  Retries the Add Host session using the same inputs.

- **update_inputs_and_retry**

  Updates the inputs and retries the Add Host session.

- **host_names**

  Names of the hosts where the Agents need to be installed, separated by a semicolon.

- **platform**

  ARU platform ID of the hosts where the Agent needs to be installed.

- **installation_base_directory**

  Directory where you want to install the Agent. Provide this option in double-quotes if it is an MS-DOS/Windows style path.

- **credential_name**

  Named credential to be used for installing the Agent.

- **credential_owner**

  Owner of the named credential.

- **instance_directory**

  Instance directory of the Agent. Provide this option in double-quotes if it is an MS-DOS/Windows style path.

Chapter 5
EM CLI Verbs

- **port**

  Port on which the Agent should communicate with the OMS.

- **deployment_type**

  Type of Agent deployment, which can be FRESH, CLONE, or SHARED. By default, it is the deployment type of the failed session you want to retry.

- **privilege_delegation_setting**

  Privilege delegation setting you want to use for installing an Agent and running the root script.

- **additional_parameters**

  Additional parameters you want to use for installing an Agent.

- **source_agent**

  Source Agent you want to use for installing a cloned Agent.

- **master_agent**

  Master Agent you want to use for installing a shared Agent.

- **preinstallation_script**

  Script you want to run before installing the Agent. Provide this option in double-quotes if it is an MS-DOS/Windows style path.

- **preinstallation_script_run_as_root**

  Use this option if you want to run the pre-installation script as the root user.

- **preinstallation_script_on_oms**

  Use this option if the pre-installation script resides on the OMS host.

- **postinstallation_script**

  Script you want to run after installing the Agent. Provide this option in double-quotes if it is an MS-DOS/Windows style path.

- postinstallation_script_on_oms

  Use this option if the post-installation script resides on the OMS host.

- **postinstallation_script_run_as_root**

  Use this option if you want to run the post-installation script as the root user.

- **wait_for_completion**

  Runs the Add Host operation synchronously.

Examples

Example 1

This example retries the session 'ADD_HOST_SYSMAN_Dec_17_2012_2:02:28_AM_PST' using the same inputs.

```
emcli retry_add_host session_name='ADD_HOST_SYSMAN_Dec_17_2012_2:02:28_AM_PST' -
retry_using_same_inputs
```

Example 2

ORACLE®

5-720

This example retries the session 'ADD_HOST_SYSMAN_Dec_17_2012_2:02:28_AM_PST' by updating the input port to 5678.

```
emcli retry_add_host session_name='ADD_HOST_SYSMAN_Dec_17_2012_2:02:28_AM_PST' -
update_inputs_and_retry -port=5678
```

retry_instance

Retries a failed instance or failed step.

Format

```
emcli retry_instance        [-instance=<instance_guid>]
     [-exec=<execution_guid>]
     [-name=<execution_name>]
     [-owner=<execution_owner>]
     [-stateguid=<state_guid>]

[ ]  indicates that the parameter is optional
```

Options

- **instance**

  GUID of the instance.

- **exec**

  GUID of the execution.

- **name**

  Name of the execution.

- **owner**

  Owner of the execution.

- **stateguid**

  Comma-separated list of state GUIDs.

Examples

```
emcli retry_instance -instance=16B15CB29C3F9E6CE040578C96093F61 -
stateguid=51F762417C4943DEE040578C4E087168

emcli retry_instance -instance=16B15CB29C3F9E6CE040578C96093F61 -
stateguid='51F762417C4943DEE040578C4E087168,51F762417C4944DEE040578C4E087168'
```

retry_job

Restarts a previously failed job execution.

Format

```
emcli retry_job
     -exec_id="executionID"
     [-noheader]
     [-script | -format=
           [name:<pretty|script|csv>];
           [column_separator:"column_sep_string"];
```

```
        [row_separator:"row_sep_string"];
    ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **exec_id**

  ID of the job execution to be retried. Use the get_jobs verb to obtain specific job execution IDs.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This option is equivalent to -format="name:script".

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns:

Execution ID

Examples

This example restarts the job execution with Id 123456789012345678901234569789012 and displays a new execution ID.

```
emcli retry_job -exec_id=123456789012345678901234569789012
```

revoke_bipublisher_roles

Revokes roles for accessing the BI Publisher catalog.

Format

```
emcli revoke_bipublisher_roles
      (-roles="role1[;role2;...role_n]"
      [-users="user"]
      [-external_role="group"])
```

```
[ ]  indicates that the parameter is optional
```

Options

- **roles**

  Revokes one or more roles from the BI Publisher. Specify one or more roles separated by a semicolon.

- **users**

  Users to revoke the granted role.

- **external_role**

  Name of the external group to apply the revocation.

Examples

Example 1

This example revokes one role from a group.

```
emcli revoke_bipublisher_roles -roles="EMBIPViewer" -external_role="TESTGROUPNAME"
```

Example 2

This example revokes more than one role from a group.

```
emcli revoke_bipublisher_roles -roles="EMBIPViewer;EMBIPAuthor"
      -external_role="TESTGROUPNAME"
```

revoke_license_no_validation

Revokes licenses on a set of user-specified packs, or all packs to a set of user-specified targets, or all targets belonging to the input licensable target type.

For 11g database targets, you cannot enable or disable the Database Diagnostic and Tuning Packs through the user interface. You need to set the control_management_pack_access initialization parameter to manage your licenses. For information about this option, see the Enterprise Database Management chapter of *Oracle Enterprise Manager Licensing Information*.

> **Tip:**
>
> You can use this verb to revoke licenses for standalone target types, such as hosts and databases, but you cannot use this verb to revoke licenses for the parent Application Server (oracle_ias) target type, which has dependent target types of OC4J, Jserv, Web Cache, and so forth. To do this, use the revoke_license_with_validation verb instead.
>
> For example, for pack ias_config and an Application Server target of AS1 with an associated dependent target of OC4J1, this verb revokes the license to AS1, but this does not propagate to OC4J1.

Format

```
emcli revoke_license_no_validation    -type="target_type"    [-
targets="tname1;tname2;..."]    [-packs="pack1;pack2;..."]    [-
file="file_name"]    [-displayAllMessages]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **type**

  Target type as it exists in the database. Names cannot contain colons (:), semi-colons (;), or any leading or trailing blanks. You can specify only one target type at a time; for example, -type="oracle_database".

- **targets**

  Targets should be specified in the following sequence:

  ```
  TargetName1;TargetName2;
  ```

  For example:

  ```
  -targets="database1;database2;database3;"
  ```

  The semi-colon (;) is the target separator.

  See the "Examples" section below for information about providing arguments for the targets .

- **packs**

  License packs should be specified in the following sequence:

  ```
  pack1;pack2;
  ```

  For example:

  ```
  -packs="db_diag;db_config;"
  ```

  The semi-colon (;) is the pack separator.

  See the "Examples" section below for information about providing arguments for the pack .

- **file**

  Specify the file name, including the complete path. For example:

  ```
  -file="/usr/admin1/db_license.txt"
  ```

  The file should contain the list of targets and packs according to the following cases:

  – If you only need to provide a list of targets, use the following format:

  ```
  targets=database1;database2;database3;
  ```

  – If you only need to provide a list of packs, use the following format:

  ```
  packs=db_diag;db_config;
  ```

  – If you need to provide a list of both targets and packs, use the following format:

  ```
   targets=database1;database2;database3; packs=db_diag;db_config;
  ```

- **displayAllMessages**

  Displays all messages. Only error messages are displayed by default. "=value" is not allowed on the command line.

Examples

Example 1 and Example 2 below revoke licenses of specific packs for specific targets. In order to know which target types and pack names you can pass as arguments, you can use the view named mgmt_license_view to see a list of licensable targets, their target types, and the list of packs licensed on them.

To obtain this information, do the following:

1. Access SQL*Plus with your username and password, using sysman or other user that has access to sysman.mgmt_license_view.

2. Select a distinct pack name from sysman.mgmt_license_view, where:

   ```
   target_type=<oracle_database>
   ```

This example shows pack names for an Oracle database you specify as the target type.

```
PACK_NAME
-------------------
db_config
provisioning
db_sadm
db_tuning
db_diag
provisioning_db
db_chgmgt

7 rows selected.
```

Based on this information, to revoke a license to the database1 target for the db_chgmgt pack, you would enter the following command:

```
emcli revoke_license_no_validation -type="oracle_database" -targets="database1" -
packs="db_chgmgt"
```

The only limitation of mgmt_license_view is that it only lists the packs for a target type where the pack is granted to at least one target of that type. That is, if the pack is not granted to any target of that type, mgmt_license_view cannot provide any information.

Example 1

This example revokes the license of the db_diag and db_config packs to database1, database2, and database3 targets (oracle_database target type):

```
emcli revoke_license_no_validation -type="oracle_database"            -
targets="database1;database2;database3;" -packs="db_diag;db_config;"
```

Example 2

This example revokes the license of the db_diag and db_config packs to all database targets in the setup:

```
emcli revoke_license_no_validation -type="oracle_database"
       -packs="db_diag;db_config;"
```

revoke_license_with_validation

Revokes licenses on a set of user-specified packs, or all packs to a set of user-specified targets, or all targets belonging to the input licensable target type as per business rules.

For 11g database targets, you cannot enable or disable the Database Diagnostic and Tuning Packs through the user interface. You need to set the control_management_pack_access initialization parameter to manage your licenses. For information about this option, see the Enterprise Database Management chapter of *Oracle Enterprise Manager Licensing Information*.

> 💡 **Tip:**
>
> You can use this verb to revoke licenses for standalone target types, such as hosts and databases, and you also use this verb to revoke licenses for the parent Application Server (oracle_ias) target type, which has dependent target types of OC4J, Jserv, Web Cache, and so forth.
>
> For example, for pack ias_config and an Application Server target of AS1 with an associated dependent target of OC4J1, this verb revokes the license to AS1 and also propagates to OC4J1 (and all other dependent targets associated with AS1).
>
> To revoke licenses for only standalone target types, use the revoke_license_no_validation verb.

Format

```
emcli revoke_license_with_validation    -type="target_type"    [-
targets="tname1;tname2;..."]    [-packs="pack1;pack2;..."]    [-
file="file_name"]    [-displayAllMessages]

[ ]  indicates that the parameter is optional
```

Options

*   **type**

    Target type as it exists in the database. Names cannot contain colons (:), semi-colons (;), or any leading or trailing blanks. You can specify only one target type at a time; for example, -type="oracle_database".

*   **targets**

    Targets should be specified in the following sequence:

    ```
    TargetName1;TargetName2;
    ```

    For example:

    ```
    -targets="database1;database2;database3;"
    ```

    The semi-colon (;) is the target separator.

    See the "Examples" section below for information about providing arguments for the targets .

*   **packs**

    License packs should be specified in the following sequence:

    ```
    pack1;pack2;
    ```

    For example:

```
-packs="db_diag;db_config;"
```

The semi-colon (;) is the pack separator.

See the "Examples" section below for information about providing arguments for the packs.

- **file**

    Specify the file name, including the complete path. For example:

    ```
    -file="/usr/admin1/db_license.txt"
    ```

    The file should contain the list of targets and packs according to the following cases:

    – If you only need to provide a list of targets, use the following format:

    ```
    targets=database1;database2;database3;
    ```

    – If you only need to provide a list of packs, use the following format:

    ```
    packs=db_diag;db_config;
    ```

    – If you need to provide a list of both targets and packs, use the following format:

    ```
    targets=database1;database2;database3; packs=db_diag;db_config;
    ```

- **displayAllMessages**

    Displays all messages. Only error messages are displayed by default. "=value" is not allowed on the command line.

Examples

Example 1 and Example 2 below revoke licenses of specific packs for specific targets. In order to know which target types and pack names you can pass as arguments, you can use the view named mgmt_license_view to see a list of licensable targets, their target types, and the list of packs licensed on them.

To obtain this information, do the following:

1. Access SQL*Plus with your username and password, using sysman or other user that has access to sysman.mgmt_license_view.

2. Select a distinct pack name from sysman.mgmt_license_view, where:

    ```
    target_type=<oracle_database>
    ```

This example shows pack names for an Oracle database you specify as the target type.

```
PACK_NAME
------------------
db_config
provisioning
db_sadm
db_tuning
db_diag
provisioning_db
db_chgmgt

7 rows selected.
```

Based on this information, to revoke a license to the database1 target for the db_chgmgt pack, you would enter the following command:

```
emcli revoke_license_with_validation -type="oracle_database" -
targets="database1" -packs="db_chgmgt"
```

The only limitation of mgmt_license_view is that it only lists the packs for a target type where the pack is granted to at least one target of that type. That is, if the pack is not granted to any target of that type, mgmt_license_view cannot provide any information.

Example 1

This example revokes the license of the db_diag and db_config packs to database1, database2, and database3 targets (oracle_database target type):

```
emcli revoke_license_with_validation -type="oracle_database"          -
targets="database1;database2;database3;" -packs="db_diag;db_config;"
```

Example 2

This example revokes the license of the db_diag and db_config packs to all database targets in the setup:

```
emcli revoke_license_with_validation -type="oracle_database"
      -packs="db_diag;db_config;"
```

revoke_privs

Revokes the privileges from an existing Enterprise Manager user or Enterprise Manager role.

Format

```
emcli revoke_privs
        -name="username|rolename"
        [-privilege="name[;secure-resource-details]]"
        [-separator=privilege="sep_string"]
        [-subseparator=privilege="subsep_string"]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  User name or role name from which privileges will be revoked.

- **privilege**

  Privilege to grant to this administrator. You can specify this option more than once. The original administrator privileges will be revoked. Specify <secure_resource_details> as:

  ```
  resource_guid|
  [resource_column_name1=resource_column_value1[:resource_column_name2=resource
  _column_value2]..]"
  ```

- **separator**

  Specify a string delimiter to use between name-value pairs for the value of the -privilege option. The default separator delimiter is a semi-colon (;).

- **subseparator**

Specify a string delimiter to use between name and value in each name-value pair for the value of the -privilege option. The default subseparator delimiter is a colon (:).

Examples

Example 1

For user1, This example revokes full control of the jobs with ID 923470234ABCDFE23018494753091111, and revokes full control on the target host1.example.com:host:

```
emcli revoke_privs
        -name="user1"
        -privilege="FULL_JOB;923470234ABCDFE23018494753091111"
        -privilege="FULL_TARGET;host1.example.com:host"
```

Example 2

This example revokes the target privileges from Enterprise Manager role Role1:

```
emcli revoke_privs
        -name="Role1"
        -privilege="FULL_TARGET;host1.example.com:host"
```

revoke_quota

Revokes quota that has been assigned.

Format

Standard Mode

```
emcli revoke_quota
      -assignee_name="assignee name"
      -assignee_type="assignee type"
      -quota="quota"]
      [-force][ ]  indicates that the parameter is optional
```

Interactive or Script Mode.

```
emcli revoke_quota(
      assignee_name="assignee name"
      assignee_type="assignee type"
      [,quota="quota"]       [,force=True/False])
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   assignee_name

    The assignee to which the quota has been assigned.

*   assignee_type

    The type of assignee. Valid values are: Tenant, User_Profile, or User.

*   quota

    The quota that will be revoked for the assigned user. Quota allocations can be defined:

    –   at the entity level.

– on a quota assignable object if the selected entity supports quota allocation on objects. When quota is allocated on an object, all quota related computations will be done in the context of that object.

- force

   If an assignee has resource objects allocated itself at the component or entity level, quota cannot be revoked for the assignee. When a quota revoke request is made, a warning message is displayed. The administrator can either revoke all the quota or use the -force parameter to ignore the warning message and revoke the quota.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

This section contains the following examples.

Example 1

The following example revokes quota that has been assigned at the entity level.

```
emcli revoke_quota
      -assignee_name=T1
      -assignee_type=tenant
      -quota=Component1:Entity1

emcli revoke_quota
      -assignee_name=U1
      -assignee_type=user
      -quota=Component1:Entity1

emcli revoke_quota
      -assignee_name=UP1
      -assignee_type=user_profile
      -quota=Component1:Entity1
```

Example 2

The following example revokes quota that has been assigned to `OBJ1` object.

```
emcli revoke_quota
      -assignee_name=T1
      -assignee_type=tenant
      -quota=Component1:Entity1;object_name=OBJ1:object_type=type1
```

revoke_roles

Revokes the roles to an existing Enterprise Manager user or Enterprise Manager role.

Format

```
emcli revoke_roles
      -name="username|rolename"
      [-roles="role1;role2;..."]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  User name or role name from which roles will be revoked.

- **roles**

  Roles, which will be revoked from the Enterprise Manager user or role. You can specify this option more than once.

Examples

```
emcli revoke_roles
        -name="user1"
        -roles="SUPER_USER"

emcli revoke_roles
        -name="Role1"
        -roles="BLACKOUT_ADMIN;MAINTAIN_TARGET"
```

run_avail_diag

Runs diagnostics for an availability algorithm for a test-based service. This is mostly useful when the "last calculated" time stamp is running behind the current time, and the service status has been unresponsive for some time.

Format

```
emcli run_avail_diag
      -name=<target_name>
      -type=<target_type>
```

Options

- **name**

  Service target name.

- **type**

  Service target type.

Examples

```
emcli run_avail_diag  -name='MyTarget' -type='generic_service'
```

run_auto_discovery

Run auto discovery on specified hosts.

Format

```
emcli run_auto_discovery
    -host="host1.example.com;host2.example.com;..."
    [-debug]
    [-separator="separator:attribute_name:character"]
    [-subseparator="subseparator:attribute_name:character"]
```

Options

- **-host**

  List of hosts where auto discovery needs to be run.

- **-debug**

  Generate debug messages.

- **-separator**

  By default multi-value input attributes use the semicolon (;) character as separator.Specifying this option overrides the default separator value.

  Example: separator="<attribute_name=sep_char>"

  Where `attribute_name` is name of the attribute for which you want to override separator character and `sep_char` is new separator character

- **-subseparator**

  By default multi-value input attributes use the colon (:) character as the sub-separator. Specifying this option overrides the default sub-separator value.

  Example: subseparator="<attribute_name=sep_char"

  Where `attribute_name` is name of the attribute for which you want to override separator character and `sep_char` is new sub-separator character.

Examples

Following command will run auto discovery on host named host01.example.com.

```
emcli run_auto_discovery -host="host01.example.com"
```

Following command will run auto discovery on hosts named host01.example.com and host02.example.com.

```
emcli run_auto_discovery -host="host01.example.com;host02.example.com"
```

run_config_history

Runs the configuration history saved search and displays the results.

Format

```
emcli run_config_history
      -name="<Saved History Search Name>"
      [-change_category="<change_category>"]
      [-output_file="<Filename> "]
      [-mode="<Display mode> "]
      [-format=" <output_format>"]
      [-no_header]
```

Options

- name

  Name of the configuration history saved search. Allowed values:

  - Exact name of the configuration history saved search.

- change_category

  Displays the change category of Configuration changes, Relationship changes, or both.

  – C

  – R

  – B

  Default is B.

- output_file

  Absolute path name of the file where the output or results are exported. If this is not present then the output is displayed in the console. If this is not present, then the output mode is based on the internal filter settings.

- mode

  Display mode of the output. If this parameter is not specified, then the output mode is based on the internal filter settings.

  – Grouped

  – ShowAll

- format

  Specifies the format. Allowed values:

  – format="name:pretty" - Prints out the output table in a readable format.

  – format="name:script" - Sets the default column separator to a tab and the default row separator to a new line.

  – format="name:csv" - Sets the column separator to a comma and the row separator to a new line.

  Default is format="name:pretty".

- no_header

  Displays a tabular output without the column headers.

Examples

Example 1

The following command shows the result of the history configuration search named "Host History" and exports it to the file "/home/HostHistoryFile.out" in a tabular format.

```
emcli run_config_history
      -name="Host History"
      -output_file="/home/HostHistoryFile.out"
```

Example 2

The following command shows the result of the history configuration search named "Host History" in a ShowAll mode in a tabular mode without the headers.

```
emcli run_config_history
      -name="Host History"
      -mode="ShowAll"
      -format="name:pretty"
      -no_header
```

run_config_search

Runs a configuration search using a specified search name.

Format

```
emcli run_config_search
      -search_name="<Configuration Search UI Name>"
      [-target_name="<target name>"]
      [-on_host="<hostname>"]
      [-memberof="<group name>"]
      [-output_file="<output file name>"]
      [-format=name:<pretty|script|csv>;
      [column_separator:"column_sep_string"];
      [row_separator:"row_sep_string"];

[ ]  indicates that the parameter is optional.
```

Options

- search_name

  A display name for the configuration search.

- target_name

  Name of the target. It can be a full value or a pattern match using "%".

- on_host

  Name of the host where the target is running. It can be a full value or a pattern match using "%".

- memberof

  Name of the file with an absolute path where the output or results will be exported. If this parameter is not specified then the output is displayed in the console.

- output_file

  Name of the file with an absolute path where the output or results will be exported. If this is not present then the output is displayed in the console.

- format

  Format specification (default is -format="name:pretty").

  – format="name:pretty" prints the output table in a readable format not intended to be parsed by scripts.

  – format="name:script" sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings may be specified to change these defaults.

  – format="name:csv" sets the column separator to a comma and the row separator to a newline.

    format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

    – format="name:script;column_separator:<column_sep_string>" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

    – format="name:script;row_separator:<row_sep_string>" row-separates the verb output by <row_sep_string>. Columns are separated by the tab character.

- column_separator

  Specifies the column separator.

- row_separator

  Specifies the row separator.

Examples

Example 1

The following command shows the results of the search named "Search File Systems on Hosts" where the target name starts with "oracle":

```
emcli run_config_search
     -search_name="Search File Systems on Hosts"
     -target_name="oracle%"
```

Example 2

The following command shows the results of the search named "Search File Systems on Hosts" where the target name starts with "oracle" and the host name containing the pattern "host" are members of the group "group1":

```
emcli run_config_search
     -search_name="Search File Systems on Hosts"
     -target_name="oracle%"
     -on_host="%host%"
     -memberof="group1"
```

run_fa_diagnostics

Runs diagnostics checks to identify issues related to discovery, monitoring, and other features.

Format

```
emcli run_fa_diagnostics
     -input_file=fa_domain_discovery_file:file_path
     [-input_file=host_agent_mapping_file:file_path]
     [-input_file=pf_domain_cred_mapping_file:file_path]
     [-debug]

[ ]  indicates that the parameter is optional
```

Options

- **input_file**

  Fully-qualified path to a CSV-formatted file containing one line of details for each Fusion Instance to be added. The valid Fusion Instance WebLogic Server version is 10.

  The structure of the CSV file for WebLogic Server version 10.x and above is as follows:

```
<WebLogic Server version>,
<Administration Server host machine name>,
<Administration Server listen port>,
<Administration Server username>,
<Administration Server password>,
<External Options - optional>,
<JMX Protocol - required only if SSL is enabled>,
<JMX Service URL - required only if SSL is enabled>,
<Unique Domain Identifier>,
<Agent URL>,
<Discover Down Servers - optional - Default if not specified is false>,
<Use Same Credentials for All Domains in the Fusion Instance - optional -
Default if not specified is true>,
<Discover Application Versions - optional - Default if not specified is true>
```

For example:

```
fa1-CRM,weblogic,welcome1,
fa1-FIN,weblogic1,welcome2,
fa2-CRM,weblogic,welcome.host.example.com
```

- **debug**

  Runs the verb in verbose mode for debugging purposes.

Examples

This example reads the my_domains_info.csv file to determine the Fusion Instances to run diagnostic checks, reads the my_agent_mapping.csv file to determine which Management Agents to use for running discovery tests, and reads the my_domain_cred_mapping.csv file to determine which credentials to use to discover the individual product family.

```
emcli run_fa_diagnostics
-input_file=fa_domain_discovery_file:c:\emcli\my_domains_info.csv
-input_file=host_agent_mapping_file:c:\emcli\my_agent_mapping.csv
-input_file=pf_domain_cred_mapping_file:c:\emcli\my_domain_cred_mapping.csv
```

run_mda_health_check

Runs a health check job for the Middleware Diagnostics Advisor (MDA). This job verifies and fixes any issues in the MDA framework. It also validates and enables any new finding types registered with MDA. All applicable targets that are not yet enabled are also enabled.

Format

```
emcli run_mda_health_check
```

Example

The following example runs an MDA health check job:

```
emcli run_mda_health_check
```

run_prechecks

Submits the pre-check operation for any given operation plan.

Format

```
emcli run_prechecks
      -operation_plan=<operation_plan_name>
```

Options

- **operation_plan**

  Name of the operation plan.

Examples

```
emcli run_prechecks
      -operation_plan="BISystem1-switchover"
```

run_prerequisites

Runs a list of Enterprise Manager repository-related prerequisites.

Format

```
emcli run_prerequisites
      -db_user=<database_user>
      -db_password=<database_password>
      -db_role=<database_role>
      -repos_user=<repository_user>
      [-prerequisite_xml_root_dir=<xml_root_directory_for_platform_prerequisites>]
      [-prerequisite_resource_locs="<xml_resource_location_for_platform/
      plug-in_prerequisites>"]
      [-log_loc=<location_for_log_files_of_EMPrereqKit_tool>]
      [-upgrade_version=<EM_version_to_which_upgrade_is_being_done_eg_12.1.0.3>]
      [-configuration_type=<configuration/deployment_type_
       eg_MINI/SMALL/MEDIUM/LARGE>]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- db_user

  Database user account with which a connection to the database can be established, for example SYS.

- db_password

  Database user account password. If you do not provide here, you will be prompted for the password.

- db_role

  Database role. For example, sysdba. Required only when the -db_user value is SYS.

- repos_user

  Repository user account with which the prerequisite checks can be run, for example, SYSMAN. Required only when the -db_user value is SYS.

- prerequisite_xml_root_dir

  Absolute path to the `requisites/list` directory where of the all prerequisite XMLs are located. This is an optional parameter and if not provided, the value is calculated

internally. The XML files can be in a subdirectory within the `requisites/list` directory, but make sure the path that you enter leads only up to the `list` directory. For example, `$<OMS_HOME>/install/requisites/list`.

- prerequisite_resource_locs

  Absolute path to the directory where the plug-in opar files or the platform/plug-in binaries, which contains XML files for platform or plug-in prerequisite checks, are located. This option is not mandatory. For plug-in opar files, use the format `plugin_id=<<absolute_path_.opar_file>>`. For the plug-in home directory use the format `plugin_id=<<plugin_home>>`.

- log_loc

  Absolute path to a directory where the logs of the execution of the Enterprise Manager prerequisite kit can be stored.

- upgrade_version

  The Enterprise Manager version to which the upgrade is being done. For example, 12.1.0.3. If you have downloaded the Enterprise Manager prerequisite resources for two future versions, for example v1 and v2 through Self-Update then with -upgrade_version, you can see or run the prerequisite of the specified version.

- configuration_type

  Configuration or deployment type. For example, MINI, SMALL, MEDIUM, LARGE. This is an optional parameter, and if not provided, it will be calculated internally.

Examples

Example 1

Runs a list of Enterprise Manager repository-related prerequisites with the configuration type MEDIUM.

```
emcli list_prerequisites
    -db_user=SYS
    -db_password=pwd
    -db_role=sysdba
    -repos_user=SYSMAN
    -prerequisite_xml_root_dir=$ORACLE_HOME/install/requisites/list
    -configuration_type=MEDIUM
```

Example 2

Runs a list of Enterprise Manager repository-related prerequisites with the prerequisite resource location `oracle.sysman.db=<<MW_HOME>>/plugins/oracle.sysman.db.oms.plugin_x.x.x.x.x,oracle.sysman.emas=<<Absolute directory path>>/x.x.x.x.x_oracle.sysman.emas_2000_0.opar`.

```
emcli list_prerequisites
    -db_user=SYS
    -db_password=pwd
    -db_role=sysdba
    -repos_user=SYSMAN
    -prerequisite_resource_locs="oracle.sysman.db=
    <<MW_HOME>>/plugins/oracle.sysman.db.oms.plugin_x.x.x.x.x,
    oracle.sysman.emas=<<Absolute directory path>>/
    x.x.x.x.x_oracle.sysman.emas_2000_0.opar"
```

run_promoted_metric_diag

Runs promoted metric diagnostics.

Format

```
emcli run_promoted_metric_diag
       -name=<target_name>
       -type=<target_type>
       -promotedMetricName=<metric_name>
       -promotedColumn=<metric_type>
```

Options

- **name**

  Service target name.

- **type**

  Service target type.

- **promotedMetricName**

  Promoted metric name.

- **promotedColumn**

  Promoted metric type.

Examples

```
emcli run_promoted_metric_diag  -name='MyTarget' -type='generic_service'
-promotedMetricName='metric1' -promotedColumn='Performance'
```

save_latest_config

Saves the latest configuration of a target.

Format

```
emcli save_latest_config
      -name="Host Config"
      -target_type="host"
      -target_name="test_host"
      [-description="Save latest configuration of host"]

[ ]  indicates that the parameter is optional.
```

Options

- name

  Name of the configuration which is being saved. The value should be unique and non null.

- target_type

  Target type for which the configuration is being saved. The value should be the internal name.

- target_name

ORACLE®

Name of the target.

- description

    Description of the target's saved configuration. This option is not mandatory.

Example

The following command saves the latest configuration for the host target "test_host" with the description "Save latest configuration of host" and the name of "Host Config":

```
emcli save_latest_config
      -name="Host Config"
      -target_type="host"
      -target_name="test_host"
      -description="Save latest configuration of host"
```

save_masking_script

Saves a masking script already generated to the specified path or file.

Format

```
emcli save_masking_script
      -definition_name=<masking_definition_name>
      [-path=file path]
      [-file=file name]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **definition_name**

    Masking definition name.

- **path**

    Path for the file name to save the masking script. File name is automatically generated. The path and file options are mutually exclusive. Only an absolute path is allowed.

- **file**

    File name to save the masking script. The file name must include the absolute path. Either the path or file option must be specified.

Output

Success or error messages

Examples

Example 1

This example saves the masking script for the definition named mask_hr_data to the /tmp directory:

```
emcli save_masking_script
      -definition_name=mask_hr_data
      -path=/tmp/
```

Example 2

This example saves the masking script for the definition named mask_hr_data to /tmp/abc.sql :

```
emcli save_masking_script
      -definition_name=mask_hr_data
      -file=/tmp/abc.sql
```

save_metric_extension_draft

Save a deployable draft of a metric extension. The metric extension must currently be in an editable state. Once saved as a draft, the metric extension is no longer editable.

Format

```
emcli save_metric_extension_draft
      -target_type=<metric_extension_target_type>
      -name=<metric_extension_name>
      -version=<metric_extension_version>
```

Options

• **target_type**

Target type of the metric extension.

• **name**

Name of the metric extension.

• **version**

Version of the metric extension to be saved to the draft.

save_procedure_input

Configures a deployment procedure for execution.

Format

```
emcli save_procedure_input
      [-name="procedure_configuration_name"]
      [-owner="procedure_configuration_owner"]
      [-procedure="procedure_guid"]
      -input_file="file_path\file_name"
      [-grants="access_levels_for_users"]
      [-schedule=
          start_time:yyyy/MM/dd HH:mm;
          tz:{java timezone ID};
          grace_period:xxx;
      ]
      [-notification="procedure status"]
```

```
[ ]  indicates that the parameter is optional
```

Options

• **name**

Name of the configuration for the procedure.

- **owner**

  Owner of the Procedure configuration.

- **procedure**

  GUID of the procedure to execute.

- **input_file**

  GUID of the procedure to execute. The file_path should point to a file containing the data property file.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **grants**

  Specifies users and their corresponding access levels as a string of user:privilege pairs, each separated by a semi-colon (;). The user is an Enterprise Manager user name, and the privilege is either VIEW_JOB or FULL_JOB.

  See the example below.

- **schedule**

  Schedule for the deployment procedure. If not specified, the procedure is executed immediately.

  - **start_time** — When the procedure should start.

  - **tz** — Optional timezone ID.

  - **grace_period** — Optional grace period in minutes.

- **notification**

  Status of the procedure.

Example

```
emcli save_procedure_input
   -name=configProcedure -procedure=16B15CB29C3F9E6CE040578C96093F61
   -input_file=/home/data.properties -grants="user1:VIEW_JOB;user2:FULL_JOB"
   -schedule="start_time:2011/8/21 21:23;tz:America/New_York;grace_period:60"
   -notification="scheduled, action required, running"
```

schedule_siteguard_health_checks

Schedules health checks for an operation plan. Optionally, configured users can be notified about scheduled health-check reports.

Format

```
emcli schedule_siteguard_health_checks
   -operation_plan=[name_of_the_operation_plan]
   -schedule=
           start_time:yyyy| MM | dd HH:mm;
           [tz:"java timezone ID";]
           [frequency:interval | weekly | monthly| yearly;]
           [repeat:tx;]
           [end_time:yyyy | MM | dd HH:mm;]
           [grace_period:xxx;]
   [-notify="true" | "false"]
   [-
```

```
email="email_address_to_be_notified"]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **operation_plan**

    Name of the operation plan for which health checks must be scheduled.

*   **schedule**

    Time when health checks need to run. The possible values for this option are:

    –   **start_time**

        Date and time when health checks need to be executed.

    –   **tz**

        Time zone ID to run health checks.

    –   **frequency**

        Frequency at which you want to execute health checks. The valid values for this option are `once`, `interval`, `weekly`, `monthly`, and `yearly`. If the frequency is set to `interval`, then the values for the parameter `repeat` must be specified. If the frequency is set to `weekly` or `monthly`, then the days when the health check needs to be executed must be specified. If frequency is set to `yearly`, then both days and months when the health checks need to be executed must be specified.

    –   repeat

        Frequency of repetition of the health checks. You need to enter the values for this option only if the frequency is set to 'interval'. You need to specify one of the following values for this option:

        *   days

            Enter the list of days that the health checks need to be executed for the specified operation plan. Use commas to separate the items in the list. This value is required only if the frequency is set to `weekly`, `monthly`, or `yearly`. If frequency is set to `weekly`, then the valid range is `1` to `7`. If the frequency is set to `monthly` or `yearly`, then the valid range is `1` to `30`.

        *   months

            Enter the list of months that the health checks need to be executed for the specified operation plan. Use commas to separate the items in the list. This value is required only if the frequency is set to `monthly`. If the frequency is set to `monthly`, then the valid range is `1` to `12`.

    –   end_time

        Time when the health check should end. This option must option is optional. If the values for this option are not specified, the health checks run indefinitely.

    –   grace_period

        Values of the grace period for the health check scheduled for the specified operation plan. Enter the values in minutes.

*   **notify**

    Emails the health check reports to the configured users. If this option is set to `true`, then the configured users receive an email notification of the health-check execution report.

- **email**

  Email address of the configured users who should be notified about the health-check reports. The email addresses specified need to be those of registered users.

Examples

Example 1

This example schedules a health check for the `austin-switchover` operation plan to start on `2014/10/29` at `2:00 a.m.` and to run daily. The example also notifies the configured user by sending an email to `admin@example.com`:

```
emcli schedule_siteguard_health_checks
        -operation_plan="austin-switchover"
        -schedule="start_time:2014/10/29 2:00;frequency:interval;repeat:1d"
        -notify="true"
        -email="admin@example.com"
```

Example 2

This example schedules a health check for the `austin-failover` operation plan to start on `2014/08/10` at `1:00 a.m.,` `New York` timezone. The example also schedules the health check to run on Saturday and Sunday of every week, with a grace period of 60 minutes:

```
emcli schedule_siteguard_health_checks
        -operation_plan="austin-failover"
        -schedule="start_time:2014/08/10
01:00;frequency:weekly;days:6,7;grace_period:60;tz:America/New_York"
```

search_patches

Searches patches from the ARU site or software library with the specified search criteria.

Format

```
emcli search_patches
        [-swlib]
        [-patch_name="patch_name"]
        [-product="product_id" [-include_all_products_in_family]]
        [-release="release_id"]
        [-platform="platform_id" | -language="language_id"]
        [-type="patch | patchset"]
        [-noheader]
        [-script | -xml | -format=
                                [name:<pretty|script|csv>];
                                [column_separator:"column_sep_string"];
                                [row_separator:"row_sep_string"];
        ]

[ ]  indicates that the parameter is optional
```

Options

- **swlib**

Searches patches in the software library if this option is provided, whether the current connection mode is online or offline.

- **patch_name**

  Patch name, number, or Sun CR ID. This option is only valid in Simple Search mode. If you provide this option, the Simple Search mode is enabled. If the options specific to Advanced Search mode are provided along with this option, they will not take effect.

- **product**

  Patch product/product family ID. Run the command "emcli list_aru_products" to search the product ID.

- **include_all_products_in_family**

  Takes the specified product ID as a product family ID and includes all products in this product family while searching patches. This option is valid only when you provide the 'product' option.

- **release**

  Patch release ID. Run the command "emcli list_aru_releases" to search for the release ID.

- **platform**

  Patch platform ID. Run the command "emcli list_aru_platforms" to search for the platform ID.

- **language**

  Patch language ID. Run the command "emcli list_aru_languages" to search for the language ID.

- **type**

  Patch type.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This option is equivalent to -format="name:script".

- **xml**

  Displays the patch information in XML format.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

– row_separator:"row_sep_string" row-separates the verb output by
<row_sep_string>. Rows are separated by the tab character.

Examples

```
emcli search_patches -patch_name=6880880  -platform=226 -swlib

emcli search_patches -patch_name=6880880  -platform=226 -language=0 -xml

emcli search_patches -product=9480  -release=80102030  -platform=226 -type=patch
-format=name:pretty

emcli search_patches -product=9480  -release=80102030 type=patch -xml

emcli search_patches -product=9480  -release=80102030 -script

emcli search_patches -product=9480 -release=80102030 type=patchset -
format=name:csv
```

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
set_connection_mode
set_patch_plan_data
show_patch_plan
submit_patch_plan

upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management
Administrator's Guide*.

secure_agent

Secures an Agent.

Format

```
emcli secure_agent
       -agent_name="agent_target_name"
       -registration_pwd="registration_password"
       [-host_username ="agent_host_username" -host_pwd="agent_host_password"]
       [-credential_name ="credential_name"]
       [-credential_setname ="credential_setname_of_agent"]

[ ] inicates that the paramter is optional
```

Options

- **agent_name**

  Name of the Agent target.

- **regustration_pwd**

  Registration password to secure the communication with OMS.

- **host_username**

  User name of the OS user (on the host) who owns the Agent.

- **host_pwd**

  Password of the OS user (on the host) who owns the Agent.

- **credential_name**

  Name of the saved credential.

- **credential_setname**

  Name of the credential set of the Agent. Example: "HostCreds".

Examples

Example 1

```
emcli secure_agent -agent_name="agent.example.com:1234"
                   -registration_pwd="test_pwd"
                   -host_username="test_user"
                   -host_pwd="test"
```

Example 2

```
emcli secure_agent -agent_name="agent.example.com:1234"
                   -registration_pwd="test_pwd"
                   -credential_setname="HostCreds"
```

secure_agents

Secures Agents by providing a list of Agent names, a group name, and input file. If a group name is provided, Enterprise Manager resolves this to a list of Agents that monitor targets in this group. You can also provide an Agent list with an input file to this EM CLI command. For all of these options, you must provide either a user name or password, or the user must have been configured with preferred credentials on Agent targets. This verb submits a job with the list of Agents and the credentials provided as input, and outputs the Job Name and Job ID that you can use to track the status of the job.

This verb also calculates the list of Agents to resecure by filtering out invalid Agents, Agents that are not secure, Agents that are down, and Agents that already have an active job execution. This verb also filters out Agents that are already secured by the correct CA, but you can disable this particular filter by using the -disable_ca_check option .

Format

```
emcli secure_agents
      [-agt_names="agt1;agt2;..."] [-agt_names_file="<file>"]
      [-group_name="group_name"]
      [-use_pref_creds]
```

```
            [-username="username"]
            [-password="password"]
            [-disable_ca_check]
```

```
[ ] incicates that the paramter is optional
```

Options

- **agt_names**

  Semicolon-separated list of Agent names.

- **agt_names_file**

  Absolute path of file containing list of Agent names, each on a new line.

- **group_name**

  Identifies the list of Agents to secure. Enterprise Manager resolves the list of Agents that monitor (not just members of the group) the list of targets in the group.

- **use_pref_creds**

  Uses preferred credentials configured for the Agent to execute the secureAgent job.

- **username**

  User name to execute the secureAgent job at the Agent.

- **password**

  User password to execute the secureAgent job at the Agent.

- **disable_ca_check**

  Disables the check to verify if the Agents are secured with the latest CA.

Examples

```
emcli secure_agents -agt_names="agent_host1:1831;agent_host2:3872" -
use_pref_creds
```

```
emcli secure_agents -agt_names="agent_host1:1831;agent_host2:3872" -
username=oracleagt
```

```
emcli secure_agents -agt_names_file=/tmp/agents_list.txt -use_pref_creds
```

```
emcli secure_agents -agt_names_file=/tmp/agents_list.txt -username=oracleagt
```

send_system_broadcast

Sends a message of up to 200 characters to specified users or all users logged in to Enterprise Manager.

Format

```
emcli send_system_broadcast
        -toOption="ALL|SPECIFIC"
      [-to="comma separated user names"]
      [-messageType="INFO|CONF|WARN|ERROR|WARNING" (default is INFO)]
      -message="message details"
```

```
[ ]  indicates that the parameter is optional.
```

Options

- toOption

  Enter the value ALL to send to all users logged into the Enterprise Manager UI or enter SPECIFIC to send message to users specified in the -to option.

- to

  Comma separate list of users. This is only used if the value of the -toOption option is SPECIFIC.

- messageType

  Type of messages. Message can be one of following types: INFO|CONF|WARN|ERROR| WARNING

- message

  Message to be sent.

Example

The following example displays the custom message "EM will be taken down in an hour for an emergency patch" on every screen in the Enterprise Manager Cloud Control console:

```
emcli send_system_broadcast
    -messageType="INFO"
    -toOption="ALL"
    -message="EM will be taken down in an hour for an emergency patch"
```

set_agent_property

Modifies a specific Management Agent property. You can use this command if you have operator privilege for the Management Agent.

Format

```
emcli set_agent_property      -agent_name="<agent_target_name>"      -
name="<agent_property_name>"
    -value="<agent_property_value>"
    [-new]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **agent_name**

  Name of the Management Agent target.

- **name**

  Name of the Management Agent property you want to modify.

- **value**

  New value for the Management Agent property.

- **new**

  Denotes whether this is a new Agent property being added.

Examples

Example 1

This example sets the value of the UploadInterval property in emd.properties to 15.

```
emcli set_agent_property -agent_name="agent.example.com:1234"
     -name=UploadInterval
     -value=15
```

Example 2

This example sets the value of new property 'newprop' in emd.properties to 15.

```
emcli set_agent_property -agent_name="agent.example.com:1234"
     -name=newprop
     -value=15
     -new
```

set_availability

Changes the availability definition of a given service.

Format

```
 emcli set_availability        -name=<target_name>        -
type=<target_type>       -availType=TESTS|SYSTEM|SUB_SERVICE       -
availOp=and|or
        [-sysAvailType=SYSTEM_TARGET_DIRECTLY|SELECTED_COMPONENTS_OF_A_SYSTEM]
        [-keycomponents=<'keycomp1name:keycomp1type;
          keycomp2name:keycomp2type;...'>]
```

Options

- **name**

  Service target name.

- **type**

  Service target type. Aggregate services target type are also supported. Use the
  get_targets verb to get the target type of a target.

- **availType**

  Type of availability. Switches the availability to either test-based, system-based, or
  subservice-based. SUB_SERVICE is supported only for aggregate services.

- **availOp**

  If `and`, it uses all key tests/components to decide availability.

  If `or`, it uses any key tests/components to decide availability.

- **sysAvailType**

  Type of availability when the availType is system-based. Sets the availability to
  either SYSTEM_TARGET_DIRECTLY or
  SELECTED_COMPONENTS_OF_A_SYSTEM .

  – If availability is set to 'system target directly', the system associated with the
    service needs to define availability[status], systemname, and systemtype are
    required arguments.

- – If availability is set to 'selected components of a system', systemname, systemtype, and keycomponents are required arguments.

- – If availability is set to 'system target directly', and if availability[status] is not defined, the availability set is invalid. Therefore, the only option that can be set is 'selected components of a system'.

- **keycomponents**

  Name-type pair (that is, keycomp_name:keycomp_type) list of key components in the system used for the service.

Examples

Example 1

This example sets the availability of service MyTarget to be based on any key components of a system.

```
emcli set_availability -name='MyTarget' -type='generic_service'
                       -availType='system' -availOp='or'
                       -keycomponents='database:oracle_database; host1:host'
```

Example 2

This example sets the availability of service MyTarget to be based on system targets availability.

```
emcli set_availability -name='MyTarget' -type='generic_service'
                       -availType='system' -availOp='and'
                       -sysAvailType='system target directly'
emcli set_availability -name='MyTarget' -type='generic_service'
                       -availType='system' -availOp='and'
                       -sysAvailType='selected components of a system'
                       -keycomponents='database:oracle_database; host1:host'
emcli set_availability -name='MyTarget' -type='generic_service'
                       -availType='system' -availOp='or'
                       -sysAvailType='selected components of a system'
                       -keycomponents='database:oracle_database; host1:host'
```

set_awr_cred

Sets the database and host credential for the AWR Warehouse specified by the target name and target type parameters.

Format

```
emcli set_awr_cred
        -target_name=sample_database
        -target_type=oracle_database
        [-db_cred=<database_named_credential>]
        [-host_cred=<database_host_named_credential>]
[ ] indicates that the parameter is optional
```

Options

- target_name

  Name of the target AWR Warehouse database.

- target_type

Type of target. The possible values for target type are `oracle_database`, `oracle_pdb`, and `rac_database`.

- db_cred

  New named credential for the target database.

- host_cred

  New named credential for the target database host.

Output

Success/Error

Example

The following example sets the database and host credential for the target AWR Warehouse database, `sample_database`:

```
emcli set_awr_cred
          -target_name=sample_database
          -target_type=oracle_database
          -db_cred=SYS_CRED
          -host_cred=HOST_CRED
```

set_config_history_retention_period

Sets the amount of time for which the configuration history is retained.

Format

```
emcli set_config_history_retention_period
      -period="Retention period in months"
```

Options

- **period**

  Retention period in months. The value must be in the range of 1 to 60 inclusive.

Example

This example sets the retention period to 12 months.

```
emcli set_config_history_retention_period
-period=12
```

set_connection_mode

Sets the new MOS connection mode.

Format

```
emcli set_connection_mode
        -mode="online | offline"
```

Examples

```
emcli set_connection_mode -mode="offline"

emcli set_connection_mode -mode="online"
```

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_patch_plan_data
show_patch_plan
submit_patch_plan

upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

set_credential

Sets preferred credentials for given users.

> **Note:**
>
> This command does not support the COLLECTION credential sets.

Format

```
emcli set_credential
      -target_type="ttype"
      [-target_name="tname"]
      -credential_set="cred_set"
      [-user="user"]
      -columns="col1:newval1;col2:newval2;PDP:SUDO/POWERBROKER;RUNAS:oracle;
         PROFILE:user1..."
      [-input_file="tag1:file_path1;tag2:file_path2;..."]
      [-oracle_homes="home1;home2"]
      [-monitoring]

[ ]  indicates that the parameter is optional
```

Options

- **target_type**

  Type of target. This must be "host" if the `-oracle_homes` parameter is specified.

- **target_name**

  Name of the target. Omit this argument to set enterprise preferred credentials. This must be the host name if the `-oracle_homes` parameter is specified.

- **credential_set**

  Credential set affected.

- **user**

  Enterprise Manager user whose credentials are affected. If omitted, the current user's credentials are affected.

- **columns**

  Name and new value of the column(s) to set. Every column of the credential set must be specified. Alternatively, a tag from the `-input_file` argument can be used so that the credential values are not seen on the command line. You can specify this argument more than once.

- **input_file**

  Path of the file that has the `-columns` argument(s). This is used to hide passwords. Each path must be accompanied by a tag referenced in the `-columns` parameter. You can specify this option more than once.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **oracle_homes**

  Name of Oracle homes on the target host. Credentials will be added/updated for all specified homes.

  Note: The list of columns and the credential sets they belong to is included in the metadata file for each target type. This and other credential information is in the `<CredentialInfo>` section of the metadata.

- **monitoring**

  Flag indicating that credentials affected are monitoring credentials. If omitted, the credentials affected are preferred credentials. Monitoring credentials require specifying the target_name option.

Examples

Example 1

In this example, `FILE1` is a tag to refer to the contents of `passwordFile`. Note that Example 2 has the same effect as Example 1.

```
emcli set_credential
      -target_type=oracle_database
      -target_name=myDB
      -credential_set=DBCredsNormal
      -user=admin1
```

```
            -column=FILE1
            -input_file=FILE1:passwordFile
```

Example 2

In this example, the contents of the `passwordFile`: is
`username:joe;password:newPass;role:newRole`

```
emcli set_credential
      -target_type=host
      -target_name=host.example.com
      -credential_set=OHCreds
      -user=admin1
      -column="OHUsername:joe;OHPassword:newPass"
      -oracle_homes="database1;mydb"
```

set_db_service_properties

Sets and updates the Database as a Service (DBaaS) target properties by providing the
database unique name of an underlying database target and property name/value for the
service target.

Format

```
emcli set_db_service_properties
      -db_unique_name="database unique name"
      -property_name="property name"
      -property_value="property value"
```

Options

*   db_unique_name

    The database name of the database target on which the DBaaS target relies. You can
    find this name on the Last Collected page of the database target, or you can query for it.

*   property_name

    The target property name of a DBaaS target (for example, `company_gtp_cost_center`,
    `company_gtp_line_of_bus`, `company_gtp_contact`).

*   property_value

    Value you want to assign to the database target's property name of the DBaaS target.

Example

The following example shows how to set a property value of `web_group1` for the
`dev_cost_center` property name of the `company_e_commerce` database:

```
emcli set_db_service_properties -db_unique_name="company_e_commerce" -
property_name="dev_cost_center" -property_value="web_group1"
```

set_default_pref_cred

Sets a named credential as a default preferred credential. If you decide to use preferred
credentials for an Enterprise Manager operation and preferred credentials are not set for the
target, the default credentials for this target type that you set are used. Default credentials are
set at the target-type level.

Format

```
emcli set_default_pref_cred
        -set_name="set_name"
        -target_type="ttype"
        -credential_name="cred_name"
        [-credential_owner ="owner]"
        [-test]
        [-test_target_name="test_target_name"]

[ ]  indicates that the parameter is optional
```

Options

- **set_name**

  Sets the preferred credential for this credential set.

- **target_type**

  Target type for the credential set.

- **credential_name**

  Name of the credential.

- **credential_owner**

  Owner of the credential. This defaults to the currently logged-in user.

- **test**

  Tests the credential before setting it as the default credential.

- **test_target_name**

  Tests the target name if the global credential is set as the default preferred credential.

Examples

Example 1

This example sets the named credential MyHostCredentials as the default preferred credential for the target type host as HostCredsNormal.

```
emcli set_default_pref_credential
        -set_name=HostCredsNormal
        -target_type=host
        -credential_name=MyHostCredentials
        -credential_owner="Joe"
```

Example 2

This example sets the named credential MyHostCredentials as the default preferred credential for the target type host as HostCredsNormal. The command tests the named credential MyHostCredentials against server1.example.com before setting it as a default preferred credential.

```
emcli set_default_pref_cred
        -set_name=HostCredsNormal
        -target_type=host
        -credential_name=MyHostCredential
        -credential_owner="Joe"
```

```
-test
-test_target_name=server1.example.com
```

set_default_privilege_delegation_setting

Sets the default privilege delegation settings for one or more platforms.

Format

Standard Mode

```
emcli set_default_privilege_delegation_setting
    -default_setting_list="platform1:setting_name1;platform2:setting_name2"
    [-separator="separator:attribute_name:character"]
    [-subseparator="subseparator:attribute_name:character"]
```

Interactive or Script Mode

```
set_default_privilege_delegation_setting(
    default_setting_list="platform1:setting_name1;platform2:setting_name2"
    [,separator="separator:attribute_name:character"]
    [,subseparator="subseparator:attribute_name:character"]
    )
```

```
[ ]  indicates that the parameter is optional
```

Exit Codes

0 on success. A non-zero value means verb processing was not successful.

Options

*   **default_setting_list**

    List of default settings per platform. Supported platforms: Linux, HP-UX, AIX, SunOS.

*   **separator**

    By default, multi-value input attributes use the semicolon (;) character as a separator. Specifying this option overrides the default separator value.

    Example: `separator="<attribute_name=sep_char>"` where attribute_name is the name of the attribute for which you want to override the separator character, and sep_char is new separator character.

    Example: `separator="att=#"` changes the separator character to a pound sign (#).

*   **subseparator**

    By default, multi-value input attributes use the colon (:) character as the sub-separator. Specifying this option overrides the default sub-separator value.

    Example: `subseparator="<attribute_name=sep_char"` where `attribute_name` is the name of the attribute for which you want to override the separator character, and `sep_char` is the new sub-separator character.

    Example: subseparator="att=#" changes the sub-separator character to a pound sign.

Examples

Example 1

This example sets the privilege delegation setting to SUDO1 for Linux platforms and SUDO2 for HP-UX platforms.

```
emcli set_default_privilege_delegation_setting
        -default_setting_list="Linux:SUDO1;HP-UX:SUDO2"
```

Example 2

This example sets the privilege delegation setting to SUDO_SETTING_1 for Linux and SUDO_SETTING_2 for HP-UX. The default separator has been changed to a comma (,) and the subseparator to a hash tag (#).

```
emcli set_default_privilege_delegation_setting
        -default_setting_list="Linux#SUDO_SETTING_1,HP-UX#SUDO_SETTING_2"
        -separator="default_setting_list=,"
        -subseparator="default_setting_list=#"
```

set_engr_sys_patching_options

Sets up the patch deployment options for the component of an engineered system.

Format

```
emcli set_engr_sys_patching_options
      -system_target_name="system_target_name"
      -system_target_type="system_target_type"
      -component_type="component_type"
      -input_file=data:"input_file_path"
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- component_type

  Specifies the engineered system component type.

- input_file

  Specifies the path of the file that contains the patching deployment options. The following is an example of an input file: deployment.

Options

```
.StageLocation=/u01/stagepatch
                deploymentOptions.PatchingMode=rolling
                deploymentOptions.StagePatches=true
                deploymentOptions.OpatchUpgrade=false
                deploymentOptions.SafeMode=false
```

Note: "safeMode" indicates whether a blackout must be created on the target while patching it. The recommended value is true.

Examples

The following example sets up the patching deployment options for the component "Oracle Infiniband Switch" of the engineered system "DB Machine slcm12.example.com":"oracle_dbmachine", using the inputs specified in "/tmp/deploymentoptions.prop"

```
emcli  set_engr_sys_patching_options
       -system_target_name="DB Machine slcm12.example.com"
       -system_target_type="oracle_dbmachine"
       -component_type="Oracle Infiniband Switch"
       -input_file=data:"/tmp/deploymentoptions.prop"
```

set_gold_agent_update

Creates a Management Agent Gold Image Policy and defines default values for Management Agent upgrade.

Format

```
emcli set_gold_agent_update_policy
     [-additional_parameters="additional_parameters"]
     [-pre_script_loc="pre_script_loc"]
     [-post_script_loc= " post_script_loc"]
     [-is_pre_script_on_oms= " is_pre_script_on_oms"]
     [-is_post_script_on_oms= " is_post_script_on_oms"]]
      [-stage_location= " stage_location"]
     [-is_staged= " is_staged"]
     [-stage_action= " stage_action"]
     [-batch_size= " batch_size"]
     [-frequency= " frequency"]
     [-success_rate= " success_rate"]
     [-update_profile= " update_profile"]
     [-profile_path= " profile_path"]
     [-email= " email"]
     [-run_preCleanup= " run_preCleanup"]
     [-run_postCleanup= " run_postCleanup"]
```

[ ] indicates that the parameter is optional.

Parameters

- additional_parameters

  Additional parameters set in the repository to create a Management Agent Gold Image Policy.

- pre_script_loc

  Prescript location set in the repository to create a Management Agent Gold Image Policy.

- post_script_loc

  Postscript location set in the repository to create a Management Agent Gold Image Policy.

- is_pre_script_on_oms

  Value set for this parameter to create a Management Agent Gold Image Policy.

- is_post_script_on_oms

Value set for this parameter to create a Management Agent Gold Image Policy.

- stage_location

  Stage location value set in the repository to create a Management Agent Gold Image Policy.

- is_staged

  Value set for this parameter in the repository to create a Management Agent Gold Image Policy.

- stage_action

  Value set for this parameter in the repository to create a Management Agent Gold Image Policy.

- batch_size

  Value set for batch size in the repository to create a Management Agent Gold Image Policy.

- frequency

  Value set for frequency in the repository to create a Management Agent Gold Image Policy.

- success_rate

  Value set for success rate in the repository to create a Management Agent Gold Image Policy.

- update_profile

  Value set for this parameter in the repository to create a Management Agent Gold Image Policy.

- profile_path

  Value set for profile path in the repository to create a Management Agent Gold Image Policy.

- email

  Email set in the repository to create a Management Agent Gold Image Policy.

- run_preCleanup

  Value set for this parameter in the repository to create a Management Agent Gold Image Policy.

- -run_postCleanup

  Value set for this parameter in the repository to create a Management Agent Gold Image Policy.

Examples

Example 1

The following example sets additional parameters in the repository':

```
emcli set_gold_agent_update_policy
```

Example 2

The following example sets prescript location in the repository:

```
emcli set_gold_agent_update_policy
-pre_script_loc=/home/john/pretscript
```

Example 3

The following example sets stage location in the repository':

```
emcli set_gold_agent_update_policy
-stage_location=/scratch/tmp
```

set_key_beacons_tests

Defines key beacons and tests of the service.

Format

```
emcli set_key_beacons_tests
      -name=<target_name>
      -type=<target_type>
      [-beacons=<beacon_names>]+
      [-tests='test1:type1;test2:type2;...']+
      [-removeKey]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Service target name.

- **type**

  Service target type.

- **beacons**

  Names of beacons to set as key (or non-key).

- **tests**

  Names and types of tests to set as key (or non-key).

- **removeKey**

  If specified, the mode is (remove key); that is, the specified tests and beacons will be set as non-key.

  If not specified, the mode is (add key); that is, the specified tests and beacons will be set as key.

Examples

Example 1

This example sets `MyTest/HTTP`, `MyTest2/FTP` and `MyBeacon` as non-key elements of service `MyTarget/generic_service`.

```
emcli set_key_beacons_tests
      -name='MyTarget'
      -type='generic_service'
      -tests='MyTest:HTTP;MyTest2:FTP'
      -beacons='MyBeacon' -removeKey
```

Example 2

This example sets `MyBeacon` and `MyBeacon2` as key beacons of service `MyTarget/generic_service`.

```
emcli set_key_beacons_tests
     -name='MyTarget'
     -type='generic_service'
     -beacons='MyBeacon;MyBeacon2'
```

set_logging_property

Sets the property value corresponding to the specified logging property name.

Format

```
emcli set_logging_property
       -property_name="propertyName"
       [-oms_name="omsName"]
       -property_value="propertyValue"
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **property_name**

    Name of the logging property whose value needs to be set.

*   **oms_name**

    Name of the management server where the logging property needs to be set.

*   **property_value**

    Value to be set.

Examples

Example 1

This example sets the value for the property name "propName" on the management server myhost:1159_Management_Service to "propValue."

```
set_logging_property -property_name=propName -property_value=propValue
-oms_name="myhost:1159_Management_Service"
```

Example 2

This example sets the value for the property name "propName" to "propValue" on all of the management servers.

```
set_logging_property -property_name=propName -property_value=propValue
```

set_metric_promotion

Creates or edits a metric promotion based on a test or system.

Format

```
emcli set_metric_promotion
     -name=<service_target_name
```

```
                -type=<service_target_type
                ***[-category=Usage/Performance/Business]
                -basedOn=SYSTEM|TESTS|SUB_SERVICE
                -aggFunction=AVG|MAX|MIN|SUM|COPY
                [-promotedMetricName=<promoted_metric>]
                [-promotedMetricColumn=<promoted_metric_column>]
                -promotedMetricKey=<key_value_of_promoted_metric>
                [-metricName=<dependent_metric_name>]
                -column=<dependent_metric_column>
                *[-depTargetType=<target_type_of_dependent_targets>]
                *#[-depTargets='target1;target2...']
                *#[-depTargetKeyValues='target1:key11|key12|key13..;
                    target2:key21|key22|key23..']
                *[-depMetricKeyValue=<dependent_metric_key_column>]
                **[-testname=<dependent_test_name]
                **[-testtype=<dependent_test_type]
                **[-metricLevel=TXN|STEP|STEPGROUP]
                **[-beacons='bcn1;bcn2..']
                **[-depTestComponent=<step_or_stepgroup_name>]
                [-threshold='critical_threshold_value;warning_threshold_value;
                    threshold_operator (EQ|LE|LT|GT|GE)']
                -mode=CREATE|EDIT
                #[-includeRuleBasedTargets = YES|NO]
                [-targetFilter = ALL|STARTS_WITH:<filter>|ENDS_WITH:<filter>|
                    CONTAINS:<filter>|EQUALS:<filter>]
[ ]  indicates that the parameter is optional.
```

Key:

* — Might be required if `basedOn` is set to SYSTEM ** — Might be required if `basedOn` is set to TESTS *** — Might be required if `basedOn` is set to `SUB_SERVICE` # — One of these values is required for system-based metrics.

Options

- **category**

  Defines whether the promoted metric is a usage, performance, or business metric of a service. Category is used to determine the promoted metric name and metric column. If you do not specify this option, you must specify the `promotedMetricName` and `promotedMetricColumn` options.

- **basedOn**

  Determines whether the promotion is test-based or system-based.

- **aggFunction**

  Determines the aggregate function to be used to compute the promoted metric. `AVG/MAX/MIN/SUM` takes average, max, min, and sum of the dependent metrics, respectively. `COPY` only copies over a single dependent metric to the promoted metric.

- **promotedMetricName**

  Promoted metric name. This is optional if the category is specified.

- **promotedMetricColumn**

  Promoted metric column. This is optional if the category is specified.

- **promotedMetricKey**

  Required argument that determines the key value of the promoted metric. It is equivalent to the displayed name of the promoted metric in the UI.

- **metricName**

  Required argument if the dependent metric column is collected by more than one metric.

- **column**

  Dependent metric column.

- **depTargetType**

  All dependent targets should be of this target type.

- **depTargets**

  Specifies the dependent targets. This argument is ignored if you specify `depTargetKeyValues`.

- **depTargetKeyValues**

  Specifies the key values associated with the dependent targets. Specify multiple key values for a single target by repeating the entry in the following format: `'tgt1:key1;tgt1:key2...'`

- **depMetricKeyValue**

  Required if the dependent metric is a transpose metric. It is the key value that applies to all the dependent targets.

- **testname**

  Defines the name of the test to be used in promoting the metric.

- **testtype**

  Defines the type of test to be used in promoting the metric.

- **metricLevel**

  Some metrics can be promoted on step-level. This option defines the level to be used during promotion.

- **beacons**

  List of beacons to be used for promoting the metric data.

- **depTestComponent**

  If `metricLevel` is not `TXN`, this option is required to specify which step or which step group is being promoted.

- **threshold**

  Defines a threshold on the promoted metric.-mode: The mode can be `CREATE` or EDIT.

- **includeRuleBasedTargets**

  If YES, the system member targets available at the time of metric evaluation are considered for metric evaluation. The default is NO. This option is applicable only for system-based metrics.

- **targetFilter**

  The given target filter value is compared with target names of system member targets. The member targets that meet this filter value will participate in the metric evaluation. For a target filter, wild cards such as *, % an so forth are not accepted.

e.g., ALL, STARTS_WITH:EM, ENDS_WITH:EM, CONTAINS:EM and EQUALS:EM.

Examples

Example 1

This example creates a promoted Performance metric with key value `mymetric1` on service `MyTarget` using `MyTest/HTTP`. The promoted metric takes the maximum of the `dns_time` metric column returned by the `MyBeacon` and `mybcn1` beacons. It also has a threshold with 'greater or equal to' operator (GE) with the critical value set to 200 and warning value set to 100.

```
emcli set_metric_promotion -name='MyTarget' -type='generic_service'
    -category=Performance -basedOn=test -aggFunction=MAX
    -testname='MyTest' -testtype=HTTP
    -beacons='MyBeacon, mybcn1'
    -promotedMetricKey=mymetric1 -column=dns_time -metricName=http_response
    -metricLevel=TXN -threshold='200;100;GE' -mode=CREATE
```

Example 2

This example creates a promoted `Usage` metric with the key value `AppServerComponentUsage` on service `MyTarget`. The dependent target is `'myapp_server'` with type `'oracle_ias'`. The promoted metric computes the average value of the `cpu.component` metric column for the specified key values.

```
emcli set_metric_promotion -name='MyTarget' -type='generic_service'
    -category=Usage -basedOn=system -aggFunction=AVG
    -promotedMetricKey=AppServerComponentUsage -depTargetType=oracle_ias
    -column=cpu.component
    -metricName=opmn_process_info
    -depTargetKeyValues='myapp_server:petstore;myapp_server:http_server'
    -mode=CREATE
```

set_monitoring_credential

Sets a monitoring credential set for a target. You can provide input parameters using command line arguments or the input properties file. It also supports the input_file parameter for passwords and parameter values.

Format

```
emcli set_monitoring_credential
        -target_name=<target_name>
        -target_type=<ttype>
        -set_name=<set_name>
        -cred_type=<credential_type>
        -auth_target_type=<auth_ttype>
        -test
        -input_file=<tag|value>
        -properties_file=<filename>
        -attributes=<p1:v1;p2:v2;...>
```

Options

• **target_name**

  Sets the monitoring credential for this target.

• **target_type**

Target type for the target.

- **set_name**

  Sets the monitoring credential for this credential set name.

- **cred_type**

  Credential type for the credential to set as the monitoring credential.

- **auth_target_type**

  Authenticating target type. Defaults to target_type.

- **test**

  Tests the credential against the target(s) before setting the monitoring credential.

- **input_file**

  Supplies sensitive property values from the file.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **properties_file**

  Passes all parameters from the file. Values provided on the command line take precedence.

- **attributes**

  Specify credential columns as follows:

  ```
  colname:colvalue;colname:colvalue
  ```

  You can change the separator value using -separator=attributes=<newvalue>, and you can change the subseperator value using -subseparator=attributes=<newvalue>.

Examples

Example 1

This example sets the monitoring credential set DBCredsMonitoring for the target testdb.example.com:oracle_database with user name foo, password bar, and role normal.

```
emcli set_monitoring_credential
        -target_name=testdb.example.com
        -target_type=oracle_database
        -set_name=DBCredsMonitoring
        -cred_type=DBCreds
        -attributes="DBUserName:foo;DBPassword:bar;DBRole:normal"
```

Example 2

This example reads the password from the mypasswordfile.txt file.

```
emcli set_monitoring_credential
        -target_name=testdb.example.com
        -target_type=oracle_database
        -set_name=DBCredsMonitoring
        -cred_type=DBCreds
        -attributes="DBUserName:foo;DBPassword:tag;DBRole:normal"
        -input_file="tag:mypasswordfile.txt"
```

ORACLE®

set_mos_credentials

Sets My Oracle Support credentials in OMS. When prompted for a password, enter you My Oracle Support password.

Format

```
emcli set_mos_credentials
      -username="My Oracle Support Username"
      [-password "My Oracle Support Password"]
      [-verbose_exception]
```

Options

- **username**

  My Oracle Support username.

- **password**

  My Oracle Support password. If this option is not specified, the user is prompted for the password interactively.

Example

The follow example sets My Oracle Support credentials in OMS to - username=xyz@oracle.com.

```
emcli set_mos_credentials
      -username=xyz@oracle.com
```

set_oms_property

Sets the property value corresponding to the specified property name.

Format

```
emcli set_oms_property
      -property_name="propertyName"
      [-oms_name="omsName"]
      -property_value="propertyValue"
```

```
[ ]  indicates that the parameter is optional
```

Options

- **property_name**

  Name of the property whose value needs to be set.

- **oms_name**

  Name of the management server for which the property needs to be set.

- **property_value**

  Property value to be set.

Examples

Example 1

This example sets the value for the property name "propName" on the management server myhost:1159_Management_Service to "propValue."

```
set_oms_property -property_name=propName -property_value=propValue -
oms_name="myhost:1159_Management_Service"
```

Example 2

This example sets the value for the property name "propName" to "propValue" on all of the management servers.

```
set the value for the property name "propName" to "propValue" on all the
management servers
```

set_patch_plan_data

Sets user-editable data. The get_patch_plan_data verb is useful when used preceding this verb.

Format

```
emcli set_patch_plan_data
        -name="name"
        -input_file=data:"file_path"
        [-impact_other_targets="add_all|add_original_only|cancel"]
        [-problems_assoc_patches="ignore_all_warnings|cancel"]

[ ]  indicates that the parameter is optional
```

Options

* **name**

  Sets the preferred credential for this credential set.

* **input_file**

  Sets the preferred credential for this target.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

* **impact_other_targets**

  Target type for the target/credential set.

* **problems_assoc_patches**

  Name of the credential.

Examples

```
emcli set_patch_plan_data -name="plan name"
-input_file=data:"/tmp/patchplan.pros"

emcli set_patch_plan_data -name="plan name"
-input_file=data:"/tmp/patchplan.pros" -impact_other_targets="add_all"

emcli set_patch_plan_data -name="plan name"
-input_file=data:"/tmp/patchplan.pros" -impact_other_targets="add_all"
-problems_assoc_patches="ignore_all_warnings"
```

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
show_patch_plan
submit_patch_plan

upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

set_preferred_credential

Sets a named credential as a target-preferred credential for the user.

Format

```
emcli set_preferred_credential
       -set_name="set_name"
       -target_name="target_name"
       -target_type="ttype"
       -credential_name="cred_name"
       [-credential_owner ="owner]"
       [-test]

[ ]  indicates that the parameter is optional
```

Options

- **set_name**

  Sets the preferred credential for this credential set.

- **target_name**

  Sets the preferred credential for this target.

- **target_type**

  Target type for the target/credential set.

- **credential_name**

  Name of the credential.

- **credential_owner**

  Owner of the credential. This defaults to the currently logged in user.

- **test**

  Tests the credential against the target_name before setting the preferred credential.

  Examples

  Example 1

  This example sets the named credential MyHostCredentials as the target preferred credential for the target test.example.com:host as HostCredsNormal.

  ```
  emcli set_preferred_credential
          -set_name=HostCredsNormal
          -target_name=test.oracle.com
          -target_type=host
          -credential_name=MyHostCredentials
          -credential_owner="Joe"
  ```

  Example 2

  This example sets the named credential MyDBCredentials as the target preferred credential for the target myDB:oracle_database as Normal Database Credentials. The command tests the named credential against myDB:oracle_database before setting the preferred credential.

  ```
  emcli set_preferred_credential
          -target_type=oracle_database
          -target_name=myDB
          -set_name=DBCredsNormal
          -credential_name=MyDBCredentials
          -credential_owner="Joe"
          -test
  ```

set_properties

Sets the property for a test or beacons.

Format

```
emcli set_properties
      -name=<target_name>
      -type=<target_type>
      -testname=<test_name>
      -testtype=<test_type>
      [-beacons=<beacon_names>]
      [-properties='prop1:value1;prop2:value2;..']+

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Service target name.

- **type**

  Service target type.

- **testname**

Name of the test to set the property on.

- **testtype**

  Type of test to set the property on.

- **beacons**

  Names of the beacons to set the property on.

- **properties**

  Names and values of the properties to be set (can be multiple).

Examples

Example 1

This example sets the property `timeout` to `30000` and granularity to `transaction` for the test `MyTest` defined on `MyTarget` for all beacons.

```
emcli set_properties -name='MyTarget' -type='generic_service'
     -testname='MyTest' -testtype='HTTP'
     -propertyName='timeout:30000;granularity:transaction'
```

Example 2

This example sets the property value to `30000` of the test `MyTest` defined on `MyTarget` for only `MyBeacon` and `MyBeacon2`. This only works if the specified properties can be set on a per beacon level.

```
emcli set_properties -name='MyTarget' -type='generic_service'
     -testname='MyTest' -testtype='HTTP'
     -bcnName='MyBeacon;MyBeacon2'
     -propertyName='timeout' -propertyValue='30000'
```

set_reverse_ping_interval

Modifies the maximum waiting time for the Management Agents. You need to provide Agent names for the modification.

Format

```
emcli set_reverse_ping_interval
      -agent_names="agent1[;agent2...]"|-all_agents
      -value=" "|-reset_to_default

[ ]  indicates that the parameter is optional
```

Options

- **agent_names**

  Management agents (host:port) on which the modification needs to be performed.

- **all_agents**

  Use only when all Agents need to be modified with the new value.

- **value**

  New value to which the existing waiting time needs to be updated.

- **reset_to_default**

Use when the value needs to be reset to the default value.

Examples

Example 1

This example modifies the existing waiting time with the new value provided, which in this case is 240.

```
emcli set_reverse_ping_interval -agent_names="myhost1.example.com:1838" -
value=240
```

Example 2

This example modifies the existing waiting time for the provided Agents with the default value in the Ping System.

```
emcli set_reverse_ping_interval -
agent_names="myhost1.example.com:1838;myhost2.example.com:4352" -reset_to_default
```

set_standby_agent

Permits targets to relocate from one Management Agent to another. This verb always populates a table that determines which targets from the source Management Agent to the destination Management Agent are permitted to relocate for the Enterprise Manager target.

Format

```
emcli set_standby_agent
      -src_agent=<source_agent>
      -dest_agent=<destination_agent>
      -target_name=<target_name>
      -target_type=<target_type>

[ ]  indicates that the parameter is optional
```

Options

- **src_agent**

  Management Agent currently monitoring the targets. If srcAgent is not known, enter currentOwner as the argument.

- **dest_agent**

  Management Agent for which you want to monitor the targets.

- **target_name**

  Name of the target to be moved.

- **target_type**

  Type of target to be moved.

Output

Output message of the command execution.

set_target_property_value

Sets the value of a target property for a specified target. Any prior values of the target property are overwritten. When assigning values to the Oracle-provided target properties, use the English names of these target properties:

Comment, Lifecycle Status, Line of Business, Location, Contact

Acceptable values for Lifecyle Status are:

- Development

- MissionCritical

- Production

- Stage

- Test

For cluster target types, the value of the target property automatically propagates to all of its member targets. This happens even without the -propagate_to_members parameter. The propagate_to_members parameter is used for aggregate non-cluster targets where the desired behavior is to propagate the target property values to members of the aggregate target. Note that it will propagate to current members of the aggregate, and not targets that are added in the future.

> **✎ Note:**
>
> You can only set up and propagate one property at a time to members.

Format

```
emcli set_target_property_value        -
property_records="target_name:target_type:property_name:property_value"        [-
separator=property_records="sep_string"]        [-
subseparator=property_records="subsep_string"]        [-
input_file="parameter_tag:file_path"]
        [-propagate_to_members]

[ ]  indicates that the parameter is optional
```

Options

- **property_records**

  List of property records. The following parts comprise each property record:

  <target_name>:<target_type>:<property_name>:property_value>

  – target_name — Target name of the target for which you want to update the property.

  – target_type — Target type of the target.

  – property_name — Name of the property whose value you want to update. Property names are case sensitive. You can execute the list_target_property_names verb for a list of possible property names.

  – property_value — Value to be assigned/updated for the property.

- **separator**

  When specifying multiple property records, use the separator string delimiter as a delimiter between property records. The default separator delimiter is ";".

- **subseparator**

  String delimiter to be used between parts of a property record. The default subseparator delimiter is ":".

- **input_file**

  Used in conjunction with the -property_records option, this option enables you to provide the property records in a file. This option specifies a mapping between a tag and a local file path. The tag is specified in lieu of property records. The tag cannot contain colons (:) or semi-colons (;) .

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **propagate_to_members**

  Used for group and system targets to also propagate the property to all of its members.

Examples

Example 1

This example sets the 'Owner Name' property to Jane Smith for the database test_database.

```
emcli set_target_property_value
      -property_records="test_database:oracle_database:Owner Name:Jane Smith"
```

Example 2

This example sets the Owner property to Jane Smith for the database test_db, and also sets the Asset Number property to 100 for the database test_db1.

```
emcli set_target_property_value
         -property_records="test_db:oracle_database:Owner:Jane Smith;
          test_db1:oracle_database:Asset Number:100"
```

Example 3

This example sets the Lifecycle Status for all members of the composite system, which includes the following target types.

```
myserver.myhost.com_sys: Database System
myserver.myhost.com: Database Instance
myserver.myhost.com_CDBROOT: Pluggable Database
myserver.myhost.com_PDB1: Pluggable Database
LISTENER_myserver.myhost.com: Listener
OraDB12Home1_1_myserver.myhost.com_5355: Oracle Home

emcli set_target_property_value -
property_records="myserver.myhost.com_sys:oracle_dbsys:LifeCycle
Status:Production" -propagate_to_members
```

set_test_threshold

Sets a test threshold.

Format

```
emcli set_test_threshold
      -name=<target_name>
      -type=<target_type>
      -testname=<test_name>
      -testtype=<test_type>
      -metricName=<metric_name>
      -metricColumn=<metric_column>
      -occurrences=<occurrences>
      [-warningThres=<warning_threshold>]
      [-criticalThres=<critical_threshold>]
      [-operator=<operator>]
      [-beaconName=<beacon_name>]
      [-stepName=<step_name>]
      [-stepGroupName=<stepgroup_name>]

[ ]  indicates that the parameter is optional
```

Examples

```
emcli set_test_threshold -name="Service Name"
              -type="generic_service"
              -testname="Test Name"
              -testtype="HTTP"
              -metricName="http_response"
              -metricColumn="timing"
              -occurrences=1
              -warningThres=100000
```

setup

Configures EM CLI to work with a specific management server.

You can set up the EM CLI client either in secure mode by specifying the -noautologin option, or unsecure mode by specifying the -autologin otpion. -noautologin is the default, so if you do not specify either option, the EM CLI client is automatically set up in secure mode.

The configuration directory will contain log files generated by EM CLI to record informational and error messages generated during operations.

Format

```
emcli setup
      -url="http[s]://host:port/em"
      -username=<EM_console_username]
      [-password=<password_of_user>]
      -dir=<local_emcli_config_directory>
      [-localdirans=yes|no]
      [-licans=yes|no]
      [-trustall]
      [-certans=yes|no]
      [-nocertvalidate]
      [-novalidate]
```

```
[-autologin]
[-noautologin]
[-noregister]
[-custom_attrib_file=<custom_attr_file_path>]
```

`[ ]  indicates that the parameter is optional`

Options

- **url**

  URL of the Oracle Management Server (OMS). `host` specifies the host of the OMS. `port` specifies the listening port of the OMS. Both http and https protocols are supported. (https is recommended for security reasons).

- **username**

  Enterprise Manager user name to be used by all subsequent EM CLI commands when contacting the OMS.

  If the SSO user is also an Enterprise Manager user (that is, authenticated in LDAP/OID), you can only register EM CLI with the ssousername. After you enable SSO for the OMS, you cannot subsequently register EM CLI with only username.

- **password**

  Enterprise Manager user password. If you do not specify this option, you are prompted for the password interactively.

  > **✎ Note:**
  >
  > Providing a password on the command line is insecure and should be avoided.

- **dir**

  Directory where an EM CLI configuration directory will be created. This directory must be on a locally mounted file system. A warning and confirmation is issued for an HTTPS URL if the directory is not heuristically identified as such (unless you specify `trustall`). The directory can be relative to the working directory where setup is called, or it can be absolute. This option defaults to the user's home directory.

- **localdirans**

  Indicates whether the setup directory given with the -dir option is a local directory. Specify yes to indicate that the setup directory is local, and specify no to indicate that the setup directory is non-local.

- **licans**

  Indicates whether the license is accepted or not accepted by the user. Specify yes to accept the license, or specify no to not accept the license.

- **trustall**

  Automatically accepts any server certificate from the OMS, which results in lower security.

- **certans**

Indicates whether the certificate needs to be trusted without having to prompt the user. Specify yes to trust the certificate, and specify no to not trust the certificate.

- **nocertvalidate**

  Does not validate the host name in the SSL certificate provided by the OMS.

- **novalidate**

  Does not authenticate the Enterprise Manager user name or SSO user name against the OMS. Assume the given user name is valid. This enables the configuration to be stored (Enterprise Manager URL and user) without validating or connecting to Enterprise Manager. This might be useful in scenarios where Enterprise Manager is not up when you do run the setup command.

- **autologin**

  In this mode, credentials are stored on the EM CLI client system. Autologin mode is preserved until emcli logout is executed. If the session has expired when a verb is executed, login is automatically performed and the verb is executed.

  Verbs executed after emcli logout may fail with the message "Error: Session expired. Run emcli login to establish a session." You need to run the login verb to log in to EM CLI after an emcli logout. After the Enterprise manager user's password has changed, you need to log in with the ID and the new password. The new password will subsequently be stored.

  Note that noautologin is the default mode.

- **noautologin**

  In this default mode, credentials are not stored on the EM CLI client system. If the session has expired when a verb is executed, you have to explicitly run the login verb and then run the required verb.

- **noregister**

  Does not register this EM CLI instance.

- **custom_attrib_file**

  Path name of a file containing Audit Custom Attribute values. This option is required when the OMS is configured for Audit Custom Attributes. If you do not provide custom_attrib_file, you are prompted to enter the values of the custom attributes.

  The file can contain up to three lines, each containing the description of one custom attribute. Each line should be of the form:

  ```
  <attr-name>#<attr-displayname>#<isMandatory>#<attr-value>
  ```

  - **#** — Field separator.
  - **attr-name** — Name of the attribute.
  - **attr-displayname** — Display name of the attribute.
  - **isMandatory** — 1 if the attribute is mandatory, otherwise 0.
  - **attr-value** — Value of the custom attribute.

Examples

```
emcli setup -url=http://omsmachine.example.com:7770/em -username=sysman
```

To configure the EM CLI Client to function with multiple OMSes by implementing multiple setups, do the following:

1. Set up the EM CLI client for OMS1 at location dir1:

```
emcli setup -dir=<dir1> -url=<Url of OMS1> -user=<EM Username for OMS1>
```

2. Set up the EM CLI client for OMS2 at location dir2:

```
emcli setup -dir=<dir2> -url=<Url of OMS1> -user=<EM Username for OMS2>
```

3. Set the environment variable EMCLI_STATE_DIR to point to the setup directory for OMS1:

```
setenv EMCLI_STATE_DIR <dir1>
```

This sets the EM CLI Client to function with OMS1.

4. Set the environment variable EMCLI_STATE_DIR to point to the setup directory for OMS2:

```
setenv EMCLI_STATE_DIR <dir2>
```

This sets the EM CLI Client to function with OMS2.

setup_bipublisher

Sets up a relationship between Enterprise Manager and a BI Publisher Web Application. If a relationship already exists, you must provide the -force option. The Enterprise Manager System Reports are deployed to the newly configured BI Publisher Web Application. To just change the registration details without deploying the reports, use the -nodeploy option. Detailed status messages are provided for all operations.

Use the -force option to overwrite existing copies of reports if they exist. If you do not want to deploy following setup, you can specify the -nodeploy option.

> **✎ Note:**
>
> This verb requires Enterprise Manager Super Administrator privileges.

Format

```
emcli setup_bipublisher
    [-force]
    -protocol=http|https
    -host=<hostname>
    -port=<portnumber>
    -uri=xmlpserver
    [-nodeploy]

[ ]  indicates that the parameter is optional
```

Options

- **force**

  Overwrites existing copies of reports if they exist. The following scenarios are affected by this option:

- If a relationship exists between Enterprise Manager and a BI Publisher Web Application, this option overrides it with a new relationship (-host, -port, and so forth).

- If you do not specify -nodeploy, this option causes deployed reports to overwrite any that may already exist for the BI Publisher Web Application in the "Enterprise Manager Cloud Control" folder.

- **protocol**

  Must be either http or https.

- **host**

  Name of the host that is running, or server load balancer fronting the BI Publisher Web Application.

- **port**

  Port number of the web service.

- **uri**

  Web application context root, which must be xmlpserver.

- **nodeploy**

  Specify if you do not want to deploy following setup. Suppresses the Enterprise Manager BI Publisher System Reports to the BI Publisher Web Application. You can do this later with the deploy_bipublisher_reports cli.

Examples

Example 1

```
emcli setup_bipublisher
    -protocol=https
    -host=www.somehost.com
    -port=7801
    -uri=xmlpserver
```

Example 2

This example reconfigures the BI Publisher Managed Server (BIP) inside the WebLogic Server console to listen on a different port (9704):

```
emcli setup_bipublisher -protocol=https -host=somehost.com -port=9704 -uri=xmlpserver -
force -nodeploy
```

show_bda_clusters

Lists all Hadoop clusters in the BDA network. If a host is specified, lists all Hadoop clusters in the network where the host is present.

Format

```
emcli show_bda_clusters
       [-host="host_name"]
```

[ ] indicates that the parameter is optional

Options

- **host_name**

Name of a particular host in the BDA network.

Examples

Example 1

The following example lists all Hadoop cluster targets in the BDA network:

```
emcli show_bda_clusters
```

Example 2

The following example lists all Hadoop clusters in the network where the host `acme101.com` is present:

```
emcli show_bda_clusters
        -host="acme101.com"
```

show_applicable_engr_sys_operations

Shows applicable patching operations that can be performed on the given engineered system target.

Format

```
emcli show_applicable_engr_sys_operations
    -system_target_name="system_target_name"
    -system_target_type="system_target_type"
    -target_name="target_name" -target_type="target_type" | -
component_type="component_type"
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- target_name

  Specifies the target name.

- target_type

  Specifies the target type.

- component_type

  Specifies the engineered system component target type.

Examples

The following example displays all of the applicable patching operations for the member target "clusteradm0102.example.com":"cluster" of the engineered system target "slcm12adm01.example.com":"oracle_dbmachine":

```
emcli  show_applicable_engr_sys_operations
    -system_target_name="slcm12adm01.example.com"
    -system_target_type="oracle_dbmachine"
    -target_name="clusteradm0102.example.com"
    -target_type="cluster"
```

show_applicable_engr_sys_patches

Lists all the applicable system patches for a member target of an engineered system.

Format

```
emcli show_applicable_engr_sys_patches
      -system_target_name="system_target_name"
      -system_target_type="system_target_type"
      [-target_name="target_name" -target_type="target_type" | -
input_file=data:"input_file_path"]
      [-show_recommended] | [-show_all]
```

Options

- system_target_name

  Specifies the engineered system target name.

- input_file

  Specifies input file path. The following is an example of an input file:

  slcm12celadm01.example.com:oracle_exadata

  slcm12celadm02.example.com:oracle_exadata

  slcm12celadm03.example.com:oracle_exadata

- system_target_type

  Specifies the engineered system target type.

- target_name

  Specifies the target name.

- target_type

  Specifies the target type.

- show_recommended

  Specifies the engineered system component target type. If you do not specify this option, all the applicable patches will be displayed.

- show_all

  Displays all the applicable patches. This is the default option.

Examples

The following example displays only the recommended patches for the member target "clusteradm0102.example.com":"cluster" of the engineered system " DB Machine slcm12.example.com ":"oracle_dbmachine":

```
emcli  show_applicable_engr_sys_patches
      -system_target_name="DB Machine slcm12.example.com"
      -system_target_type="oracle_dbmachine"
      -target_name="clusteradm0102.example.com"
      -target_type="cluster" -show_recommended
```

ORACLE

show_audit_settings

Shows the following details of the current audit settings:

— Audit Switch

— Externalization Switch

— Directory

— File Prefix

— File Size

— Data Retention Period

Format

```
emcli show_audit_settings
      -view="SUMMARY|DETAIL"
```

show_credential_set_info

Displays the parameters of credential sets defined with target types.

Format

```
emcli show_credential_set_info       [-target_type="<target_type>"]       [-
set_name="<credential_set_name>"]
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **target_type**

    Type of target. The default is to display the credential set defined for all target types.

*   **set_name**

    Name of the credential set. The default is to display all credential sets defined for a target type.

Examples

Example 1

This example displays the details of all credential sets defined with all target types:

```
emcli show_credential_set_info
```

Example 2

This example displays all credential sets defined with the oracle_database target type:

```
emcli show_credential_set_info -target_type=oracle_database
```

show_credential_type_info

Displays the parameters of credential types defined for target types.

Format

```
emcli show_credential_type_info        [-target_type="<target_type>"]        [-
type_name="<credential_type_name>"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **target_type**

    Type of target. The default is to display the credential set defined for all target types.

- **type_name**

    Name of the credential type. The default is to display all credential types defined for a target type.

Examples

Example 1

This example displays the details of the HostUDMCreds credential type defined for the oracle_database target type.

```
emcli show_credential_type_info -target_type=oracle_database
    -type_name=HostUDMCreds
```

Example 2

This example shows output for various credential types.

```
emcli show_credential_type_info -target_type=host
```

```
Target Type   Cred Type Name   Cred Type Column Name   Key Column
host          HostCreds        HostPassword            No
                               HostUserName            Yes
              HostSSHCreds     SSH_PUB_KEY             No
                               SSH_PVT_KEY             No
                               USERNAME                Yes
              ProvisionCreds   InstallPassword         No
                               InstallUserName         Yes
                               OMSRegistrationPassword No
                               ProvCompPasswd          No
              WBEMCreds        WBEMPassword            No
                               WBEMUserName            Yes
```

show_engr_sys_operation_status

Displays the patching operation status of a member target or a component of an engineered system target.

Format

```
emcli show_engr_sys_operation_status
    -system_target_name="system_target_name"
    -system_target_type="system_target_type"
    -target_name="target_name" -target_type="target_type" | -
component_type="component_type"
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- target_name

  Specifies the target name.

- target_type

  Specifies the target type.

- component_type

  Specifies the engineered system component target type.

Examples

Example 1

The following example displays the patching status for the member target "clusteradm0102.example.com":"cluster" in the engineered system "DB Machine slcm12.example.com":"oracle_dbmachine":

```
emcli show_engr_sys_operation_status
      -system_target_name="DB Machine slc12.example.com"
      -system_target_type="oracle_dbmachine"
      -target_name="clusteradm0102.example.com"
      -target_type="cluster"
```

Example 2

The following example displays the patching status for the component "Exadata Grid slcm12.example.com" in the engineered system "DB Machine slcm12.example.com":"oracle_dbmachine":

```
emcli show_engr_sys_operation_status
      -system_target_name="DB Machine slcm12.example.com"
      -system_target_type="oracle_dbmachine"
      -component_type="Oracle Infiniband Switch"
```

show_engr_sys_patchable_targets

Shows the targets of an engineered system for which the given patch is applicable.

Format

```
emcli show_engr_sys_patchable_targets
      -system_target_name="system_target_name"
      -system_target_type="system_target_type"
      -system_patch_composite_id="system_patch_composite_id"
      [-applied="true | false"]

[ ]  indicates that the parameter is optional.
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- system_patch_composite_id

  Specifies the system patch composite ID.

- component_type

  Specifies the engineered system component type.

- applied

  Specifies whether to list the targets on which the patch has already been applied. The possible values for this option are:

  – true: Lists the targets on which the patch is applied.

  – false: Lists only the targets on which the patch hasn't been applied.

  This is the default option.

Examples

The following example displays the member targets of the engineered system " DB Machine slcm12.example.com ":"oracle_dbmachine", on which the patch "p18706488_600000000001381_226_0" has already been applied:

```
emcli  show_engr_sys_patchable_targets
       -system_target_name="DB Machine slcm12.example.com"
       -system_target_type="oracle_dbmachine"
       -system_patch_composite_id="p18706488_600000000001381_226_0"
       -applied="true" -level="target"
```

show_engr_sys_patching_history

Shows the history of recent patching operations for a member target of an engineered system target.

Format

```
emcli show_engr_sys_patching_history
       -system_target_name="system_target_name"
       -system_target_type="system_target_type"
       -target_name="target_name"
       -target_type="target_type"
       [-operation=analyze | deploy | rollback]
       [-numrows="number_rows"]

[ ]  indicates that the parameter is optional.
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- target_name

  Specifies the target name.

- target_type

  Specifies the target name.

- system_patch_composite_id

  Specifies a target type.

- operation

  Specifies the operation for which history will be displayed. The possible values for this option are analyze, deploy, and rollback.

- numrows

  Specifies the number of rows of history to be shown.

Examples

The following example displays history of last 10 'Analyze' operations performed for the member target "clusteradm0102.example.com":"cluster" in the engineered system "slcm12adm01.example.com":"oracle_dbmachine":

```
emcli  show_engr_sys_patching_history
      -system_target_name="slcm12adm01.example.com"
      -system_target_type="oracle_dbmachine"
      -target_name="clusteradm0102.example.com"
      -target_type="cluster"
      -action=analyze
      -numrows="10"
```

show_engr_sys_targets

Lists all the engineered system targets discovered in Enterprise Manager.

Format

```
emcli show_engr_sys_targets
      [-system_target_type="system_target_type"]

[ ]  indicates that the parameter is optional.
```

Options

system_target_name

Specifies the engineered system target name.

Examples

Example 1

The following example displays all the engineered system targets discovered in Enterprise Manager:

```
emcli show_engr_sys_targets
```

Example 2

The following example displays all the engineered system targets of type "oracle_dbmachine":

```
emcli show_engr_sys_targets
      -system_target_type="oracle_dbmachine"
```

show_operations_list

Shows the list of all auditable Enterprise Manager operations names.

Format

```
emcli show_operations_list
```

Output

Output appears as shown in This example:

```
ADD_AGENT_REGISTRATION_PASSWORD
AGENT_REGISTRATION_PASSWORD_USAGE
AGENT_RESYNC
APPLY_TEMPLATE
AUDIT_EXPORT_SETTINGS
AUDIT_SETTINGS
CHANGE_PASSWORD
CHANGE_PREFERRED_CREDENTIAL
CREATE_PG_SCHED
CREATE_ROLE
CREATE_TEMPLATE
CREATE_UDP
CREATE_UDPG
CREATE_USER
DELETE_AGENT_REGISTRATION_PASSWORD
DELETE_JOB
DELETE_PG_EVAL
DELETE_PG_SCHED
DELETE_ROLE
DELETE_TEMPLATE
DELETE_UDP
DELETE_UDPG
DELETE_USER
EDIT_AGENT_REGISTRATION_PASSWORD
EDIT_JOB
EDIT_PG_SCHED
EDIT_TEMPLATE
EDIT_UDP
EDIT_UDPG
EVALUATE_UDP
FILE_TRANSFER
GET_FILE
GRANT_JOB_PRIVILEGE
GRANT_ROLE
GRANT_SYSTEM_PRIVILEGE
GRANT_TARGET_PRIVILEGE
IMPORT_UDP
JOB_OUTPUT
LOGIN
LOGOUT
MODIFY_METRIC_SETTINGS
```

ORACLE®

```
MODIFY_POLICY_SETTINGS
MODIFY_ROLE
MODIFY_USER
PUT_FILE
REMOTE_OPERATION_JOB
REMOVE_PRIVILEGE_DELEGATION_SETTING
REPOSITORY_RESYNC
REVOKE_JOB_PRIVILEGE
REVOKE_ROLE
REVOKE_SYSTEM_PRIVILEGE
REVOKE_TARGET_PRIVILEGE
SAVE_MONITORING_SETTINGS
SET_PRIVILEGE_DELEGATION_SETTING
SUSPEND_JOB
```

show_patch_plan

Shows the details of a particular patch plan.

Format

```
emcli show_patch_plan
        -name="name"
        [-info [-showPrivs]] [-actions [-onlyShowEnabled]]
        [-patches]
        [-targets]
        [-deplOptions]
        [-analysisResults]
        [-conflictFree]
        [-impactedTargets]
        [-deploymentProcedures]

[ ]  indicates that the parameter is optional
```

Options

* **name**

  Plan name. If you only provide this option with no other options, the full details of the patch plan are shown.

* **info**

  Shows the generic information of the given patch plan.

* **show_Privs**

  Shows the user privileges on the given patch plan along with the generic information.

* **actions**

  Show the actions that are possible to be taken on the given patch plan.

* **onlyShowEnabled**

  Only show the enabled actions on the given patch plan.

* **patches**

  Shows details of the patches contained in the given patch plan.

* **targets**

Shows details of the targets contained in the given patch plan.

- **deplOptions**

  Shows details of the deployment options contained in the given patch plan.

- **analysisResults**

  Shows details of the analysis results of the given patch plan.

- **conflictFree**

  Shows details of the conflict-free patches of the given patch plan.

- **impactedTargets**

  Shows details of the impacted targets of the given patch plan.

- **deploymentProcedures**

  Shows the deployment procedure of the given patch plan.

Examples

```
emcli show_patch_plan -name="plan name"

emcli show_patch_plan -name="plan name" -info

emcli show_patch_plan -name="plan name" -actions -onlyShowEnabled

emcli show_patch_plan -name="plan name" -info -showPrivs
```

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data
submit_patch_plan

upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

show_patch_readme

Displays the patch readme.

Format

```
emcli show_patch_readme
      -patch_composite_id= "patch_composite_id"
      [-output_location="output_location"]
```

`[ ]  indicates that the parameter is optional.`

Options

- patch_composite_id

  Specifies the patch composite ID. The format of the patch composite ID must follow "p<PatchID>_<ReleaseID>_<PlatformID>_<LanguageID>".

- output_location

  Specifies the location where the patch readme must be saved. The readme file name is saved in the format "<Patch_Composite_ID>_Readme.html"

Examples

Example 1

The following example displays the readme of the patch "p18706488_600000000001381_226_0":

```
emcli  show_patch_readme
      -patch_composite_id="p18706488_600000000001381_226_0"
```

Example 2

The following example obtains the readme of the patch "p18706488_600000000001381_226_0" and saves the README in the /tmp directory:

```
emcli  show_patch_readme
      -patch_composite_id="p18706488_600000000001381_226_0"
      -output_location=/tmp
```

show_proxy

Shows the details of a HTTP(S) proxy identified by the specified name. You can customize the details shown by using the -details option.

Syntax

```
emcli show_proxy
   -name="<name>"
   [-details=<config | agents | both>]
```

`[ ]  indicates that the parameter is optional.`

Options

- -name

  Name identifying the proxy.

- -details

  Scope of the details to be shown. Valid values:

- config - Proxy configuration only. (default)

- agents - Management Agents associated with the proxy only.

- both - Both proxy configuration and Management Agents associated with the proxy.

Examples

Example 1

The following command shows the configuration of the proxy identified by us-proxy-1.In this example, the "-details=config" option is implicit.

```
emcli show_proxy
      -name="us-proxy-1"
```

Example 2

The following command shows a list of names of all Management Agents associated with the proxy identified by "us-proxy-1". Note that this list includes Management Agents which are associated with the proxy directly by their names as well as by their name patterns.

```
emcli show_proxy
      -name="us-proxy-1"
      -details=agents
```

signoff_agents

Performs Agent sign-off prerequisites and submits the Agent sign-off job.

Format

```
emcli signoff_agents
      -agents="List_of_agents" | -input_file="agents_file:Location of_output file"
      [-job_name="Name_of_job"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **agents**

  Submits a job to clean up old Agent homes matching Agent names or an Agent names pattern separated by commas.

- **input_file**

  Checks whether Agents specified in the file are available for sign-off, and submits the Agent sign-off job.

  You can pass all of these parameters in a response file. The usage is:

  ```
  -input_file="response_file:/scratch/response_file.txt"
  ```

  You must provide the file name with the full path, and each parameter should be given in each line. If you pass a parameter both in the command line and in a response file, the command-line option is given precedence.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **job_name**

  Submits the clean-up job with the job name specified in this option.

Examples

Example 1

This example submits a job to clean up the old Agent homes on Agent names matching the pattern abc% and on the xyz.domain.com Agent.

```
emcli signoff_agents -agents="abc%,xyz.domain.com:1243"
```

Example 2

This example submits a job to clean up the old Agent homes on the Agents specified in the file.

```
emcli signoff_agents -input_file="agents_file:/scratch/agents_file.txt"
```

stage_gold_agent_image

Stages a Management Agent gold image on to a destination host.

Format

```
emcli stage_gold_agent_image
     -version_name="gold_image_version_to_stage"
     -image_name="gold_image_name"
     -host_name="staging_destination_host"
     -stage_location="stage_location_on_destination_host"
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   version_name

    Management Agent gold image version that should be staged.

*   image_name

    Management Agent gold image that should be staged.

*   host_name

    Destination host where the Management Agent gold image should be staged. As a prerequisite, a Management Agent should be running on this host.

*   stage_location

    Location on the destination host where the Management Agent gold image should be staged.

Example

The following example stages the Management Agent gold image `OPC_AGI_DB_JUL_13` of gold image `OPC_AGI_DB`, at the stage location `/net/stage/agent` on the host `example.com`.

```
emcli stage_gold_agent_image
     -version_name=OPC_AGI_DB_JUL_13
     -stage_location=/net/stage/agent
     -host_name=example.com
```

stage_swlib_entity_files

Stages one or more files associated with an entity revision available in the Software Library to a file system location on a host target.

Format

```
emcli stage_swlib_entity_files
      -entity_rev_id="entity_rev_id"
      -host="hostname"
      -file="<file name as specified during upload>"]
       [-credential_set_name="setname"] | [-credential_name="name" -credential_
owner="owner"]
      [-stage_path="<path on host to stage file(s)>"]
      [-use_latest_revision]
      [-overwrite_files]

[ ] indicates that the parameter is optional.
```

Parameters

- entity_rev_id

    Identifier of the entity revision. You can view the entity ID by logging in to the Cloud Control console. The Software Library Home page exposes the identifier for folders and entities as a custom column called Internal ID. By default, this column is hidden.

- host

    Target name of the host where the files are staged.

- file

    Name of the file associated with the entity revision.

- credential_set_name

    The name given to a set of preferred credential stored in the Management Repository for the host target.

    HostCredsNormal - default unprivileged credential set

    HostCredsPriv - privileged credential set

- credential_name

    Named credential stored in the Management Repository. This option

    must be specified along with the -credential_owner option.

- credential_owner

    Owner of a named credential stored in the Management Repository. This option must be specified along with the -credential_name option.

- stage_path

    Location on the host where the files are copied.

- use_latest_revision

    This is an option to indicate whether staging should be carried out for the latest revision of the entity or the revision identified by entity_rev_id.

- overwrite_files

This is an option to indicate whether the file should be overwritten in the stage location or not. By default, the files will not be overwritten. Ensure that the file is overwritten in the stage location.

Example

The following example copies the file 'myfile.zip' associated with the specified entity revision to '/u01/stage_loc' on host 'fs1.example.com' using the named credential "MyAcmeCreds" owned by "ACME_USER".

```
emcli stage_swlib_entity_files
     -entity_rev_id="oracle:defaultService:em:provisioning:1:cmp:COMP_
Component:SUB_Generic:B1B1880C6A8C62AAE040548C42832D14:0.1"Generic:B1B1880C6A8C62
AAE040548C42832D14:0.1"  Generic:B1B1880C6A8C62AAE040548C42832D14:0.1"
    -file="myfile.zip"
    -stage_path="/u01/stage_loc"
    -host="fs1.example.com"
    -credential_name="MyAcmeCreds"
    -credential_owner="ACME_USER"
```

stage_swlib_entity_files_local

Stages one or more files associated with an entity revision available in the Software Library to a file system location on a host target, not monitored by an EM Agent.

Format

```
emcli stage_swlib_entity_files_local
    -entity_rev_id="entity_rev_id"
    [-file="<file name as specified during upload>"]
    [-stage_path="<local file system path to stage the file(s)>"]
    [-use_latest_revision]
    [-overwrite_files]

[ ] indicates that the parameter is optional.
```

Parameters

*   entity_rev_id

    Identifier of the entity revision. The Software Library Home page exposes the identifier for folders and entities as a custom column (Internal ID). However, this is hidden by default.

*   file

    Name of the file associated with entity revision. For staging multiple files of the entity, repeat the option. If this is not specified, all the files of the entity will be staged.

*   stage_path

    Location on the host where the files are to be copied.

*   use_latest_revision

    This is an option to indicate whether staging should be carried out for the latest revision of the entity or the revision identified by the entity_rev_id.

*   overwrite_files

This is an option to indicate whether the file should be overwritten in the stage location or not. By default, files will not be overwritten.

Example

The following example copies the file '`myfile.zip`' associated with the specified entity revision to '`/u01/stage_loc`' on the local file system.

```
emcli stage_swlib_entity_files_local
        -
entity_rev_id="oracle:defaultService:em:provisioning:1:cmp:COMP_Component:SUB_
Generic:B1B1880C6A8C62AAE040548C42832D14:0.1"
    -file="myfile.zip"
    -stage_path="/u01/stage_loc"
```

start_agent

Starts up a Management Agent. This verb requires operator privilege or full privilege on the Management Agent.

Format

```
emcli start_agent
        -agent_name="agent_target_name"
        [-host_username="agent_host_username" -host_pwd="agent_host_password"]
        [-credential_name="credential_name"]
        [-credential_setname="credential_setname_of_agent"]

[ ]  indicates that the parameter is optional
```

Options

*   **agent_name**

    Name of the Management Agent target.

*   **host_username**

    User name of the OS user (on the host) who owns the Management Agent.

*   **host_pwd**

    Password of the OS user (on the host) who owns the Management Agent.

*   **credential_name**

    Name of the saved credential.

*   **credential_setname**

    Name of the credential set of the Management Agent. Example: "HostCreds".

Examples

Example 1

```
emcli start_agent -agent_name="agent.example.com:1234"
                          -host_username="test_user"
                          -host_pwd="test"
```

Example 2

```
emcli start_agent -agent_name="agent.example.com:1234"
                          -credential_name="MyMachineCredential"
```

start_mda_engine

Starts the MDA engine. By default the command starts the MDA engine with five processors. Optionally, add the processor_count parameter to change the number of processors.

Format

```
emcli start_mda_engine
      [-processors_count=<count>]
```

[ ] indicates that the parameter is optional.

Parameter

* processors_count

    Starts the MDA engine with the specified number of processors.

Example

The following example starts the MDA engine with six processors:

```
emcli start_mda_engine
      -processors_count=6
```

start_replay

Start a replay in the specified target database.

Sample XML File:

```
<?xml version="1.0" encoding="UTF-8"?>
      <replay xmlns="http://xmlns.oracle.com/sysman/db/dbreplay/replay">
        <taskName>tkSiTask</taskName>
        <name>tkSiTrial_3</name>
        <replayTargetName>database</replayTargetName>
        <replayTargetType>oracle_database</replayTargetType>
        <dbHostName>slc00tny.mycompany.com</dbHostName>
        <dbCredentialReference>
          <credName>TESTDB121</credName>
          <credOwner>SYSMAN</credOwner>
        </dbCredentialReference>
        <dbHostCredentialReference>
          <credName>TESTDBHOST121</credName>
          <credOwner>SYSMAN</credOwner>
        </dbHostCredentialReference>
        <directory>/storage/captures/DBReplayWorkload_captureRepos_10Mins_3_185</
directory>
        <consolidated>false</consolidated>
        <captures>
          <capture>
            <captureName>captureRepos_10Mins_3</captureName>
            <startTime>1395746473000</startTime>
          </capture>
        </captures>
        <synchronization>OBJECT_ID</synchronization>
```

```
            <connectTimeScale>100</connectTimeScale>
            <thinkTimeScale>100</thinkTimeScale>
            <thinkTimeAutoCorrect>true</thinkTimeAutoCorrect>
            <stsEnabled>true</stsEnabled>
            <minimumClients>1</minimumClients>
            <maximumWaitSeconds>10</maximumWaitSeconds>
            <replayConnectionMappings>
              <replayConnectionMapping workloadId="2">
                <connectionMethod>SHARED_DESCRIPTOR</connectionMethod>
                <sharedDescriptor>(DESCRIPTION=(ADDRESS_LIST = (ADDRESS=(PROTOCOL=tcp)
(HOST=slc00tny.mycompany.com)(PORT=15045)))(CONNECT_DATA=(SID=ttny2)
(SERVER=DEDICATED)))</sharedDescriptor>
              </replayConnectionMapping>
              <replayConnectionMapping workloadId="1">
                <connectionMethod>SHARED_DESCRIPTOR</connectionMethod>
                <sharedDescriptor>(DESCRIPTION=(ADDRESS_LIST = (ADDRESS=(PROTOCOL=tcp)
(HOST=slc00tny.mycompany.com)(PORT=15045)))(CONNECT_DATA=(SID=ttny2)
(SERVER=DEDICATED)))</sharedDescriptor>
              </replayConnectionMapping>
            </replayConnectionMappings>
            <clientHostConfigurations>
              <clientHostConfiguration>
                <hostName>slc00tny.mycompany.com</hostName>
                <hostOsType>Linux</hostOsType>
                <clientHostCredentialReference>
                  <credName>TESTHOST</credName>
                  <credOwner>SYSMAN</credOwner>
                </clientHostCredentialReference>
                <replayDbName>database</replayDbName>
                <replayDbType>oracle_database</replayDbType>
                <replayDbCredentialReference>
                  <credName>TESTDB121</credName>
                  <credOwner>SYSMAN</credOwner>
                </replayDbCredentialReference>
                <clients>1</clients>
                <serverConnectionIdentifier>slc00tny.mycompany.com:15045/
ttny2.regress.rdbms.dev.mycompany.com</serverConnectionIdentifier>
                <clientOracleHome>/storage/oracle</clientOracleHome>
                <clientReplayDirectory>/storage/db12/captures/
DBReplayWorkload_captureRepos_10Mins_3_185</clientReplayDirectory>
              </clientHostConfiguration>
            </clientHostConfigurations>
          </replay>
```

Format - Standard Mode

```
emcli start_replay
        [-input_file="template:<input file path>"]
```

[ ] indicates that the parameter is optional.

Format - Interactive or Script Mode

```
start_replay(
        [input_file="template:<input file path>"]
        )
```

[ ] indicates that the parameter is optional.

Options

- input_file

    Fully qualified path to an XML file containing parameters for the verb.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1 - Standard Mode

The following example starts a replay.

```
emcli start_replay      -input_file=template:/storage/xml/newReplay.xml
```

Example 2 - Interactive or Script Mode

The following example starts a replay.

```
start_replay(input_file="template:/storage/xml/newReplay.xml"      )
```

status

Shows whether EM CLI is configured or not, and shows the EM CLI setup details. It also displays the Java home, version, EM CLI home, and all of the EM CLI configuration detials if it is configured.

Standard Mode

```
emcli status
```

Interactive or Script Mode

```
status()
```

Options

None.

Output

This example shows output when EM CLI setup has not been done:

```
Oracle Enterprise Manager Cloud Control 12c Release 12.1.0.0.0.
Copyright (c) 1996, 2011 Oracle Corporation and/or its affiliates. All rights
reserved.

Instance Home : /home/sumadas
Status        : Not Configured
```

This example shows output after EM CLI setup has been done:

```
Oracle Enterprise Manager Cloud Control 12c Release 12.1.0.0.0.
Copyright (c) 1996, 2013 Oracle Corporation and/or its affiliates. All rights
reserved.

Instance Home           : /ade/sumadas_emcli/oracle/work/.emcli
```

```
Status                : Configured
EMCLI Home            : /ade/sumadas_emcli/emcore/emcli/bin
EMCLI Version         : 12.1.0.0.0
Java Home             : /ade_autofs/nfsdo_base/EMGC/MAIN/LINUX/110811/jdk6/jre
Java Version          : 1.6.0_24
Log file              : /ade/sumadas_emcli/oracle/work/.emcli/.emcli.log
EM URL                : https://dadvma0121.example.com:14487/em
EM user               : SYSMAN
Auto login            : true
Trust all certificates : true
```

This example shows output in interactive shell mode:

```
emcli>status()
Oracle Enterprise Manager 12c EM CLI with Scripting option Version 12.1.0.3.0.
Copyright (c) 1996, 2013 Oracle Corporation and/or its affiliates. All rights reserved.

Verb Jars Home (EMCLI_VERBJAR_DIR)      : <EMCLI_LOCATION>/int/./bindings/
12.1.0.3.0/.emcli
EM CLI Home (EMCLI_INSTALL_HOME)        : <EMCLI_LOCATION>/int/.
EM CLI Version                          : 12.1.0.3.0
Java Home                               : /jdk6/jre
Java Version                            : 1.6.0_43
Log file (EMCLI_LOG_LOC)                : CONSOLE
Log level (EMCLI_LOG_LEVEL)             : SEVERE
EM URL (EMCLI_OMS_URL)                  : https://<hostname>:<port>/em
EM user (EMCLI_USERNAME)                : sysman
Auto login (EMCLI_AUTOLOGIN)            : false
Trust all certificates (EMCLI_TRUSTALL) : true
```

stop_agent

Shuts down a Management Agent. This verb requires operator privilege or full privilege on the Agent.

Format

```
emcli stop_agent
        -agent_name="agent_target_name"
        [-host_username="agent_host_username" -host_pwd="agent_host_password"]
        [-credential_name="credential_name"]
        [-credential_setname="credential_setname_of_agent"]
```

[ ] indicates that the parameter is optional

Options

• **agent_name**

  Name of the Management Agent target.

• **host_username**

  User name of the OS user (on the host) who owns the Management Agent.

• **host_pwd**

  Password of the OS user (on the host) who owns the Management Agent.

• **credential_name**

  Name of the saved credential.

- **credential_setname**

  Name of the credential set of the Management Agent. Example: "HostCreds".

Examples

Example 1

```
emcli stop_agent -agent_name="agent.example.com:1234"
                         -host_username="test_user"
                         -host_pwd="test"
```

Example 2

```
emcli stop_agent -agent_name="agent.example.com:1234"
                         -credential_name="MyMachineCredential"
```

stop_blackout

Stops a blackout.

You can stop a blackout before it has fully started, for example, when it has a "Scheduled" status. You can also stop a blackout while it is in effect.

Format

```
emcli stop_blackout
      -name="name"
      [-createdby="blackout_creator"]
      [-emd_url="emd_url"]
      [-max_wait_duration="maximum duration in seconds]
```

[ ] indicates that the parameter is optional. Note that the emd_url parameter is mandatory for blackouts created through emctl.

Options

- name

  Name of the blackout to stop.

- createdby

  Enterprise Manager user who created the blackout. The default is the current user. The SUPER_USER privilege is required to stop a blackout created by another user.

- emd_url

  The emd_url of the agent through which the emctl blackout was created. This is a mandatory parameter for blackouts created through emctl.

- max_wait_duration

  The maximum duration (seconds) to wait for the blackout status to be updated.

Examples

Example 1

This example stops blackout `backup_db3` created by the current user.

```
emcli stop_blackout -name=backup_db3
```

Example 2

This example stops blackout `weekly_maint` created by user `joe`. The current user must either be user `joe` or a user with the `SUPER_USER` privilege.

```
emcli stop_blackout -name=weekly_maint -createdby=joe
```

stop_engr_sys_maintenance

Stops a maintenance operation for a member target or a list of member targets of an engineered system target.

Format

```
emcli stop_engr_sys_maintenance
      -system_target_name="system_target_name"
      -system_target_type="system_target_type"
      -target_name="target_name" -target_type="target_type" | -
input_file=data:"input_file_path"
      -operation=analyze | deploy | rollback

[ ]  indicates that the parameter is optional.
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- target_name

  Specifies the target name.

- target_type

  Specifies the target type.

- input_file

  Specifies the input file path.

- operation

  Specifies the operation to be stopped. The possible values for this option are analyze, deploy, and rollback.

Examples

The following example stops the deploy operation for the member target "clusteradm0102.example.com":"cluster" of the engineered system "slcm12adm01.example.com":"oracle_dbmachine":

```
emcli stop_engr_sys_maintenance
      -system_target_name="slcm12adm01.example.com"
      -system_target_type="oracle_dbmachine"
      -target_name="clusteradm0102.example.com"
      -target_type="cluster" -operation=deploy
```

stop_instance

Stops a scheduled, failed, or running deployment instance.

Format

```
emcli stop_instance
      [-instance=<instance_guid>]
      [-exec=<execution_guid>]
      [-name=<execution_name>]
      [-owner=<execution_owner>]
```

Options

- **instance**

  GUID of the instance.

- **exec**

  GUID of the execution.

- name

  Name of the execution.

- **owner**

  Owner of the execution.

Examples

```
emcli stop_instance -instance=16B15CB29C3F9E6CE040578C96093F61
```

stop_job

Stops a specified job. You can use the `get_jobs` verb to obtain a list of job IDs and names.

Format

```
emcli stop_job
      [-job_id="ID1;ID2;..."]
      [-name="job_name_pattern"]
      [-owner="job_owner"]
      [-type="job_type"]
      [-targets="target_name:target_type"]
      [-input_file=property_file:"filename"]
      [-preview]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **job_id**

  Semi-colon (;) separated list of job(s) to stop.

  Note: This filter cannot be used with other filters.

- **name**

Name or pattern of the job(s) to stop.

- **owner**

  Owner of the job(s).

- **type**

  Job type of the job(s).

- **targets**

  Targret name and target type of the job(s) to stop.

- **input_file**

  The properties for filtering jobs may be specified in "filename."

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **preview**

  Lists only the jobs to stop.

Examples

Example 1

This example stops a job with the specified ID.

```
emcli stop_job -job_id=123456789012345678901234567889012
```

Example 2

This example stops all jobs owned by the Administrator "Jennifer".

```
emcli stop_job -owner=Jennifer
```

stop_mda_engine

Stops the MDA engine on the current OMS instance.

Format

```
emcli stop_mda_engine      [-persist=true|false]
[ ]  indicates that the parameter is optional.
```

Options

- persist

  If persist is set to true, the engine will not be started on OMS restart. The default value for the persist option is false, for example if the persist option is not specified, the engine will be restarted on OMS restart.

Examples

Example 1

The following example stops the MDA engine without the persist attribute defined. The MDA engine will restart on OMS restart.

```
emcli stop_mda_engine
```

Example 2

The following example stop the MDA engine with the persist attribute defined. The MDA engine will not restart on OMS restart.

```
emcli stop_mda_engine
      -persist=true
```

stop_siteguard_health_checks

Retrieves and shows the configured limit for Apply lag and Transport lag for all or selected databases of the system.

Format

```
emcli stop_siteguard_health_checks
        [-operation_plan=name_of_the_operation_plan]

[ ]  indicates that the parameter is optional
```

Parameter

* **operation_plan**

  Name of the operation plan for which execution of health checks must be stopped.

Examples

Example 1

This example displays the details of the Apply lag limit configured on all of the databases of the system `austin-system`:

```
emcli get_siteguard_lag
      -system_name="austin-system"
      -property_name="ApplyLag"
```

Example 2

This example stops health checks for operation plan `austin-switchover`:

```
emcli stop_siteguard_health_checks
      -operation_plan="austin-switchover"
```

submit_add_host

Submits an Add Host session that installs management Agents on unmanaged hosts, thereby converting them to managed hosts.

Format

```
emcli submit_add_host
      -host_names=<host_list>        -platform=<platform_id>      -
installation_base_directory=<installation_base_directory>      -
credential_name=<credential_name>        [-
instance_directory=<instance_directory>]        [-
credential_owner=<credential_owner>]        [-
properties_file=<properties_file>]        [-
session_name=<deployment_session_name>]        [-
privilege_delegation_setting=<privilege_delegation_setting>]        [-
port=<agent_port>]        [-deployment_type=FRESH|SHARED|CLONE]        [-
preinstallation_script=<preinstallation_script_location>]        [-
```

```
preinstallation_script_on_oms]        [-preinstallation_script_run_as_root]        [-
postinstallation_script=<postinstallation_script_location>]        [-
postinstallation_script_on_oms]        [-postinstallation_script_run_as_root]        [-
additional_parameters=<parameter1 parameter2 parameter3 .... >]        [-
wait_for_completion]        [-source_agent=<clone_source_agent_name>]        [-
master_agent=<master_agent_name>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **host_names**

  Names of the hosts where the Agents need to be installed, separated by a semi-colon.

- **platform**

  ARU platform ID of the hosts where the Agent needs to be installed. To show the list of supported agent platforms, run the command emcli list_add_host_platforms -all.

- **installation_base_directory**

  Directory where you want to install the Agent. Provide this option in double-quotes if it is an MS-DOS/Windows-style path.

- **credential_name**

  Named credential to be used for installing the Agent.

- **instance_directory**

  Instance directory of the Agent. Provide this option in double-quotes if it is an MS-DOS/Windows-style path.

- **credential_owner**

  Owner of the named credential owner.

- **session_name**

  Session name that uniquely identifies the Add Host session.

- **privilege_delegation_setting**

  Privilege delegation setting you want to use to install an Agent and run the root script.

- **port**

  Port on which the Agent should communicate with the OMS.

- **deployment_type**

  Type of Agent deployment, which can be FRESH, CLONE, or SHARED. The default is FRESH.

- **preinstallation_script**

  Script you want to run before installing the Agent. Provide this option in double-quotes if it is an MS-DOS/Windows-style path.

- **preinstallation_script_on_oms**

  Use this option if the pre-installation script resides on the OMS host.

- **preinstallation_script_run_as_root**

  Use this option if you want to run the pre-installation script as the root user.

- **postinstallation_script**

Script to run after installing the Agent. Provide this option in double-quotes if it is an MS-DOS/Windows-style path.

- **postinstallation_script_on_oms**

  Use this option if the post-installation script resides on the OMS host.

- **postinstallation_script_run_as_root**

  Use this option if you want to run the post-installation script as the root user.

- **additional_parameters**

  Additional parameters you want to use to install an Agent.

- **wait_for_completion**

  Runs the Add Host operation synchronously. If you specify this option, the command waits until the add host session completes before returning control to you on the command line.

- **source_agent**

  Source Agent you want to use to install a cloned Agent. The source Agent name should have the format of "agent host name:agent port". For example: foo.example.com:3872 .

- **master_agent**

  Master Agent you want to use to install a shared Agent. The master Agent name should have the format of "agent host name:agent port". For example: foo.example.com:3872 .

Examples

Example 1

This example submits an Add Host session on the host 'example.com', having platform ID '226' with '/opt/agent' as the installation base directory,using the named credential 'oracle' and privilege delegation setting /usr/bin/sudo -u %RUNAS% %COMMAND%.

```
emcli submit_add_host -host_names="example.com" -platform=226 -
credential_name=oracle -installation_base_directory=/opt/agent -
privilege_delegation_setting="/usr/bin/sudo -u %RUNAS% %COMMAND%"
```

Example 2

This example submits an Add Host session on the host 'example2.com', having platform ID '233' with 'C:\agent' as the installation base directory, and using the named credential 'oracle'.

```
emcli submit_add_host -host_names=example2.com -platform=233 -
installation_base_directory="C:\agent" -credential_name=oracle
```

submit_engr_sys_maintenance

Submits a maintenance operation for one or more member targets or a component of an engineered system target.

Format

```
emcli submit_engr_sys_maintenance
      -system_target_name="system_target_name"
      -system_target_type="system_target_type"
```

```
-operation=analyze | deploy | rollback
-component_type="component_type" | -input_file=data:"input_file_path"
```

Options

- system_target_name

  Specifies the engineered system target name.

- system_target_type

  Specifies the engineered system target type.

- operation

  Specifies the operation that must be submitted. The possible values for this option are analyze, deploy and rollback.

- component_type

  Specifies the engineered system component type.

- input_file

  Specifies the input file path. The input data must be provided in property name-value pairs and the add_engr_sys_patches verb should have been executed. The following is an example of an input file:

```
target.0.target_name=slcm12adm01.example.com
target.0.target_type=oracle_exadata
target.1.target_name=slcm12adm02.example.com
target.1.target_type=oracle_exadata
target.2.target_name=slcm12adm03.example.com
target.2.target_type=oracle_exadata
```

Examples

The following example uses the input file "/tmp/inputprop1.prop" to submit an analysis operation on the engineered system target "slcm12adm01.example.com":"oracle_dbmachine":

```
emcli  submit_engr_sys_maintenance
       -system_target_name="slcm12adm01.example.com"
       -system_target_type="oracle_dbmachine"
       -operation=analyze
       -input_file=data:"/tmp/inputprop1.prop"
```

submit_job

Creates and submits a job. This verb has been deprecated in favor of create_job. For more information, refer to this verb in this chapter, or enter:

```
emcli help create_job
```

submit_masking_job

Submits a masking job and returns the display job ID and execution ID.

Format

```
emcli submit_masking_job
    -definition_name=<masking_defn_name>
```

```
-target_name=<database_target_name>
-target_type=<database_target_type>
-parameters=name1:value1;name2:value2;...
[-host_pref_creds_name=<preferred_host_credentials_name>
  OR -host_cred_name=<host_credential_name>]
[-db_pref_creds_name=<preferred_db_credentials_name>
  OR -db_cred_name=<db_credential_name>]
[-encryption_key=<encryption_key_string]
[-script_file_location=<script_file_location>]
[-script_file_name=<script_file_name>]
[-input_file=PWD_FILE_TAG:<credentials_file_name>]
[-script | -format=[name:<pretty|script|csv>];
    [column_separator:"column_sep_string"];
    [row_separator:"row_sep_string"];
]
```

```
[ ]  indicates that the parameter is optional
```

> **✎ Note:**
>
> Unless values for the Host and DB credentials are specified in the -parameters parameter, either the host_pref_creds_name or host_cred_name parameter should be specified. Similarly, either the db_pref_creds_name or the db_cred_name parameter should be specified.

Options

- **definition_name**

  Masking definition name.

- **target_name**

  Database target name to mask.

- **target_type**

  Database target type to mask.

- **parameters**

  List of name-value pairs that represent the credentials required to connect to the database instance. The supported parameters are 'db_username', 'db_password', 'db_role', 'db_cred_name', 'host_username', 'host_password', and 'host_cred_name'. If PDP needs to be used, additional parameters to be specified are 'PDP','RUNAS', and 'PROFILE'. The 'PROFILE' option is only applicable for Powerbroker.

- **host_pref_creds_name**

  Type of preferred credentials to use to connect to the database host, which can either be HostCredsNormal or HostCredsPriv.

- **host_cred_name**

  Credential name to use to connect to the database host.

- **db_pref_creds_name**

  Type of preferred credentials to use to connect to the database instance, which can either be DBCredsNormal or DBCredsSYSDBA.

- **db_cred_name**

  Credential name to use to connect to the database instance.

- **encryption_key**

  Specify an encryption key if the masking definition involves usage of a substitute format.

- **script_file_location**

  Location where the SQL script is to be copied and executed. Default values of $ORACLE_HOME/dbs are used if a value is not specified.

- **script_file_name**

  Name of the script file to store the masking SQL script. If you do not specify a name, a system-generated file name is used.

- **input_file**

  Used in conjunction with the 'parameters' option, this option enables you to store parameter values, such as user name and password, in a separate file. The 'input_file' option specifies a mapping between a tag and a local file path. The tag is specified in lieu of specific parameter values of the 'parameters' option. You can specify multiple -input_file parameters. The result would be a combination of all of the files.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **script**

  This option is equivalent to -format="name:script" .

- **format**

  Format specification (default is `-format="name:pretty"`).

  – `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  – `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  – `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  – `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  – row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Examples

Example 1

This example submits a masking job for the definition name MASKING_DEF and returns the job ID and execution ID.

```
emcli submit_masking_job -definition_name=MASKING_DEF -target_name=testdb -
target_type=oracle_database -
parameters="db_username:sys;db_password:password;db_role:SYSDBA;db_cred_name:DBCREDS;ho
st_username:test;host_password:password;host_cred_name:HOSTCREDS"
```

Example 2

This example takes the credentials from the provided input files host_creds.txt and db_creds.txt.

```
emcli submit_masking_job -definition_name=MASKING_DEF -target_name=testdb -
target_type=oracle_database  -parameters="HOST_CREDS;DB_CREDS" -
input_file=HOST_CREDS:host_creds.txt -input_file=DB_CREDS:db_creds.txt
```

It is also possible to specify both of the credentials in one file and use only one -input_file tag. If PDP must be used, you must provide values in the parameters/input_file as follows:

- SUDO:

  ```
  db_username:sys;db_password:password;db_role:SYSDBA;host_username:user2;host_
  password:password;PDP:SUDO;RUNAS:user1
  ```

- POWERBROKER:

  ```
  db_username:sys;db_password:password;db_role:SYSDBA;host_username:user2;host_
  password:password;PDP:POWERBROKER;RUNAS:user1;PROFILE:profile
  ```

submit_operation_plan

Submits the specified operation plan for execution.

Format

```
emcli submit_operation_plan
     -name=<operation_plan_name>
     [-run_prechecks=true|false]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the operation plan.

- **run_prechecks**

  Optionally run pre-checks by specifying either true or false.

Examples

```
emcli submit_operation_plan
     -name="austin-switchover"
     -run_prechecks="true"
```

See Also

create_operation_plan
get_operation_plans

submit_patch_plan

Submits action on a given patch plan, such as analyzing, preparing, deploying, and switchbacking, or finds the next action automatically, then runs it.

Format

```
emcli submit_patch_plan
        -name="name"
        -action="action name"
```

Options

- **name**

  Patch plan name.

- **action**

  Action to submit on the given patch plan.

Examples

```
emcli submit_patch_plan -name="plan name"

emcli submit_patch_plan -name="plan name" -action="analyze"
```

See Also

> create_patch_plan
> delete_patches
> describe_patch_plan_input
> get_connection_mode
> get_patch_plan_data
> list_aru_languages
> list_aru_platforms
> list_aru_products
> list_aru_releases
> list_patch_plans
> search_patches
> set_connection_mode
> set_patch_plan_data
> show_patch_plan

upload_patches

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

submit_procedure

Submits a deployment procedure or a pre-saved procedure configuration.

Format

```
emcli submit_procedure
      -input_file=data:"file_path"
      [-procedure="procedure_guid"]
      [-name="procedure_name"]
      [-owner="procedure_owner"]
      [-parent_proc="procedure_of_procedure_config"]
      [-instance_name="procedure_instance_name"]
```

```
            [-grants="users_and_their_corresponding_access_levels"]
            [-schedule=
                start_time:yyyy/MM/dd HH:mm;
                tz:{java timezone ID};
                grace_period:xxx;
            ]
```

[ ] indicates that the parameter is optional

Options

- **input_file**

  Input data for the Deployment Procedure. The `file_path` should point to a file containing the data properties file.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **procedure**

  GUID of the procedure to execute.

- **name**

  Name of the procedure or procedure configuration.

- **owner**

  Owner of the procedure or procedure configuration.

- **parent_proc**

  Procedure of the procedure configuration, this applies to a procedure configuration when there is both a procedure and a procedure configuration with the same name.

- **instance_name**

  Name of the procedure instance.

- **grants**

  Users and their corresponding access levels designated as a string of user:privilege pairs each separated by ; .

  where:

  user = Enterprise Manager user name

  privilege = VIEW_JOB or FULL_JOB

- **schedule**

  Schedule for the deployment procedure. If not specified, the procedure is executed immediately.

  `start_time` — When the procedure should start

  `tz` — Optional time zone ID

Output Columns

Instance GUID

Examples

```
emcli submit_procedure -input_file=data:data.properties -
procedure=16B15CB29C3F9E6CE040578C96093F61 -grants="user1:VIEW_JOB;user2:
FULL_JOB" -schedule="start_time:2006/6/21 21:23;tz:America/New_York;
grace_period:60" -instance_name="MyProcedureInstance_0001"
```

For more information on the procedures to deploy, redeploy, and undeploy a Java EE application using EMCLI, see Deploying / Redeploying / Undeploying Java EE Applications Using EMCLI in the *Enterprise Manager Lifecycle Management Administrator's Guide.*

subscribe_agents

Subscribes the specified Management Agents to the specified Management Agent gold image.

Format

```
emcli subscribe_agents
      -image_name="Image Name"
      [-agents="agent_name_pattern"]
      [-groups="group_name"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

* image_name

  Management Agent gold image to which the Management Agents should subscribe.

* agents

  Name pattern of the Management Agents that should subscribe to Management Agent gold image.

* groups

  Group of the Management Agents that should subscribe to the Management Agent gold image.

Examples

Example 1

The following example subscribes the Management Agents that match the name pattern `abc%` or `xyz.domain.com:1243` to the Management Agent gold image `OPC_AGT_ADC_POD`.

```
emcli subscribe_agents
      -image_name="OPC_AGT_ADC_POD"
      -agents="abc%,xyz.domain.com:1243"
```

Example 2

The following example subscribes all the Management Agents to the Management Agent gold image `OPC_AGT_ADC_POD`.

```
emcli subscribe_agents
      -image_name="OPC_AGT_ADC_POD"
```

subscribeto_rule

Subscribes the user to a rule with email notification.

It is not an error to specify email addresses that are already in the `assignto` user's preferences.

A message appears if the outgoing mail server (SMTP) has not been set up. When you specify the `-fail_if_no_mail_server`, this condition is an error and prevents the subscribe from occurring; otherwise, this condition is a warning that does not affect the success of this command.

Format

```
emcli subscribeto_rule
        -ruleset_name="ruleset_name"
        -rule_name="rule_name"
        -owner="rule_owner"
        [-assignto="em_username"]
        [-email="email_address";...]
        [-fail_if_no_mail_server]

[ ]  indicates that the parameter is optional
```

Options

- **ruleset_name**

  Name of the incident rule set.

- **rule_name**

  Name of the rule.

- **owner**

  Owner of the rule set.

- **assignto**

  User to subscribe to the notification rule. If the `assignto` user is not the current user, or if the owner of the rule is not the current user, the super-user privilege is needed. The default is the current user.

- **email**

  List of email addresses to associate with the rule to which the `assignto` user is being subscribed. These addresses are first added to the preferences of the `assignto` user (duplicates are ignored) before being assigned to the notification rule. The email addresses are added only if the current user has the privilege to subscribe the `assignto` user to the rule.

- **fail_if_no_mail_server**

  A message appears if the outgoing mail server (SMTP) has not been set up. When you specify the `-fail_if_no_mail_server` option, this condition is an error and prevents the subscribe from occurring; otherwise, this condition is a warning that does not affect the success of this command.

Examples

Example 1

This example subscribes the current user to the rule "`Agent Upload Problems`" using the current user's email addresses for notification. The current user must have the `SUPER_USER` (or have `sysman`) privilege for this to succeed, since `sysman` owns the rule. Also, the current user must already have at least one email address in his/her preferences for this command to succeed.

```
emcli subscribeto_rule -name="Agent Upload Problems" -owner=sysman
```

Example 2

This example first adds the two specified email addresses to the preferences for user `joe`. Then user `joe` is subscribed to the rule "`Agent Upload Problems`" using joe's email addresses for notification. The current user must have `SUPER_USER` privilege (or be `joe`) for this command to succeed.

```
emcli subscribeto_rule -name="Agent Upload Problems" -owner=sysma
      -assignto=joe -email="joe@work.com;joe@home.com"
```

suppress_compliance_rule_violations

Suppress the violations by the `cs_iname`, `author`, `version`, `rule iname`, `target type`, and `target name`. Unsuppress compliance rule violations by providing the `suppress_until` field. Optionally, you can provide comma-separated violation GUIDs.

Format

```
emcli suppress_compliance_rule_violations
 -cs_iname=<standard internal name>
 -author=<standard author>
 -version=<standard version>
 -rule_iname=<rule internal name>
 -target_type=<target type>
 -target_name=<target name>
 [-violation_guid_list=<list of violation guids separated by comma]
 [-suppress_until=<suppress until date with date format MM-DD-YYYY>]
 [-reason=<reason for violations suppression>]
```

Options

- cs_iname

  The name of the compliance rule violation suppresion.

- author

  The author that created this violation suppression.

- version

  The version number of the violation suppression.

- rule_iname

  The name of the rule this violation supression belongs to.

- target_type

The target type that is affected by this violation suppression.

- target_name

    The target name that is affected by this violation suppression.

- violation_guid_list (Optional)

    A comma-separated list that contains the compliance rule violation GUIDs.

- suppress_until (Optional)

    Unsuppress compliance rule violations by using this flag. The date is in format `"MM-DD-YYYY"`.

- reason (Optional)

    The reason this compliance rule violation suppression is active.

Examples

The following command suppresses the compliance rule violations with GUIDs: `violation_guid1,violation_guid2,violation_guid3` until `05-27-2020`.

```
emcli suppress_compliance_rule_violations
  -cs_iname="mycs"
  -author="myusername"
  -version="2"
  -rule_iname="myrule"
  -target_type="myhost"
  -target_name="mymachine"
  -violation_guid_list="violation_guid1,violation_guid2,violation_guid3"
  -suppress_until="05-27-2020"
  -reason="It has been approved"
```

suppress_incident

Suppresses an incident. Incidents can be suppressed until the suppression type is met.

Format

```
emcli suppress_incident
    -incident_id="id of the incident to be suppressed"
    -suppress_type="Type of suppression"
   [-date="Date argument in MMDDYYYY format"]
```

```
[ ]  indicates that the parameter is optional.s
```

Options

- incident_id

    ID of the incident to be suppressed.

- suppress_type

    Type of suppression. It must be one of the following:

    – UNTIL_MANUALLY_REMOVED: Incident will be suppressed until the user manually unsuppress the incident.

    – UNTIL_SPECIFIED_DATE: Incident will be suppressed until the specified date

– UNTIL_SEVERITY_CHANGE: Incident will be suppressed until its severity changes.

– UNTIL_CLEARED: Incident will be suppressed until it gets cleared.

- date (Optional)

  The date option is only used if an incident is to be suppressed until a specified date. If the *suppress_type* is UNTIL_SPECIFIED_DATE, then the date *until* the incident needs to be suppressed must be provided in MMDDYYYY format.

Examples

Example 1

The following command suppresses the incident with ID 173 until December 31, 2015.

```
 emcli suppress_incident -incident_id="173" suppress_type="UNTIL_SPECIFIED_DATE" -
date="12312015"
```

Example 2

The following command suppresses the incident with ID 173 until an administrator manually unsuppresses the incident.

```
emcli suppress_incident -incident_id="173" suppress_type="UNTIL_MANUALLY_REMOVED"
```

suppress_problem

Suppresses a problem until it meets the suppression type criteria.

Format

```
emcli suppress_problem
        -problem_id="problem ID"
        -suppress_type="Type of suppression"
        [-date="MMDDYYYY"]

[ ]  indicates that the parameter is optional.
```

Options

- problem_id

  ID of the problem to be suppressed.

- suppress_type

  Type of suppression. It must be one of the following:

  – UNTIL_MANUALLY_REMOVED: Problem will be suppressed until the user manually unsuppress the problem.

  – UNTIL_SPECIFIED_DATE: Problem will be suppressed until specified date.

  – UNTIL_SEVERITY_CHANGE: Problem will be suppressed until its severity change.

  – UNTIL_CLEARED: Problem will be suppressed until it gets cleared.

- date (Optional)

  Used when a problem needs to be suppressed until a specific date.When the *suppress_type* is set to UNTIL_SPECIFIED_DATE, you use the *date* option to specify the date when the problem suppression ends. MMDDYYYY format must be used.

Examples

Example 1

The following command suppresses a problem with the ID 173 until December 31, 2015.

```
emcli suppress_problem -problem_id="173" suppress_type="UNTIL_SPECIFIED_DATE" -
date="12312015"
```

Example 2

The following command suppresses a problem with ID 173 until it manually cleared by an administrator.

```
emcli suppress_problem -problem_id="173" suppress_type="UNTIL_MANUALLY_REMOVED"
```

suspend_instance

Suspends a running deployment instance.

Format

```
emcli suspend_instance
      -instance=<instance_guid>
      [-exec=<execution_guid>]
      [-name=<execution_name>]
      [-owner=<execution_owner>]

[ ]  indicates that the parameter is optional
```

Options

*   **instance**

    GUID of the instance.

*   **exec**

    GUID of the execution.

*   **name**

    Name of the execution.

*   **owner**

    Owner of the execution.

Examples

```
emcli suspend_instance -instance=16B15CB29C3F9E6CE040578C96093F61
```

suspend_job

Suspends a job or set of jobs matching the filter criteria. Executions on any of the targets and scheduled to start within the beginning and ending time window are suspended.

Format

```
emcli suspend_job
   [-name="job_name_pattern"]
   [-owner="job_owner"]
   [-type="job_type"]
   [-targets="target_name:target_type"]
   [-input_file=property_file:"filename"]
   [-preview]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name or pattern of the job(s) to suspend.

- **owner**

  Owner of the job(s).

- **type**

  Job type of the job(s).

- **targets**

  Target name and target type of the job(s).

- **input_file**

  Specify the filtering properties of the file in "filename."

  Any jobs matching all the specified filter criteria are resumed. You must specify at least one filter, and the logged in administrator must have the necessary privileges on the matching jobs.

  If the property file is provided, criteria can be read from it as well as the command line. You can specify the execution targets and/or starting and ending time window in this file. All other properties in this file are ignored.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **preview**

  Only lists the jobs that would be suspended.

Examples

Example 1

This example suspends a job named MYJOB.

```
emcli suspend_job -name=MyJob
```

Example 2

This example suspends jobs or job executions matching search criteria in suspend_prop.txt. If the property file contains job details, matching jobs are suspended. If the property file contains time or target details, matching executions are suspended. If the property file contains job, time, and target details, matching executions of the matching jobs are suspended.

```
emcli suspend_job -input_file=property_file:/tmp/suspend_prop.txt
```

switch_cluster_agent

Switches the monitoring Management Agent of a particular cluster to the Management Agent of another host. The other host must be a member of the same cluster, and Cluster target must be available on that host.

Format

Standard Mode

```
emcli switch_cluster_agent
      -cluster="cluster"
      -host="host"
      [-debug]
```

```
[ ]  indicates that the parameter is optional.
```

Interactive or Script Mode

```
switch_cluster_agent(
      cluster="cluster"
      ,host="host"
      [,debug=True/False]
      )
```

```
[ ]  indicates that the parameter is optional.
```

Options

*   cluster

    Target name of the cluster whose monitoring Management Agent should be switched to the Management Agent on another host in the cluster.

*   host

    Name of the other host to which the monitoring Management Agent of the cluster should be switched. The other host must be a member of the same cluster, the Management Agent on that host must be up and running, and the cluster target on that host must be available.

*   debug

    Whether you want to turn on or turn off the debugging mode.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example switches the monitoring Management Agent of the cluster `cluster_target_name_1` to the Management Agent of the host `host_target_name_1`.

```
emcli switch_cluster_agent        -cluster=cluster_target_name_1
      -host=host_target_name_1
      -debug
```

ORACLE

switch_database

Switches the Oracle listener to a new Oracle home

Format

```
emcli switch_database
-input_file_loc=" path of input file"
-dest_oh=<path of new Gold image based Oracle home>
-host_creds=<Normal Named host credential>
-privilege_creds=<Named credential with root privileges>
```

[ ] indicates an optional parameter

If the target type, destination Oracle home, and credentials are consistent throughout multiple databases, users can perform the same task without a response file.

Options

- input_file_loc: Input file path location

- target_name: Database target name which needs to be switched.

- target_type:

  – For single instance database: oracle_database

  – For RAC/RAC One database: rac_database

- dest_oh: Path of new Oracle home where the database needs to be switched to.

- host_creds: Named credentials for the Oracle home owner on the host following "<Named Credentials>:<Owner>"

  The "Owner" field is the owner of the named credential in Enterprise Manager.

  If no "Owner" is specified, the logged-in user is taken by default.

- privilege_creds: Named credentials on the database host which has root privileges.

- dataguard_role: This is an explicit mandatory parameter which the user has to set to "STANDBY" in case this is a standby switch operation.

> **Note:**
>
> If database names are provided in the command line, input file will not be used.
>
> If input file is specified, for each database, the inputs in the response file override the one in the command line.
>
> In case the preferred credentials are set for the host, users can choose not to provide "host_creds" and "privilege_creds" as part of emcli command (or input file).
>
> We have a validation in initialize DP step for the same. If credentials are not provided as part of emcli and also the preferred credentials are not set, the DP will error out without any processing.
>
> In case this flag is not specified, it is considered to be a switch operation for databases without data guard configuration.

Example Input File

input_file: This file will be a "xml" format file and will contain details for each database target

```
<root>
<target>
<target_name>racm</target_name>
<target_type>rac_database</target_type>
<dest_oh>/scratch/aimedb/home1</dest_oh>
<host_creds>AIMEDB_NORMAL:SYSMAN</host_creds>
<privilege_creds>AIMEDB_P:SYSMAN</privilege_creds>
</target>
<target
><target_name>racp</target_name>
<target_type>rac_database</target_type>
<dest_oh>/scratch/aimedb/home2</dest_oh>
<host_creds>AIMEDB_NORMAL:SYSMAN</host_creds>
<privilege_creds>AIMEDB_P:SYSMAN</privilege_creds>
</target>
</root>
```

Example 5-2   Examples

This examples demonstrates how to use switch_database when there is more than one database.

```
emcli switch_database
-target_name=<DB target name>[,<DB target name2>]*
-target_type=<oracle_database|rac_database>
-dest_oh=<path of new Gold image based Oracle home>
-host_creds=<Normal Named host credential>
-privilege_creds=<Named credential with root privileges>
-dataguard_role=<Standby|Primary> -startupAfterSwitch=false
```

switch_gi

This command switches the Grid Infrastructure Oracle homes to the selected target.

Format

```
emcli switch_gi
-input_file_loc="path of input file"
-dest_oh=<path of new Gold image based Grid Infrastructure Oracle home>
-host_creds=<Normal Named host credential>
-privilege_creds=<Named credential with root privileges>
```

Options

- input_file_loc: Input file path location

- target_name: Database target name which needs to be switched.

- target_type:

  – For SIHA: "has"

  – For cluster: "cluster"

- dest_oh: Path of new Oracle home where the database needs to be switched to.

- host_creds: Named credentials for the Oracle home owner on the host following "<Named Credentials>:<Owner>"

  The "Owner" field is the owner of the named credential in Enterprise Manager.

  If no "Owner" is specified, the logged-in user is taken by default.

- privilege_creds: Named credentials on the database host which has root privileges.

> **Note:**
>
> If database names are provided in the command line, input file will not be used.
>
> If input file is specified, for each database, the inputs in the response file override the one in the command line.
>
> In case the preferred credentials are set for the host, users can choose not to provide "host_creds" and "privilege_creds" as part of EMCLI command (or input file).
>
> We have a validation in initialize DP step for the same. If credentials are not provided as part of emcli and also the preferred credentials are not set, the DP will error out without any processing.

Example 5-3   Examples

This example demonstrates how to switch multiple targets.

```
emcli switch_gi
-target_name=<Cluster target name>[,<Cluster target name2>]*
-target_type=<cluster|has>
-dest_oh=<path of new Gold image based Grid Infrastructure Oracle home>
-host_creds=<Normal Named host credential>
-privilege_creds=<Named credential with root privileges>
```

switch_swlib_oms_agent_storage

Modify a Software Library OMS Agent storage location to change the associated OMS Host and the credential for accessing the location.

Format

```
emcli switch_swlib_oms_agent_storage
      -name="location_name"
      -host="hostname"
      [-credential_set_name="setname"] | [-credential_name="name"
        -credential_owner="owner"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of an existing OMS Agent storage location.

- **host**

  Target name of the OMS host where the file system path for the storage location exists.

- **credential_set_name**

  Set name of the preferred credential stored in the repository for the host target. The value can be one of the following:

  – HostCredsNormal — Default unprivileged credential set

  – HostCredsPriv — Privileged credential set

- **credential_name**

  Name of a named credential stored in the repository. You must specify this option with the -credential_owner option.

- **credential_owner**

  Owner of a named credential stored in the repository. You must specify this option with the -credential_name option.

Examples

This example modifies the OMS Agent file system storage location named 'myOMSAgtLocation' to use the specified host 'fs1.us.acme.com', and the named credential 'MyAcmeCreds' owned by 'ACME_USER' for reading/writing files from/to this location.'

```
emcli switch_swlib_oms_agent_storage
       -name="myOMSAgtLocation"
       -host="fs1.us.acme.com"
       -credential_name="MyAcmeCreds"
       -credential_owner="ACME_USER"
```

switchover (deprecated)

Performs a Data Guard switchover between a primary database and a standby database.

> **Note:**
>
> Since switchover is deprecated, use dg_switchover instead.

Format

```
emcli switchover
      -primary_target_name="<primary target name>"
      -primary_target_type="oracle_database|rac_database"
      -standby_target_name="<standby target name>"
      -standby_target_type="oracle_database|rac_database"
      [-primary_db_creds_name="<primary database credential name>"]
      [-primary_host_creds_name="<primary database host credential name>"]
      [-standby_db_creds_name="<standby database credential name>"]
      [-standby_host_creds_name="<standby database host credential name>"]
      [-swap_jobs]
      [-swap_thresholds]
```

Options

- primary_target_name

Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- primary_target_type

  Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- standby_target_name

  Standby database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

- standby_target_type

  Standby database target type. Specify oracle_database for single instance, or rac_database for cluster.

- primary_db_creds_name

  Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- primary_host_creds_name

  Primary database host named credential for an operating system user who can access the primary database Oracle Home. Default is to use preferred credential.

- standby_db_creds_name

  Standby database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- standby_host_creds_name

  Standby database host named credential for an operating system user who can access the standby database Oracle Home. Default is to use preferred credential.

- swap_jobs

  Specifies that scheduled jobs on either the primary or standby database that are of a transferable type (Backup, RMAN Script, and SQL Script) will be moved to the other database after Enterprise Manager detects a fast-start failover. Default is not enabled.

- swap_thresholds

  Specifies that monitoring settings will be swapped between the primary and standby databases after Enterprise Manager detects a fast-start failover. Default is not enabled.

Examples

Example 1

The following command performs a switchover between a primary single-instance database named "database" and a standby single-instance database named "database1".

```
emcli switchover
     -primary_target_name="database"
     -primary_target_type="oracle_database"
     -standby_target_name="database1"
     -standby_target_type="oracle_database"
```

Example 2

The following command performs a switchover between a primary single-instance database named "primary" and a standby single-instance database named "standby". The scheduled

jobs and the monitoring settings on "primary" will be moved to "standby" after the switchover is complete.

```
emcli switchover
      -primary_target_name="primary"
      -primary_target_type="rac_database"
      -standby_target_name="standby"
      -standby_target_type="rac_database"
      -swap_jobs
      -swap_thresholds
```

sync

Synchronizes the EM CLI client with an OMS. After synchronization, all verbs and associated command-line help available to this OMS become available at the EM CLI client. Synchronization occurs automatically during a call to setup.

There are two ways to perform sync:

- With parameters

- Without parameters

  sync connects to the same OMS against which it has been set up and downloads the latest jar files.

Standard Mode

```
emcli sync
        [-url="http[s]://host:port/em"]
        [-username=<EM_username>]
        [-password=<EM_user's_password>]
        [-trustall]
        [-novalidate]
```

[ ] indicates that the parameter is optional.

Options

- **url**

  URL of the Enterprise Manager OMS. Both http and https are supported, but https is recommend for security purposes.

- **username**

  User name to be used by all subsequent EM CLI commands when contacting the OMS.

- **password**

  Enterprise Manager user's password. If you do not specify this option, you are interactively prompted for the password. Providing a password on the command line is insecure and should be avoided.

- **trustall**

  Automatically accepts any server certificate from the OMS, which results in lower security. Also indicates that the setup directory is local and trusted. Either pass this option or the set environment variable EMCLI_CERT_LOC, which has the certificate keystore file. If the file is not present, the system stores the certificate at this location.

- **novalidate**

  Does not authenticate the Enterprise Manager user name against the OMS. Assumes that the given username is valid.

Examples

This example synchronizes the EM CLI client with the OMS by connecting as Enterprise Manager user john_doe. The user is prompted for the password interactively.

```
emcli sync
          -url="https://mymachine.example.com"
          -username=john_doe
          -trustall
          -novalidate
```

sync_alerts

Synchronizes all alerts for the specified target between the Agent and the repository. You typically use this command when you think that the Agent has not uploaded the latest alert to the repository, and the repository is therefore out of sync with the Agent state.

To determine if alerts are out of sync between the Agent and the repository for the specified target, run the get_unsync_alerts command.

Format

```
emcli sync_alerts
        -target_type=type
        -target_name=name
        -agent_name=agent
```

Options

- **target_type**

  Internal target-type identifier (host, oracle_database, emrep, and so forth).

- **target_name**

  Name of the target.

- **agent_name**

  Name of the Agent.

Examples

Example 1

This example synchronizes alert states for target_type "host" and target_name "hostname.oracle.com".

```
emcli sync_alerts -target_type=host -target_name=hostname.oracle.com
```

Example 2

This example synchronizes alert states for all targets that the Agent "hostname.xyz.com:port" monitors.

```
emcli sync_alerts -agent_name=hostname.xyz.com:port
```

sync_beacon

Synchronizes a beacon that is monitoring the target (reloads all collections to the beacon).

Format

```
emcli sync_beacon
      -name=target name
      -type=target type
      -bcnName=beacon name
```

Options

- **name**

  Service target name.

- **type**

  Service target type.

- **bcnName**

  Beacon name to synchronize.

Examples

This example synchronizes `MyBeacon`, which is monitoring the `MyTarget` target of type `generic_service`.

```
emcli sync_beacon -name='MyTarget' -type='generic_service'
      -bcnName='MyBeacon'
```

synchronize_storage

Synchronizes a storage registered in Enterprise Manager.

Format

Standard Mode

```
emcli synchronize_storage
      -storage_name="<storage name>"
```

Interactive or Script Mode

```
synchronize_storage(
      storage_name="<storage name>"
)
```

```
[ ]  indicates that the parameter is optional.
```

Options

- storage_name

  Name of the storage.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example synchronizes the storage server:

```
emcli synchronize_storage
            -storage_name="sunzfs1"
```

test_named_credential

Tests the named credentials provided in the list. Instance credentials are tested against the credential target. Global credentials are tested against the target provided.

Format

```
emcli test_named_credential
      -cred_names=<cred_name_list>
      [-target_name=<target_name>]
      [-target_type=<target_type>]
```

Options

- **cred_names**

  List of credential names to be tested.

- **target_name**

  Target name to test the global credentials. Instance credentials are tested against their respective targets.

- **target_type**

  Target type to test the global credentials.

Examples

Example 1

This example tests the instance named credentials NC1 owned by the current logged in user and NC2 owned by ADMIN1.

```
emcli test_named_credential
      -cred_names="NC1;NC2:ADMIN1"
```

Example 2

This example tests the global host named credentials NC1, NC2, and NC3 against the target testhost.example.com.

```
emcli test_named_credential
      -cred_names="NC1;NC2;NC3"
      -target_name="testhost.example.com"
      -target_type="host
```

test_privilege_delegation_setting

Tests privilege delegation settings on a specified host.

Format

Standard Mode

```
emcli test_privilege_delegation_setting
        -host_name="Host Name"
        -cred_name="Cred Name"
        [-cred_owner="Cred Owner"]
```

Interactive or Script Mode

```
test_privilege_delegation_setting(
        host_name="Host Name"
        ,cred_name="Cred Name"
        [,cred_owner="Cred Owner"]
        )
```

```
[ ]  indicates that the parameter is optional
```

Options

* **host_name**

  Target name of the host.

* **cred_name**

  Credential name.

* **cred_owner**

  Credential owner

Exit Codes

0 on success. A non-zero value means verb processing was not successful.

Examples

Example 1

This example tests the privilege delegation settings for a host named *my_host* and credentials named *my_cred*.

```
emcli test_privilege_delegation_setting
            -host_name="my_host"
            -cred_name="my_cred"
```

Example 2

This example tests the privilege delegation settings for a host named "my_host" and credential named "my_cred" owned by "owner1."

```
emcli test_privilege_delegation_setting
            -host_name="host"
            -cred_name="cred"
            -cred_owner="owner1"
```

ORACLE®

test_proxy

Tests whether a HTTP(S) proxy identified by the specified name can be reached.

Syntax

```
emcli test_proxy
      -name="<name>"
       [-timeout=<minutes>]
```

`[ ]  indicates that the parameter is optional.`

Options

- -name

  Name identifying the proxy.

- -timeout

  Timeout, in minutes, for proxy test procedures. Timeout can be any value between 1 and 60 minutes. Default is 5 minutes.

Example

The following command tests whether a proxy identified by the name "us-proxy-1" is reachable. The test procedure times out after 2 minutes.

```
emcli test_proxy
     -name="us-proxy-1"
     -timeout=2
```

trace

Enables or disables tracing for OMS.

Format

```
emcli trace
        -enable="true|false"
        -user="username"
```

Options

- enable

  Specify true to enable and false to disable.

- user

  Name of the user.

Example

This example enables tracing for user sysman.

```
emcli trace -enable=true -user=sysman
```

trace_set_property

Sets the property name and corresponding value for the trace facility. These values are not persistent.

Format

```
emcli trace_set_property       -name=<property name>
      -value=<property value>
```

Options

- name

  Property name.

- value

  Property value.

Example

The following example enables tracing for the user.

```
emcli trace_set_property -name=trace.backgroundthreads -value=true
```

trigger_compliance_ca

Initiates the execution of a corrective action for a specified compliance event.

Format

```
emcli trigger_compliance_ca
      -event_instance_id
```

Options

- event_instance_id

  Identifier of the event instance. To get the event_instance_id, execute the following emcli command:

```
emcli  get_compliance_rule_violation_event
        -rule_iname="myrule"
        -target_type="host"
        -target_name="my_machine"
```

Example

The following example initiates the corrective action for the event with the ID 0123456789ABCDEF0123456789ABCDEF.

```
emcli trigger_compliance_ca
      -event_instance_id="0123456789ABCDEF0123456789ABCDEF"
```

udmmig_list_matches

Lists all the metric extensions that match the UDMs in a given migration session.

Format

```
emcli udmmig_list_matches
        -session_id=<sessionId>
```

Options

• **session_id**

Specify the ID that was returned when the session was created, or from the output of udmmig_summary.

udmmig_request_udmdelete

Deletes the UDMs that have been replaced by Metric Extensions.

Format

```
emcli udmmig_request_udmdelete
        -session_id=<sessionId>
        -input_file=metric_tasks:<complete_path_to_file>
```

Options

• **session_id**

Specify the ID that was returned when the session was created, or from the output of udmmig_summary.

• **input_file**

Specify a file name that contains a target, UDM, one per line in the following format:

```
<targetType>,<targetName>,<collection name>
```

For more information about the input_file parameter, see -input_file Syntax Guidelines.

Example

This example attempts to delete the UDM from all relevant targets. This step is indicative of the end of the migration process. The file input_tasks lists the locations where the UDM is present.

```
emcli udmmig_request_udmdelete -session_id=<sessionId> -
input_file=metric_tasks:input_tasks
```

udmmig_retry_deploys

Retries the deployment of metric extensions to a target.

Format

```
emcli udmmig_retry_deploys
        -session_id=<sessionId>
        -input_file=metric_tasks:<complete path to file>
```

Options

• **session_id**

Specify the ID that was returned when the session was created, or from the output of udmmig_summary.

- **input_file**

  Specify a file name that contains a target, UDM, one per line in the following format:

  ```
  <targetType>,<targetName>,<collection name>
  ```

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

Example

This example attempts to deploy the metric extension to all targets where the UDM was present. The file input_tasks lists these locations.

```
emcli udmmig_retry_deploys -session_id=<sessionId> -
input_file=metric_tasks:input_tasks
```

udmmig_session_details

Provides details of the specified migration session, including the targets, templates, UDMs, and metric extensions involved.

Format

```
emcli udmmig_session_details
       -session_id=<sessionId>
```

Options

- **session_id**

  Specify the ID that was returned when the session was created, or from the output of udmmig_summary.

udmmig_submit_metricpicks

Supply the metric picks to use to replace UDMs per target in a session.

Format

```
emcli udmmig_submit_metricpicks
       -session_id=<sessionId>
       -input_file=metric_picks:<complete_path_to_file>
```

Options

- **session_id**

  Specify the ID that was returned when the session was created, or from the output of udmmig_summary.

- **input_file**

  Specify a file name that contains a target, UDM, metric pick, one per line in the following format:

  ```
  <targetType>,<targetName>,<collection name>,[N/E],<metric>,<column>
  ```

Use N if a new metric should be created, or E if an existing metric is referenced.

For more information about the input_file parameter, see -input_file Syntax Guidelines.

Example

This example provides the mapping from UDM to the metric extension through the file input_picks.

```
emcli udmmig_submit_metricpicks -session_id=<sessionId> -
input_file=metric_picks:input_picks
```

udmmig_summary

Displays all the active migration sessions in the system.

Format

```
emcli udmmig_summary
      [-showAll]

[ ]  indicates that the parameter is optional
```

Options

- **showAll**

  Prints out all sessions including those that are complete. By default, only in-progress sessions are listed.

udmmig_update_incrules

Updates incident rules that reference UDMs with a reference to replacing a metric extension.

Format

```
emcli udmmig_update_incrules
      -session_id=<sessionId>
      -input_file=udm_inc_rules:<complete_path_to_file>
```

Options

- **session_id**

  Specify the ID that was returned when the session was created, or from the output of udmmig_summary.

- **input_file**

  Specify a file name that contains a rule, UDM, metric, one per line in the following format:

  ```
  <ruleset id>,<rule id>,<udm name>,<metric name>
  ```

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

unassign_charge_plan

Unassigns the charge plan associated with the specified entity.

Format

```
unassign_charge_plan
      -entity_name="eName"
      -entity_type="eType"
      -[entity_guid="entity_guid"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **entity_name**

  Name of the entity for which the charge plan is to be unassigned.

- **entity_type**

  Type of entity for which the charge plan is to be unassigned.

- **entity_guid**

  guid of the entity to be added to Chargeback.

  When more than one entity is active in Chargeback with the given entity name and entity type, the command lists all such entities with additional details such as creation date, parent entity name, entity guid, and so forth to choose the correct entity. Select the correct entity from the given list and execute the command again with entity guid as the parameter instead of entity name and entity type.

Example

This example unassigns charge plan associated to "db1", an oracle_database entity.

```
emcli unassign_charge_plan -entity_name="db1" -entity_type="oracle_database"
```

See Also

add_chargeback_entity
assign_charge_plan
assign_cost_center
list_chargeback_entities
list_chargeback_entity_types
list_charge_plans
list_cost_centers
remove_chargeback_entity
unassign_cost_center

unassign_cost_center

Unassigns the cost center from the given entity.

Format

```
unassign_cost_center
      -entity_name="eName"
      -entity_type="eType"
      -[entity_guid="entity guid" ]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **entity_name**

  Name of the entity for which the cost center is to be unassigned.

- **entity_type**

  Type of entity for which the cost center is to be unassigned.

- **entity_guid**

  guid of the entity in Chargeback.

  When more than one entity is active in Chargeback with the given entity name and entity type, the command lists all such entities with additional details such as creation date, parent entity name, entity guid, and so forth to choose the correct entity. Select the correct entity from the given list and execute the command again with entity guid as the parameter instead of entity name and entity type.

Example

This example unassigns the cost center associated to "db1", an Oracle database entity.

```
emcli unassign_cost_center -entity_name="db1" -entity_type="oracle_database"
```

See Also

add_chargeback_entity
assign_charge_plan
assign_cost_center
list_chargeback_entities
list_chargeback_entity_types
list_charge_plans
list_cost_centers
remove_chargeback_entity
unassign_charge_plan

undeploy_diagchecks

Undeploys diagcheck scripts for targets.

Format

```
emcli undeploy_diagchecks
      {-target_name=<target_name_to_be_updated>
       -target_type=<target_type_to_be_updated> }
      | {-input_file=targetList:<complete_path_to_file>};
```

Options

- **target_name**

  Name of the target to be updated.

- **target_type**

  Type of target to be updated.

- **input_file**

Specify a file name that contains a list of targets, one per line in the following format:

```
<targetType>:<targetName>
```

For more information about the input_file parameter, see -input_file Syntax Guidelines.

undeploy_plugin_from_agent

Undeploys an Enterprise Manager plug-in from the Management Agents. Undeploying a plug-in from a Management Agent removes all targets of any type belonging to this plug-in from Enterprise Manager.

Defaults to the version currently deployed on the given Management Agent.

Format

```
emcli undeploy_plugin_from_agent
        -plugin="pluginId[:pluginVersion]"
        -agent_names="agent1;agent2"
```

Options

* **plugin**

  Plug-in ID and version to be undeployed. Version is optional, and it defaults to the latest version deployed on the management server.

* **agent_names**

  Management Agents (host:port) from which the plug-in is to be undeployed.

Examples

Example 1

This example undeploys the oracle.sysman.db2 plug-in of version 11.2.0.1.0 from Management Agents myhost1.example.com:1159 and myhost2.example.com:1159.

```
undeploy_plugin_from_agent -plugin=oracle.sysman.db2:11.2.0.1.0
-agent_names="myhost1.example.com:1159;myhost2.example.com:1159"
```

Example 2

This example undeploys the oracle.sysman.db2 plug-in of the latest version from the Agent myhost1.example.com:1159.

```
undeploy_plugin_from_agent -plugin=oracle.sysman.db2
-agent_names="myhost1.example.com:1159"
```

undeploy_plugin_from_server

Undeploys a plug-in from the Oracle Management Server.

> **✎ Note:**
>
> You need to undeploy the plug-in from all Management Agents before you can undeploy it from the management server.

Format

```
emcli undeploy_plugin_from_server
        -plugin="plug-inId"[:"pluginVersion"]
        [-sys_password="sys_password"]

[ ]  indicates that the parameter is optional
```

Options

- **plugin**

  This is of the form -plugin=<oracle.sysman.db:12.1.0.1.0> where the plug-in id (like oracle.sysman.db) is a required parameter and the version is optional.

  You do not need to provide a version in the -plugin="plugin_id" field, because at any given time, only one version of the plug-in can be deployed on the management server. Therefore, the version is implicit. Contrast this with providing a version during deployment, because you could have downloaded more than one version.

- **sys_password**

  The repository sys user password. If not provided at the console, it will be prompted for.

Examples

Example 1

This example undeploys the "oracle.sysman.db2" plug-in from the Oracle Management Server.

```
undeploy_plugin_from_server -plugin="oracle.sysman.db2" -sys_password=welcome
```

Example 2

This example prompts you for sys_password.

```
emcli undeploy_plugin_from_server -plugin="oracle.sysman.db2"
```

unregister_bipublisher

Unregisters a previously set up relationship between Enterprise Manager and a previously set up relationship (using setup_bipublisher). You can also use this verb to determine the status of the relationship between Enterprise Manager and BI Publisher if you do not specify the -force option.

> **✎ Note:**
>
> This verb requires Enterprise Manager Super Administrator privileges.

Format

```
emcli unregister_bipublisher
    [-force]
```

Options

*   **force**

    This option severs the relationship. The BI Publisher managed server is not stopped or uninstalled.

Examples

Example 1

```
emcli unregister_bipublisher

Error: The BI Publisher Web Application named "https://
somehost.somedomain.com:9704/xmlpserver" is registered. Use -force option to
overwrite this.
```

Example 2

```
emcli unregister_bipublisher -force

BI Publisher "https://somehost.somedomain.com:9704/xmlpserver" has been
unregistered for use with Enterprise Manager.
```

unsecure_agent

Unsecures a secured Management Agent. This verb requires operator privilege or full privilege on the Management Agent.

Format

```
emcli unsecure_agent
      -agent_name="agent_target_name"
      [-host_username ="agent_host_username" -host_pwd="agent_host_password"]
      [-credential_name ="credential_name"]
      [-credential_setname ="credential_setname_of_agent"]

[ ]  indicates that the parameter is optional
```

Options

*   **agent_name**

    Name of the Management Agent target.

*   **host_username**

    User name of the OS user (on the host) who owns the Management Agent.

*   **host_pwd**

    Password of the OS user (on the host) who owns the Management Agent.

*   **credential_name**

    Name of the saved credential.

- **credential_setname**

  Name of the credential set of the Management Agent. Example: "HostCreds"

Examples

Example 1

```
emcli unsecure_agent -agent_name="agent.example.com:1234"
                          -host_username="test_user"
                          -host_pwd="test"
```

Example 2

```
emcli unsecure_agent -agent_name="agent.example.com:1234"
                          -credential_name="MyMachineCredential"
```

unsubscribe_agents

Unsubscribes the specified Management Agents that subscribe to a particular Management Agent gold image.

Format

```
emcli unsubscribe_agents
      -image_name="Image Name"
      [-agents="Full Agent Name"]
      [-groups="List of group names"]
      [-closure_related="true/false"]
      [-closure_nfs="true/false"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- image_name

  Image name from which the Management Agents should unsubscribe.

- agents

  Name pattern of the Management Agents that should unsubscribe from the specified Management Agent gold image.

- groups

  Management Agent groups that should unsubscribe from the specified Management Agent gold image.

- closure_related

  Whether or not the related Management Agents should be unsubscribed. When set to `false`, the related Management Agents are not unsubscribed.

- closure_shared

  Whether or not the related Management Agents should be unsubscribed. When set to `false`, the related shared Management Agents are not unsubscribed.

Examples

Example 1

The following example unsubscribes `xyz.domain.com:1243` and all its related shared agents that subscribe to the Management Agent gold image `OPC_AGT_ADC_POD`.

```
emcli unsubscribe_agents
      -image_name="OPC_AGT_ADC_POD"
      -agents="xyz.domain.com:1243"
      -closure_shared="true"
```

Example 2

The following example unsubscribes `xyz.domain.com:1243` and all its related Management Agents that subscribe to the Management Agent gold image `OPC_AGT_ADC_POD`.

```
emcli unsubscribe_agents
      -image_name="OPC_AGT_ADC_POD"
      -agents="xyz.domain.com:1243"
      -closure_related="true"
```

unsuppress_incident

Unsuppresses an incident with the specified ID.

Format

```
emcli unsuppress_incident
      -incident_id="Id of the incident to be unsuppressed"
```

Options

* incident_id

  Numeric ID of the incident to be unsuppressed.

Example

The following example unsuppresses and incident with the ID 173.

```
emcli unsuppress_incident -incident_id="173"
```

unsuppress_problem

Unsuppresses a specified problem.

Format

```
emcli unsuppress_problem
      -problem_id="id of the problem to be unsuppressed"
```

Options

* problem_id

  ID of the problem to be unsuppressed.

Example

The following example unsuppresses a problem with ID 173.

```
emcli unsuppress_problem -problem_id="173"
```

update_and_retry_step

Updates arguments of the failed step and retries it.

Format

```
emcli update_and_retry_step
    -stateguid=<state_guid>
    [-instance=<instance_guid>]
    [-exec=<execution_guid>]
    [-name=<execution_name>]
    [-owner=<execution_owner>]
    [-args="command1:value1;command2:value2;..."]

[ ]  indicates that the parameter is optional
```

Options

- **stateguid**

  State GUID.

- **instance**

  GUID of the instance.

- **exec**

  GUID of the execution.

- **name**

  Name of the execution.

- **owner**

  Owner of the execution.

- **args**

  Arguments of the step to be updated during retry. The format of the arguments are name-value pairs. Name and value are separated by a colon (:), and each pair is separated by a semicolon (;). The arguments take scalar data and list data. The format of list data should be like [a,b,c].

  For the full list of arguments that can be updated, see the get_retry_arguments verb.

Examples

```
emcli update_and_retry_step -instance=16B15CB29C3F9E6CE040578C96093F61 -
stateguid=51F762417C4943DEE040578C4E087168 -args="command:ls"
```

update_agents

Prepares the environment for updating your Management Agents and submits the Management Agent update job.

Format

```
emcli update_agents
-gold_image_name | -gold_image_series
-agents="agent_names" | -input_file="agents_file:input_file_location"
```

```
[-validate_only]
[-pre_script_loc="location_of_pre_script"]
[-pre_script_on_oms]
[-post_script_loc="location_of_post_script"]
[-post_script_on_oms]
[-op_name="custom_operation_name"]
[-override_credential="named_credential"]
[-additional_parameters]
[-stage_location="custom_stage_location"]
[-is_staged="true|false"]
[-stage_action="push|pull"]

[ ]  indicates that the parameter is optional
```

Options

- gold_image_series

  Specify this option to update the selected Management Agents to the latest
  Management Agent gold image of the specified series.

- gold_image_name

  Specify this option to update the selected Management Agents to the specified
  Management Agent gold image.

- agents

  Specify this option to enter the names of all the Management Agents that you want
  to update.

- input_file

  Specify this option if the list of all the Management Agents that you want to update
  is stored in an input file.

- validate_only

  Specify this option if you only want to check whether or not the Management
  Agents that you have selected for update can be updated, and not update these
  Management Agents immediately. If you use this option, the Management Agent
  update job will not be submitted.

- pre_script_loc

  Specify this option if you want to execute a script before updating the selected
  Management Agents.

- pre_script_on_oms

  Specify this option if the post-script is present on the OMS host.

- op_name

  Specify this option to use a custom operation name for the Management Agent
  update.

- override_credential

  The preferred credentials of the Management Agent Oracle home are used to run
  root.sh (on certain Management Agents) after the update. Specify this option if you
  want to override these credentials with different named credentials

- additional_parameters

Specify this option if you want to pass additional parameters for the Management Agent update.

- stage_location

  Specify this option if you want to use a custom stage location for the Management Agent update.

- is_staged

  Specify this option as 'true' if you have already staged the Management Agent gold image.

- stage_action

  If the Management Agent gold image has not already been staged, by default, the gold image is pushed to the Management Agents that you have selected for update. Specify this option as 'pull' if you want the Management Agents selected for update to pull the Management Agent gold image instead.

It is mandatory to specify the -gold_image_name parameter or the -gold_image_series parameter. Also, it is mandatory to specify the -agents parameter or the -input_file parameter. If you specify -agents as well as -input_file, a union of the outputs (when each of these parameters is specified individually) is displayed.

All parameters can be passed in a response file, using the -input_file parameter. For example, -input_file="response_file:/scratch/response_file.txt".

In the response file, each parameter must be specified on a new line, and in name value pairs. For example, op_name=UPDATE_AGT_121020

If the same parameter is passed both on the command line as well as in the response file, the value of the command line parameter is given precedence

Examples

Example 1

The following example updates xyz.domain.com:1243 (creates an update job 'UPDATE_JOB123') using the Management Agent gold image 'OPC_AGT_ADC_POD_JUNE':.

```
mcli update_agents
    -gold_image_name="OPC_AGT_ADC_POD_JUNE"
    -agents="xyz.domain.com:1243"
    -op_name="UPDATE_JOB123"
```

Example 2

The following example updates xyz.domain.com:1243 using the latest Management Agent gold image in the series 'OPC_AGT_ADC_POD', passing two additional parameters:

```
emcli update_agents
    -gold_image_series="OPC_AGT_ADC_POD"
    -agents="xyz.domain.com:1243"
    -additional_parameters="-ignorePrereqs
    -newParameter"
```

update_audit_settings

Updates the current audit settings in the repository and restarts the OMS. Only
Enterprise Manager Super Administrators have permission to view the audited data.

Format

```
emcli update_audit_settings
      [-audit_switch="ENABLE|DISABLE"]
      [-operations_to_enable="name_of_operations_to_enable"]
      [-operations_to_disable="name_of_operations_to_disable"]
      [-externalization_switch="ENABLE|DISABLE"]
      [-directory="directory_name"]
      [-file_prefix="file_prefix"]
      [-file_size="file_size"]
      [-data_retention_period="data_retention_period"]

[ ]  indicates that the parameter is optional
```

Options

*   **audit_switch**

    Audit switch to enable auditing across Enterprise Manager.

*   **operations_to_enable**

    Enables auditing for specified operations. To enable all operations, specify ALL.
    this option is invalid if auditing is disabled.

*   **operations_to_disable**

    Disables auditing for specified operations. To disable all operations, specify ALL.
    this option is invalid if auditing is disabled.

*   **externalization_switch**

    Enable the audit data export service. The default value is DISABLE.

*   **directory**

    Database directory that is configured with an OS directory where the export
    service archives the audit data files. This directory is required to externalize audit
    data in Enterprise Manager Cloud Control. The update_audit_settings verb
    assumes that this directory has already been created. The following example
    creates the database directory EMDIR from the directory `/tmp/em_audit_data`
    with read/write permissions for the SYSMAN user:

    ```
    SQL>create directory EMDIR as '/tmp/em_audit_data';
    Directory created.
    SQL>grant read,write on directory "EMDIR" to SYSMAN;
    Grant succeed.
    ```

*   **file_prefix**

    File prefix to be used by the export service to create the file name where audit
    data is to be written. The default value is em_audit.

*   **file_size**

    Maximum value of each file size. The default value for this is 5000000 bytes.

*   **data_retention_period**

Maximum period the Enterprise Manager repository stores audit data. The default value is 365 days.

Examples

Example 1

This example enables all operations except LOGIN and LOGOUT:

```
emcli update_audit_settings
        -audit_switch="ENABLE"
        -operations_to_enable="ALL"
        -operations_to_disable="LOGIN;LOGOUT"
```

Example 2

This example enables Weblogic Server specific operations to be audited.

```
emcli update_audit_settings
    -operations_to_enable="WEBLOGIC_DOMAIN_UPDATE_INVOKE;WEBLOGIC_DOMAIN
    _LOGIN;WEB_LOGIC_DOMAIN_LOGOUT"
```

update_credential_set

Update privileges required to get/set global preferred credentials. You can update privileges for a single credential set, for all credential sets of a specific target type, or for the entire system (all target types).

Format

Standard Mode

```
emcli update_credential_set
        [-set_name="set_name"]
        [-target_type="ttype"]
        [-get_priv="get_priv"]
        [-update_priv="update_priv"]
        [-update_default_priv="update_default_priv"]
```

Interactive or Script Mode

```
update_credential_set(
        [set_name="set_name"]
        [,target_type="ttype"]
        [,get_priv="get_priv"]
        [,update_priv="update_priv"]
        [,update_default_priv="update_default_priv"]
        )
```

```
[ ]  indicates that the parameter is optional
```

Options

*   **set_name**

    Credential set name for which privileges need to be updated.

*   **target_type**

    Target type for the target/credential set.

*   **get_priv**

Name of the privilege required to *get* the system scoped credential set.

- **update_priv**

  Name of the privilege required to *set/clear* the system scoped credential set.

- **update_default_priv**

  Name of the privilege required to *set/clear* the default global scoped preferred credentials for the set.

Exit Codes

0 on success. A non-zero value indicates that verb processing was not successful.

Examples

This example changes the privilege *get_priv* to *VIEW_TARGET* across all target types and all credential set names.

```
emcli update_credential_set -get_priv=VIEW_TARGET
```

update_database_size

Lists all of the database sizes that have been created.

Format

```
emcli update_database_size
     -name="<Existing size name>"
     -description="<Size description>"
     [-attributes="cpu:<number of cores>;memory:<memory in GB>;processes:
        <max number of processes>;storage:<Total storage in GB allocated
        to database>;"]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- name

  The name of the existing database size.

- description

  Updates the description of the database size.

- attributes

  Defines the database size. Attributes must be separated by a semicolon(;). You can specify values for the following attributes:

  cpu: Total number of cpu cores.

  memory: Total maximum in GB.

  processes: Total number of processes that can simultaneously connect to the database.

  storage: Total storage that is allocated to the database (in GB)

Example

The following command updates the description and attributes of the database size with the
name Small.

```
emcli update_database_size
     -name=Small
     -description="Small size database"
     -attributes="cpu:4;storage:50;memory:4;processes:remove"
```

update_db_account_status

Updates the database account status to any of LOCKED,OPEN,EXPIRED and LOCKED &
EXPIRED.

Format

```
emcli update_db_account_status        -target_name="tname"        -
user_name="user_name"         -connect_as=
"DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]" or
"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"         -action=action        [-
target_type="ttype"]         [-input_file="tag1:file_path1;tag2:file_path2;..."]
[-validate_only="yes/no" default "no"]

[ ]  indicates that the parameter is optional.
```

Options

- target_name

  Name of the target.

- user_name

  Name of the database user. The user_name is case sensitive.

- action

  Action to be performed. Possible values for this option are as follows:

  – LOCK - locks the account.

  – UNLOCK- unlocks the account.

  – EXPIRE - expires the account.

  – LOCK_AND_EXPIRE - lock as well as expires the account.

  – UNLOCK_AND_EXPIRE - unlock as well as expires the account.

- target_type

  Type of target. The possible values for target type in this verb are:

  – oracle_database

  – rac_database

  The default value for this argument is oracle_database.

- connect_as

  Connect to the target database as. It should be specified in one of the two following
  formats:

- – DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]

- – [DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds

- input_file

  Use this option to hide passwords. Specify the path to the file containing the old and new passwords. Each path must be accompanied by a tag that is referenced in the password options.

- validate_only

  Indicate whether to validate the options mentioned without doing the actions.

  - – Yes - Validate the options mentioned. Do not do any actions. Through any validation errors on to the console.

  - – No - do the actions as per the mentioned options.

  The default value of this option is NO.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

The following are three examples of the update_db_account_status_verb verb.

Example 1

The following example locks account Admin1 on the target database myDB using the named credential SYS_myDB.

```
emcli update_db_account_status        -target_name=myDB        -
user_name=Admin1        -action=LOCK        -
connect_as="DBNamedCreds:SYS_myDB"
```

Example 2

The following example unlocks account Admin1 on target database myDB using SYSDBA.

```
emcli update_db_account_status        -target_name=myDB        -
user_name=Admin1        -action=UNLOCK        -
connect_as="DBUserName:sys;DBPassword:welcome;DBRole:sysdba"
```

update_db_password

Updates the target database password change in the Enterprise Manager Credential sub-system and can change the password on the target database as well. This verb also propagates the collection or monitoring credentials to Enterprise Manager Management Agents.

Standard Mode

```
emcli update_db_password
       -target_name="tname"
       -user_name="user_name"
       [-target_type="ttype"]
       [-change_all_references="yes/no"]
       [-change_at_target="yes/no"]
       [-role="DBRole"]
```

```
[-unlock="yes/no"]
[-connect_as="DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]"]
[-copy_password_file="yes/no"]
[-host_creds="(HostName:hname;HostUserName:huname;HostPassword:hpassword)"]
[-input_file="tag1:file_path1;tag2:file_path2;..."]
[-validate_only="yes/no"]
```

```
[ ]  indicates that the parameter is optional
```

Interactive or Script Mode

```
update_db_password
        (target_name="tname"
        ,user_name="user_name"
        [,target_type="ttype"]
        [,change_all_references="yes/no"]
        [,-change_at_target="yes/no"]
        [,-role="DBRole"]
        [,-unlock="yes/no"]
        [,-connect_as="DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]"]
        [,-copy_password_file="yes/no"]
        [,-host_creds="(HostName:hname;HostUserName:huname;HostPassword:hpassword)"]
        [,-input_file="tag1:file_path1;tag2:file_path2;..."]
        [,-validate_only="yes/no"])
```

```
[ ]  indicates that the parameter is optional
```

Options

- **target_name**
  Name of the target.

- **user_name**
  Name of the database user.

- **target_type**
  Type of target. The possible values for target type in this verb are `oracle_database` and `rac_database`. The default value for this option is oracle_database. For the rac_database type, the password should be changed at the database and not at the individual instance level.

- **change_all_references**
  Specify if the password must be changed for all references in Enterprise Manager. Possible values are:

  – **yes** - Update all password references in Enterprise Manager for a database monitoring user who has an old password that matches the new password.

  – **no** - Update the password for the currently logged in user.

  The default value of this option is **yes**.

- **change_at_target**
  Specify whether the password must also be changed on the target.

  – **yes** — Change the password on the target database.

  – **no** — Update the password only on Enterprise Manager.

  The default value of this option is **no.**

- **role**
  Database user role

- **unlock**
  Specifies if the user should be unlocked. Possible values are:

  – **yes** - unlock the user.

  – **no** - do not attempt to unlock the user.

  The default value of this option is **no.**

- **connect_as**
  Connect to the target database as. This option enables the user to change the password of a different user without having to login as that user. It should be specified in any of following formats :

  – -- `"DBUserName:uname;DBPassword:upassword[;DBRole:SYSDBA]"`

  – -- `"[DBCredsOwner:ucredsowner;]DBNamedCreds:unamedcreds"`

  > **Note:**
  >
  > If this is not specified, the password for the `user_name` will be changed.

- **copy_password_file**
  Indicates whether to copy the password file across instances in RAC or across secondary databases in Data Guard environments in case it is needed. `host_creds` must be specified when this option is **yes.** Possible values are:

  – **yes** - copy the password file.

  – **no** - do not attempt to copy the password file.

  The default value of this option is **no**.

- **host_creds**
  This specifies the named host credentials information that would be used while copying the password file. Named credentials for multiple hosts can be specified in the following format:

  `"(HostName:hname;HostUserName:huname;HostPassword:hpassword"` or
  `"HostNamedCreds=hnamedcreds")+`
  For example, if named credentials for multiple hosts are needed to copy a password file to one or more standby databases that are on hosts that are different from the primary database, this can be specified in the following format for all the hosts where the password file needs to be updated.

  –
  `host_creds="HostName:host1.example.com;HostNamedCreds:HostNamedCred1`
  `;HostName:host1Standby;HostNamedCreds:StandbyHostNamedCred1"`

- **input_file**
  Path of the file that has old and new passwords. Use this option to hide passwords displayed on the command line. You must accompany each path with a tag referenced in the password options. This can be specified in the following format:

  `tag1:file_path1;tag2:file_path2;...`
  When you execute this verb with the input_file option, you are prompted to enter the following values in non-echo mode:

  `-old_password -new_password -retype_new_password`

For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **validate_only**
  Indicates whether to only validate the options mentioned without doing the actions.

  – **yes** - You must validate the options that are mentioned. Do not perform any action and display any validation errors on to the console.

  – **no** - Perform the actions as per the meentioned options.

  The default value of this option is **no**.

> **Note:**
>
> On execution, you will be prompted to enter the following values in non-echo mode:
>
> ```
>  -old_password
> -new_password
> -retype_new_password
> ```
>
> This option hides the password to be displayed on command line.

Examples

Example 1 - Command-Line

```
emcli update_db_password
      -target_name=myDB
      -user_name=Admin1
```

Example 2 - Scripting and Interactive

```
update_db_password
        (target_name="myDB",
         user_name="Admin1")
```

Example 3 - Command-Line

```
emcli update_db_password
      -target_name=myDB
      -user_name=Admin1
      -change_at_target=yes
```

Example 4 - Scripting and Interactive

```
update_db_password
      (target_name="myDB",
       user_name="Admin1",
       change_at_target="yes")
```

Example 5 - Copy Password File

```
emcli update_db_password
      -target_name="myDB"
      -target_type="oracle_database"
      -user_name="Admin1"
      -role="Primary"
      -change_at_target=yes
      -change_all_references=yes
```

ORACLE®

```
        -copy_password_file="yes"
        -
host_creds="HostName:host1.example.com;HostNamedCreds:HostNamedCred1;HostName:hos
t1Standby;HostNamedCreds:StandbyHostNamedCred1"
```

update_dbaas_quota

Updates the database quota for an SSA user role.

Format

```
emcli update_dbaas_quota
        -role_name="<SSA_User_Role_name>"
        -databases="<Number_of_Database_Requests>"
        -schema_services="<Number_of_Schema_Service_Requests>"
        -pluggable_databases="<Number_of_Pluggable_Database_Service_Requests>"
        -memory="<Memory(GB)>"
        -storage="<Storage(GB)>"
```

Options

*   **role_name**

    Name of an SSA user role for which the quota is to be updated.

*   **databases**

    Number of database service requests allowed.

*   **schema_services**

    Number of schema service requests allowed.

*   **pluggable_databases**

    Number of pluggable database service requests allowed.

*   **memory**

    Amount of memory (GB) usage allowed.

*   **storage**

    Amount of storage (GB) usage allowed.

Examples

```
emcli update_dbaas_quota-role_name="My Role"-databases="10"-schema_services="10"-
pluggable_databases="10"-memory="99"-storage="99"
```

displays the following output:

```
Quota for "My Role" updated successfully.
```

update_dbaas_request_settings

Updates the database request settings.

Format

```
emcli update_dbaas_request_settings
        -future_reservation_length="<Future_Request_Scheduling_Period>"
```

```
                    -maximum_archive_duration="<Request_Purging_Duration>"
                    -default_retirement_period="<Default_Retention_Duration>"
```

Options

- **future_reservation_length**

  Amount of time in advance a self-service user can schedule a request. Example: "2 Months" for 2 Months, "10 Weeks" for 10 Weeks, and "No Reservation" for no restriction

- **maximum_archive_duration**

  Amount of time after which the "Completed" Self Service Create Requests will be purged from the Repository. Example: "2 Months" for 2 Months, "10 Weeks" for 10 Weeks, and "No Reservation" for no restriction

- **default_retirement_period**

  The maximum amount of time for which a self-service user can retain a service instance. Example: "2 Months" for 2 Months, "10 Weeks" for 10 Weeks, "No Reservation" for no restriction

Examples

```
emcli update_dbaas_request_settings-future_reservation_length="2 Months"-
maximum_archive_duration="10 Weeks"-default_retirement_period="No Reservation"
```

displays the following output:

```
Request settings updated successfully.
```

update_diagchecks

Updates diagnostic check scripts for targets.

Format

```
emcli update_diagchecks
     -target_name=<target_name_to_be_updated>
     -target_type=<target_type_to_be_updated>
     [-input_file=targetList:<complete_path_to_file>]
```

Options

- **target_name**

  Name of the target to be updated.

- **target_type**

  Type of the target to be updated.

- **input_file**

  Specify a file name that contains a list of targets, one per line in the following format:

  ```
  <targetType>:<targetName>
  ```

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

update_host_password

Updates the changed host password in the credential sub-system. For collection or monitoring credentials, the password change is optionalso propagated to the Enterprise Manager Management Agent.

Format

```
emcli update_host_password      -target_name="tname"      -
user_name="user_name"      [-change_all_references="yes/no"]      [-
input_file="tag1:file_path1;tag2:file_path2;..."]

[ ]  indicates that the parameter is optional
```

> **✏ Note:**
>
> When you execute this verb, you are prompted to enter the following values in non-echo mode:
>
> -old_password -new_password -retype_new_password

Options

- **target_name**

  Name of the target.

- **user_name**

  Name of the database user.

- **change_all_references**

  Specifies if the password must be changed for all references in Enterprise Manager for the given user.

  Possible values are:

  – Yes — Updates all references in Enterprise Manager for this password.

  – No — Updates the password for the current logged-in user. This is the default.

- **input_file**

  File path that has old and new passwords. This hides passwords. You must accompany each path with a tag referenced in the password.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

Examples

Example 1

This example asks the user to enter the values of the old and new passwords, then retype the new password to update the new password in Enterprise Manager for this target reference.

```
emcli update_host_password
      -target_name=myHost
      -user_name=Admin1
```

Example 2

This example asks the user to enter the values of the old and new passwords, then retype the new password to update the new password in Enterprise Manager for all users' credentials referenced with the myHost target name and Admin1 user name.

```
emcli update_host_password
      -target_name=myHost
      -user_name=Admin1
      -change_all_references=yes
```

update_mda_properties

Updates the specified property names with the provided values. Updates are performed only if the specified property name exists in the repository. If the property does not exist in the repository, the property will not be inserted.

Format

```
emcli update_mda_properties  [-props="<list of property names and values>"]   [-
separator=props="separator_for_properties"]   [-
subseparator='props="separator_between_name_value_pair_for_properties"]
```

```
[ ] indicates that the parameter is optional.
```

Options

- props

  Provides a delimited list of property names and values. The default delimiter is ';'.

- separator

  Provides the separator used for individual property and value pairs.

- subseparator

  Provides the sub-separator used between the property and it's value.

Examples

Example 1

The following command updates multiple properties:

```
emcli update_mda_properties
   -props="prop1:val1;prop2:val2"
```

Example 2

The following command updates multiple properties with a custom separator and sub-separator:

```
emcli update_mda_properties
  -props="prop1=val1:prop2=val2"
  -separator='props=:'
  -subseparator='props=='
```

update_monitoring_creds_from_agent

Finds all of the targets on the Management Agent, retrieves the monitoring credentials, and updates them in the Management Repository. In 11g Release 1 (11.1.0.0), the monitoring credentials for some targets were stored only on the Management Agent.

> **Note:**
>
> Although -emd_list and -update_all are shown syntactically as optional, you must provide either one or the other.

Format

```
emcli update_monitoring_creds_from_agent
      [-emd_list=<emd_list>]
      [-update_all]

[ ]  indicates that the parameter is optional
```

Options

*   **emd_list**

    List of EMD URLs. You must provide either this option or the update_all option.

*   **update_all**

    Update in the repository for all targets that have monitoring credentials on the Agents but not in the repository. You must provide either this option or the emd_list option.

Exit Codes

0 if successful. A non-zero value means that verb processing was not successful.

Examples

Example 1

This example finds all the targets monitored by host1.example.com:1832 and host2.example.com:1832 that have monitoring credentials on the Agent but not in the management repository, and updates the monitoring credentials in the management repository.

```
emcli update_monitoring_creds_from_agent
      -emd_list="host1.example.com:1832;host2.example.com:1832"
```

Example 2

This example finds all the targets that have monitoring credentials on the Management Agents but not in the management repository, and updates the monitoring credentials in the repository.

```
emcli update_monitoring_creds_from_agent
      -update_all
```

update_operation_plan

Updates the SiteGuard operation plan.

Format

```
emcli update_operation_plan
        [-name=<plan_name>]
        [-step_number=<step_number>]
        [-target_host=<host_name>]
        [-error_mode=<error_mode>]
        [-enabled=<true|false>]
        [-execution_mode=<Serial|Parallel>]
        [-move=<Up|Down>]

[ ]  indicates that the parameter is optional
```

Options

*   **name**

    Name of the operation plan.

*   **step_number**

    Number of the step that should be updated.

*   **target_host**

    Target host name. Specifying this updates all of the steps involving this target host.

See Also

create_operation_plan
get_operation_plan_details

Examples

```
emcli update_operation_plan -name="austin-switchover"
            -step_number="1"
            -error_mode="Continue"
            -enabled="true"
            -execution_mode="Serial"

emcli update_operation_plan -name="austin-switchover"
            -step_number="5"
            -move="Up"

emcli update_operation_plan -name="austin-switchover"
            -target_host="myhost.domain.com"
            -error_mode="Continue"
            -enabled="true"
```

update_organizational_entity

Updates the organizational entity by changing the entity name. You can also change the entity type and create associated cost centers.

Standard Mode

```
emcli update_organizational_entity
      -entity_name="entity name"
      [-entity_type="entity type"]
      [-new_entity_name="new entity name"]
      [-parent_entity_name="parent entity name"]
      [-tenant_name="tenant name"]
      [-cost_centers="cost centers"[;"cost centers"...]]
      [-separator=argument_name="separator_value"])
```

`[ ]  indicates that the parameter is optional.`

Interactive or Script Mode

```
update_organizational_entity
      (,entity_name="entity name"
      [,entity_type="entity type"]
      [,new_entity_name="new entity name"]
      [,parent_entity_name="parent entity name"]
      [,tenant_name="tenant name"]
      [,cost_centers="cost centers"[;"cost centers"...]]
      [,separator=argument_name="separator_value"])
```

`[ ]  indicates that the parameter is optional.`

Options

- entity_name

  Name of the organizational entity to be updated.

- entity_type

  New entity type if changing from department to lob or vice versa.

- new_entity_name

  New name of the original entity.

- parent_entity_name

  Specifies a parent of the organizational entity being updated. The parent must already exist and can be either a department or LOB, regardless of the type being updated. Default is no parent.

- tenant_name

  Specifies the name of the tenant to which the organizational entity being updated belongs. Default is the tenant of the logged-in user.

- cost_centers

  Specifies one or more cost centers to create and associate with the organizational entity being updated. Default is no cost centers. You can create cost centers and associate them independently, using the create_cost_centers verb.

- separator

  Overrides the separator for multi-value input arguments, which is a semicolon (;). For information about overriding the separator or subseparator, see "Overriding the Separator and Subseparator" .

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Examples

Example 1

The following example changes the name of the finance organizational entity to investments and creates three associated cost centers, C4, C5, C6.

```
emcli create_organizational_entity
      -entity_name="finance"
      -new_entity_name="investments"
      -cost_centers="c4;c5;c6"
```

Example 2

The following example changes the entity type of the finance organizational entity to LOB. It also changes the tenant to which the entity belongs to corporate.

```
emcli create_organizational_entity
      -entity_name="finance"
      -entity_type="lob"
      -tenant_name="corporate"
```

update_paas_zone

Updates a PaaS Infrastructure Zone definition.

Format

```
emcli update_paas_zone
       -name="<Name_of_PaaS_Zone>"
       [-description="<Description_of_PaaS_Zone>"]
       [-credential="<Global_Named_Credential>"]
       [-add_hosts="<Host1,Host2,Host3...>"]
       [-remove_hosts="<Host4,Host5...>"]
       [-add_ovm_zones="<OVMZone1,OVMZone2,OVMZone3...>"]
       [-remove_ovm_zones="<OVMZone4,OVMZone5...>"]
       [-add_roles="<SsaRole1,SsaRole2,..>"]
       [-remove_roles="<SsaRole3,SsaRole4,..>"]
       [-cpu_utilization="<Value_between_1_and_100>"]
       [-memory_utilization="<Value_between_1_and_100>"]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the existing PaaS Infrastructure Zone.

- **description**

  Updated description of the PaaS Infrastructure Zone.

- **credential**

Global named credentials to be updated. These will be used for provisioning in this PaaS Infrastructure Zone. The credentials should be the same for all hosts. A cloud administrator can only use the named credentials that they own.

- **add_hosts**

  Comma-separated list of the host targets to be added as members of this Paas Infrastructure Zone. The hosts must not be members of other PaaS Zones.

- **remove_hosts**

  Comma-separated list of the host targets to be removed as members from this Paas Infrastructure Zone. The hosts must not be associated with any Software Pool member.

- **add_ovm_hosts**

  Comma-separated list of the OVMZone targets to be added as members of this Paas Infrastructure Zone. The OVMZones to be added must not be already added to other existing PaaS Zones.

- **remove_ovm_hosts**

  Comma-separated list of the OVMZone targets to be removed as members from this Paas Infrastructure Zone.

- **add_roles**

  Comma-separated list of SSA roles to be added to the list of roles that can access this PaaS Infrastructure Zone. A PaaS infrastructure zone can be made available to a restricted set of users through the use of roles.

- **remove_roles**

  Comma-separated list of SSA roles to be removed from the list of roles that can access this PaaS Infrastructure Zone.

- **cpu_utilization**

  Placement policy constraints allow the cloud administrator to set maximum resource ceilings for any host in the PaaS Infrastructure Zone. This provides protection for the members of the PaaS Infrastructure Zone in terms of resource consumption. For example, a production PaaS Infrastructure Zone might limit CPU utilization to 80%, whereas a development PaaS Infrastructure Zone might allow up to 95%. The service instance will be provisioned on the first host that satisfies the placement constraints.

  The value entered must be between 1 and 100. If not provided, the default value is taken to be 80%.

- **memory_utilization**

  A Placement Policy constraint for memory used by the PaaS Infrastructure Zone.The value entered must be between 1 and 100. If not provided, the default value is taken to be 80 percent.

Examples

Example 1

```
emcli update_paas_zone
-name="My PaaS Zone"
-add_hosts="host3.mycompany.com"

PaaS Infrastructure Zone "My PaaS Zone" updated successfully.
```

Example 2

```
emcli update_paas_zone
-name="My PaaS Zone"
-cpu_utilization="65"

PaaS Infrastructure Zone "My PaaS Zone" updated successfully.
```

update_password

Updates passwords or other credentials for a given target.

Format

```
emcli update_password
        -target_type="ttype"
        -target_name="tname"
        -credential_type="cred_type"
        -key_column="column_name:column_value"
        -non_key_column="col:oldvalue:newvalue;..."
        [-input_file="tag1:file_path1;tag2:file_path2;..."]

[ ]  indicates that the parameter is optional
```

Options

*   **target_type**

    Type of target.

*   target_name

    Name of the target.

*   **credential_type**

    Credential type to use. The type must be a base type, not a derived type. A derived type contains the XML tag <CredentialTypeRef> within its definition.

*   **key_column**

    Name and value of the key column for the credential type. Usually, the key column represents the user name. To get the key column for a target type, you can execute following EM CLI verbs:

    emcli show_credential_type_info — Displays key columns for all target types.

    emcli show_credential_type_info -target_type=<target_type> — Displays key columns for a specific target type.

*   **non_key_column**

    Name, old value, and new value of the non-key column(s) to modify. Usually, this is the name of the password column. Alternatively, a tag from the -input_file argument can be used so that the credential values are not seen on the command line. You can specify this option more than once.

*   **input_file**

    Path of the file that has non_key_column argument(s). This option is used to hide passwords. You must accompany each path with a tag that is referenced in the non_key_column argument. You can specify this option more than once.

You can obtain the list of columns and the credential types they belong to by using the emcli show_credential_type_info command.

For more information about the input_file parameter, see -input_file Syntax Guidelines.

Examples

Example 1

```
emcli update_password
        -target_type=oracle_database
        -target_name=myDB
        -credential_type=DBCreds
        -key_column="DBUserName:joe"
        -non_key_column="DBPassword:oldPass:newPass"
        -non_key_column="DBRole:normal:sysdba"
```

Example 2

In This example, FILE1 is a tag used to refer to the contents of passwordFile. The contents of the password file is:

```
DBPassword:oldPass:newPass;DBRole:normal:sysdba
```

Note that this example has the same effect as Example 1.

```
emcli update_password
        -target_type=oracle_database
        -target_name=myDB
        -credential_type=DBCreds
        -key_column="DBUserName:joe"
        -non_key_column="FILE1"
        -input_file="FILE1:passwordFile"
```

update_pool

Updates the details for a Software Pool.

Format

```
emcli update_pool
      -name="<Software_Pool_name>"
      -target_type="<Target_type_of_Software_Pool>"
      [-description="<Description_of_Software_Pool>"]
      [-add_members="<Member1, Member2...>"]
      [-remove_members="<Member4, Member5...>"]
      [-placement_constraints="<constraint1=value1,constraint2=value2...>"]
      [-properties="<property1=value1, property2=value2>"]

[ ]  indicates that the parameter is optional
```

Options

• **name**

Name of an existing Software Pool.

• **target_type**

Target type of the Software Pool. Example: "mwaas_zone" for Middleware Pool, "oracle_cloud_zone" for Database Pool, and "schaas_pool" for Schema Pool.

- **description**

  Description of the Software Pool.

- **add_members**

  Comma-separated list of targets to be added as members of the Software Pool. The targets to be added must satisfy the membership constraints of the Software Pool.

- **remove_members**

  Member targets to be removed from the Software Pool.

- **placement_constraints**

  Comma-separated key-value pairs of the placement constraints that allow the self-service administrator to set maximum ceilings for resource utilization. This provides protection for the members of the Software Pool in terms of resource consumption.

- **properties**

  Comma-separated key value pairs for properties that need to be updated based on the pool target type.

Examples

```
emcli update_pool
-name="My Pool"
-target_type="mwaas_zone"
-add_members="MyMember2"
```

displays the following output:

```
Software Pool "My Pool" updated successfully.
```

update_procedure_input

Updates the configuration of a deployment procedure.

Format

```
emcli upate_procedure_input
      -name="name_of_procedure_configuration"
      [-input_file="file_path\file_name"]
      [-grants="users_and_access_levels"]
      [-schedule=
          start_time:yyyy/MM/dd HH:mm;
          tz:<java_timezone_ID>;
          grace_period:xxx;
      ]
      [-notification="procedure status"]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the configuration for the procedure.

- **input_file**

  Input property file for the deployment procedure. The file_path should point to a file containing the data property file.

For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **grants**

  Specifies users and their corresponding access levels as a string of user:privilege pairs, each separated by a semi-colon (;). The user is an Enterprise Manager user name, and the privilege is either VIEW_JOB or FULL_JOB.

  See the example below.

- **schedule**

  Schedule for the deployment procedure. If not specified, the procedure is executed immediately.

  - **start_time** — When the procedure should start.

  - **tz** — Optional timezone ID.

  - **grace_period** — Optional grace period in minutes.

- **notification**

  Status of the procedure.

Example

```
emcli update_procedure_input
    -name=configProcedure
    -input_file=/home/data.properties -grants="user1:VIEW_JOB;user2:FULL_JOB"
    -schedule="start_time:2011/8/21 21:23;tz:America/New_York;grace_period:60"
    -notification="scheduled, action required, running"
```

update_service_template

Updates a Service Template.

Format

```
emcli update_service_template
    -name="<Service_Template_name>"
    -service_family="<Service_family_name>"
    -pool_target_type="<PoolTargetType>"
    [-add_software_pools="<SwPool1,SwPool2,SwPool3,...>"]
    [-remove_software_pools="RemovePool1,RemovePool2,RemovePool3,...>"]
    [-add_roles="<SsaRole1,SsaRole2,..>"]
    [-remove_roles="<RemoveSsaRole1,RemoveSsaRole2,..>"]
    [-description="<Updated_Description_of_Service_Template>"]
    [-input_file="data:<Name_of_Service_executable_MetaData_File>"]

[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the Service Template.

- **service_family**

  Service family to which the Service Template belongs. Example: DBAAS for Database, and MWAAS for Middleware.

- **pool_target_type**

  Target type of Software Pools to be associated with the Service Template.

- **add_software_pools**

  Comma-separated list of the Software Pools to be associated with this Service Template.

- **remove_software_pools**

  Comma-separated list of the Software Pools to be removed from this Service Template.

- **add_roles**

  Comma-separated list of SSA roles to be added to this Service Template. The SSA roles must already be created before attempting to add them to the Service Template.

- **remove_roles**

  Comma-separated list of SSA roles to be removed from this Service Template.

- **description**

  Description of the Service Template.

- **input_file**

  File containing configuration and profile data that will be required for updating values of procedure configuration variables. Format the data in JSON format. For example `input_file='data:executable.json'`

Examples

```
emcli update_service_template
-name="Middleware service template August"
-service_family="MWAAS"
-add_roles="SSA_USER_ROLE_1"
-remove_roles="SSA_USER_ROLE_2"
-add_software_pools="mwPool3,mwPool4"
-description="Updated description. Large instance size Service Template."
-input_file="data:executable.json"
```

displays the following output:.

```
Service Template "Middleware service template August" updated successfully.
```

update_siebel

Updates the Siebel enterprise.

Format

```
emcli update_siebel
     -enterprise=<Siebel enterprise>
     [-server=<Siebel server>]
     [-updateAutoStartModeComponentsOnly]
     [-review_only]
     [-out_file='<fully qualified path of output_file>']
     [-debug]

[ ]  indicates that the parameter is optional
```

Options

- **enterprise**

  Fully-qualified name of the Siebel enterprise in Enterprise Manager. For example, to update a Siebel enterprise '<Enterprise>' run the command `update_siebel -enterprise=<Enterprise>`.

- **server**

  Fully-qualified name of the Siebel server in Enterprise Manager.

- **updateAutoStartModeComponentsOnly**

  Indicates that the updateNow operation is performed for only the components with 'auto' mode on.

- **review_only**

  Indicates that the `updateNow` operation only displays the targets to be updated without actually saving them in the Enterprise Manager repository.

- **out_file**

  Fully-qualified path of `output_file.` The output of the command is redirected to this file.

- **debug**

  Executes the command in verbose mode and generates additional debug log messages in the output.

Example

This example updates the specified Siebel enterprise from Cloud Control, and the output of the command is redirected to file `update_output.txt.`

```
emcli update_siebel -enterprise=<Siebel enterprise> -
out_file='c:\emcli\update_output.txt' -debug
```

update_siteguard_configuration

Updates the Site Guard configuration to add additional standby systems. One primary system can be associated with one or more standby systems.

> **✏️ Note:**
>
> If you update the site configuration, you must also update the operation plan, as described in update_monitoring_creds_from_agent.

Format

```
emcli update_siteguard_configuration
      [-primary_system_name=<primary_system_name>]
      [-standby_system_name=<standby_system_name>]
```

Options

- **primary_system**

  Name of the primary system.

- **standby_system**

  Name of the standby system. You can specify this option more than once.

See Also

create_siteguard_configuration
delete_siteguard_configuration

Examples

```
emcli update_siteguard_configuartion
      -primary_system_name="BISystem1"
      -standby_system_name="BISystem2"
```

update_siteguard_credential_association

Updates the credential association.

Format

```
emcli update_siteguard_credential_association
      [-system_name=<system_name>]
      [-target_name=<target_name>]
      [-credential_type=<credential_type>]
      [-credential_name=<credential_name>]
      [-use_preferred_credential=true|false]
      [-credential_owner=<credential_owner>]
```

Options

- **system_name**

  Name of the system.

- **target_name**

  Optional name of the target.

- **credential_type**

  Type of credential, which can be HostNormal, HostPrivileged, WLSAdmin, or DatabaseSysdba.

- **credential_name**

  Name of the credential.

- **use_preferred_credential**

  Use a preferred credential instead of the named credential. You need to specify credential_name if this option is false.

- **credential_owner**

  Owner of the credential.

See Also

create_siteguard_credential_association
get_siteguard_script_credential_params
delete_siteguard_credential_association

Examples

Example 1

```
emcli update_siteguard_credential_association
        -system_name="austin-system"
        -credential_type="HostNormal"
        -credential_name="HOST-SGCRED"
        -credential_owner="sysman"
```

Example 2

```
emcli update_siteguard_credential_association
        -system_name="utah-system"
        -credential_type="HostPrivileged"
        -use_preferred_credential="true"
        -credential_owner="sysman"
```

update_siteguard_lag

Updates the limit for Apply lag and Transport lag for all databases or selected databases of the system.

Format

```
emcli update_siteguard_lag
        [-system_name="name_of_the_system"]
        [-target_name="name_of_the_target_database"]
        [-property_name=lag_type]
        [-value="lag_limit_in_seconds"]
```

[ ] indicates that the parameter is optional

Options

• **system_name**

  Name of the system whose lag limits you want to update.

• **target_name**

  Name of the database whose lag limits you want to update.

• **property_name**

  Name of the lag property. Valid values for this option are `ApplyLag` and `TransportLag`.

• **value**

  Time limit of the lag. Specify the values of this option in seconds.

Examples

Example 1

This example updates the Apply lag property with a lag limit of 1000 seconds on all of the databases configured on `austin-system`:

```
emcli update_siteguard_lag
      -system_name="austin-system"
      -property_name="ApplyLag"
      -value="1000"
```

Example 2

This example updates the Transport lag property with a lag limit of 2500 seconds on the `OID-db` database configured on `austin-system`:

```
emcli update_siteguard_lag
      -system_name="austin-system"
      -target_name="OID_db"
      -property_name="TransportLag"
      -value="2500"
```

update_siteguard_script

Updates the path and the all_hosts flag associated with any script.

Format

```
emcli update_siteguard_script
      -script_id=<script_ID>
      [-path=<script_path>]
      [-credential_type=<type_of_credential>]
      [-all_hosts=true|false]

[ ]  indicates that the parameter is optional
```

Options

*   **script_id**

    ID associated with the script.

*   **path**

    Optional path to the script.

*   **credential_type**

    Type of credential, which can be either HostNormal or HostPrivileged.

*   **all_hosts**

    Enables the script to run on all the hosts in the system. For example: true or false.

See Also

create_siteguard_script
get_siteguard_scripts

Examples

```
emcli update_siteguard_script
      -script_id="10"
      -path="/tmp/newprescript"
      -all_hosts="true"

emcli update_siteguard_script -script_id="16"
         -path="/tmp/script"
         -credential_type="HostPrivileged"
```

update_swlib_directive_entity

Modifies an entity of the Directive type in the Software Library. A new revision of the entity is created by default.

Format

```
emcli update_swlib_directive_entity
      -entity_rev_id="entity_rev_id"
      [-arg]="[<arg prefix>;]<arg prop name>[;<arg suffix>]"
      [-shell_type]="<shell type>"
      -[not_]run_privileged
      -file="<abs/relative file path>[;<new file name>]" | [-
removefile="<existing file name>"]
      [-upload_storage="<storage location name>;<storage type>"] | [-refer_
storage="<storage location name>;<storage type>"]
      [[-host="hostname"]
      [-credential_set_name="setname"] | [-credential_name="name" -
credential_owner="owner"]
      [-desc="entity_desc"]
      [-attr="<attr name>:<attr value>"]
      [-note="note text"]
      [-use_latest_revision]
      [-show_entity_rev_guid]
      [-show_cmd_line_and_exit]
```

```
[ ] indicates that the parameter is optional.
```

Parameters

* entity_rev_id

  Identifier of the entity revision. The Software Library Home page exposes the ID for folders and entities as a custom column (Internal ID). However, this is hidden by default.

* arg

  Command line argument property name, specified optionally with a prefix and/or a suffix. To specify multiple arguments, repeat the option.

* shell_type

  Shell type can be one of the following:

  – SUB_Exec - Specified in the script

  – SUB_Perl - Perl

  – SUB_Bash - Bash

Default value is SUB_Bash - Bash.

- run_privileged

  This is an option to specify whether the directive should be executed with privileged credentials or not. Will be executed with normal credentials by default.

- file

  If `-upload_storage` is specified, it is the absolute path of the file that is uploaded. If `-refer_storage` is specified, it is the relative path of the file that is referred from the storage location specified. File name stored in the Software Library after the file is uploaded is defaulted to the name of the file being uploaded/referred. A different file name can be specified, optionally, separated by ';'. The first file specified in the command line will be defaulted as the main file and will be executed when the directive is run.

- removefile

  Name of the file to be removed. This is an existing file carried forward from the specified entity revision. Alternatively, the following values may be specified:

  – ALL - to remove all existing files

  – NONE - to retain all carried forward files

    Default value is NONE.

- host

  Target name of the host where the files are available. This should be used in conjunction with the `-upload_storage` option.

- credential_set_name

  The set name of the preferred credential stored in the Management Repository for the host target. This can be one of the following:

  – HostCredsNormal - default unprivileged credential set

  – HostCredsPriv - privileged credential set

    This should be used in conjunction with `upload_storage` option.

- credential_name

  Named credential stored in the Management Repository. This option must be specified along with the -credential_owner option. This must be used in conjunction with the -upload_storage option.

- credential_owner

  Owner of a named credential stored in the Management Repository. This option must be specified along with the `-credential_name` option. This must be used in conjunction with the `-upload_storage` option.

- upload_storage

  Destination storage location and type for the upload, separated by ';'. The location specified must be in the 'active' status. Defaulted to storage type and location of the first upload location configured for Software Library. The storage type can be one of the following:

  – OmsShared (OMS Shared File System)

  – OmsAgent (OMS Agent File System)

- refer_storage

Storage location and type for referring to files, separated by ';'. The location specified must be in the 'active' status. The storage type can be one of the following:

- http
- NFS
- ExtAgent

If specified, this option takes precedence over the `-upload_storage` option.

- desc

  Description of the entity. The new description is visible to all existing revisions.

- attr

  Attribute and its value is separated by ':'. To specify multiple attributes, repeat the option.

- note

  A note about the entity. Repeat the option for multiple notes.

- use_latest_revision

  An option to indicate whether the latest revision of the entity should be updated or the revision identified by the entity_rev_id.

- show_entity_rev_guid

  Option to enable printing of the internal GUID of the new entity revision or not.

- show_cmd_line_and_exit

  Option to enable printing of the command line and exiting without creating a new entity revision or not.

Example

The following example modifies a directive entity identified by `entity_rev_id`. The entity revision ID value can be found from the Software Library Home page. The Software Library Home page exposes the ID for folders and entities, as a custom column. However, this is hidden by default.

```
emcli update_swlib_directive_entity
    -entity_rev_id="oracle:defaultService:em:provisioning:1:cmp:COMP_
Directives:none:06865A85D99C5340E0530AD5F00A4E0F:0.6"
    -arg="-home :oh_home"
    -shell_type=SUB_Perl
    -file=/u01/scripts/myscript.pl
    -file=/u01/scripts/mymodule.pm;common.pm
```

update_swlib_entity

Modifies an entity in the software library. A new revision of the entity is created by default. Changing only the description or attribute values does not create a new revision, and such changes will be visible across all existing revisions of the entity.

Format

```
emcli update_swlib_entity
      -entity_rev_id="entity_rev_id"
```

```
[-desc="entity_desc"]
[-attr="<attr_name>:<attr_value>"]
[-prop="<prop_name>:<prop_value>"]
[-secret_prop="<secret_prop_name>:<secret_prop_value>"]
[-note="note_text"]
[-use_latest_revision]
```

[ ] indicates that the parameter is optional

Options

- **entity_rev_id**

  Identifier of the entity revision. The software library home page exposes the identifier for folders and entities as a custom column (Internal ID) and is hidden by default.

- **desc**

  Description of the entity. The new description is visible to all existing revisions.

- **attr**

  An attribute and its value, separated by a colon (:). To specify values for multiple attributes, repeat this option. The new attribute value is visible to all existing revisions.

- **prop**

  Configuration property and its value, separated by a colon (:). To specify values for multiple attributes, repeat this option.

- **secret_prop**

  Configuration property and its secret value separated by a colon (:). It is recommended that the secret value not be specified on the command line. If omitted from the command line, the value is prompted for. To specify values for multiple properties, repeat this option.

- **note**

  Note on the entity. For multiple notes, repeat this option.

- **use_latest_revision**

  Indicates that the the latest revision of the entity should be updated instead of the revision identified by entity_rev_id.

Examples

This example modifies the entity revision identified by entity_rev_id. The entity revision identifier value can be found from the Software Library home page. The software library home page exposes the identifier for folders and entities as a custom column, which is hidden by default.

A new description is specified. Values for the entity attributes (PRODUCT, PRODUCT_VERSION and VENDOR) are specified. The value for the DEFAULT_HOME configuration property is specified. A note on the entity is also specified.

A new revision is created for the modifications, but the specified entity revision (identified by entity_rev_id) remains unchanged. The identifier of the newly created entity is printed on the standard output.

```
entity_rev_id="oracle:defaultService:em:provisioning:1:cmp:COMP_Component:SUB_Generic:B
1B1880C6A8C62AAE040548C4D14:0.1"
        -entity_desc="myAcmeInstall description"
```

```
-attr="PRODUCT:Acme"
-attr="PRODUCT_VERSION:3.0"
-attr="VENDOR:Acme Corp"
-prop="DEFAULT_HOME:/u01/acme3/"
-note="myAcmeInstall for test servers"
```

update_target_password

Updates the changed target password in the Enterprise Manager credential sub-system. For collection or monitoring credentials, the password change is also propagated to Enterprise Manager Management Agents.

Format

```
emcli update_target_password        -target_type="ttype"        -
target_name="tname"        -key_column="column_name:column_value"        [-
change_all_references="yes/no"]        [-
input_file="tag1:file_path1;tag2:file_path2;..."]

[ ]  indicates that the parameter is optional
```

> **Note:**
>
> When you execute this verb, you are prompted to enter the following values in non-echo mode:
>
> -old_password -new_password -retype_new_password

Options

- **target_type**

  Type of target.

- **target_name**

  Name of the target.

- **key_column**

  Name and value of the key column for the credential type. The key column usually represents the user name.

  To obtain the key column for a target type, enter the following command:

  ```
  emcli show_credential_type_info -target_type=<target_type>"
  ```

  To obtain the key column for all target types, enter the following command:

  ```
  emcli show_credential_type_info
  ```

  To obtain the key column for a target type, enter the following command:

  ```
  emcli show_credential_type_info -target_type=<target_type>"
  ```

- **change_all_references**

  Specifies if the password must be changed for all references in Enterprise Manager for the given user.

Possible values are:

– Yes — Updates all references in Enterprise Manager for this password.

– No — Updates the password for the current logged-in user. This is the default.

- **input_file**

    File path that has old and new passwords. This **option** hides passwords. You must accompany each path with a tag referenced in the password **option**s. You can specify this **option** more than once.

    For more information about the input_file parameter, see -input_file Syntax Guidelines.

Examples

Example 1

This example asks the user to enter the values of the old and new passwords, then retype the new password to update the new password in Enterprise Manager for this target reference.

```
emcli update_target_password    -target_type=host    -target_name=myHost    -
key_column=HostUserName:Admin1
```

Example 2

This example asks the user to enter the values of the old and new passwords, then retype the new password to update the new password in Enterprise Manager for all users' credentials referenced with the mydb target name and Admin1 user name.

```
emcli update_target_password    -target_type=oracle_database    -
target_name=mydb    -key_column=DBUserName:Admin1    -change_all_references=yes
```

update_tenant_owner

Changes the owner for a tenant.

Format

Standard Mode

```
emcli update_tenant_owner
    -name="name"
    -new_owner="new_owner"
```

Interactive or Script Mode

```
emcli update_tenant_owner(
    name="name"
    ,-new_owner="new_owner"
)
```

Options

- name

    The name of the tenant.

- new_owner

    The user name of the new tenant owner.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example changes the name of the owner of `my_tenant` to `John`.

```
emcli update_tenant_owner
      -name="my_tenant"
      -new_owner="John"
```

update_tenant_state

Marks an Enterprise Manager tenant as Active or Inactive.

Format

Standard Mode

```
emcli update_tenant_state
      -name="tenant name"
      -active="true/false"
```

Interactive or Script Mode

```
emcli update_tenant_state(
      name="tenant name"
      ,-active="true/false"
)
```

Options

* name

  The name of the tenant whose status is to be updated.

* active

  Sets the status of the tenant. The status "true" indicates an active state, and "false" indicates that the tenant is marked as inactive.

Exit Codes

0 if successful. A non-zero value indicates that verb processing was unsuccessful.

Example

The following example marks the status of `my_tenant` as inactive.

```
emcli update_tenant_state
      -name="my_tenant"
      -active="false"
```

update_ticket_status

Updates the ticket status and last modified time stamp in Enterprise Manager from the external ticketing system based on the ticket_guid and connector_guid.

Format

```
emcli update_ticket_status   -ticket_guid="ticket guid"   -connector_guid="connector
guid"   -status="Incident status"   -last_updated_date="last modified date"   -
date_format=
```

Options

- **ticket_guid**

  Ticket ID for which the status is modified.

- **connector_guid**

  Ticketing Connector ID.

- **status**

  Modified status of an incident ticket.

- **last_updated_date**

  Specifies the last modified date of an incident ticket.

- **date_format**

  Specify a date format followed in the Ticketing System, as in "MM/dd/yyyy hh:mm:ss" if
  the date field in Incident management is "10/13/2009 5:38:24 AM".

Example

This example updates the ticket INC00000024 status as 'In Progress' in Enterprise Manager
after the same ticket status was recently modified on the ticketing system.

```
emcli update_ticket_status
  -ticket_guid="INC21000024"
  -connector_guid="ccc1234"
  -status="2"
  -last_updated_date="05/28/2011 3:14:56PM"
  -date_format="MM/dd/yyyy hh:mm:ss"
```

upgrade_agents

Performs Agent upgrade prerequisites and submits the Agent upgrade job.

Format

```
emcli upgrade_agents
    -agents="full_agent_name"|-input_file="agents_file:location_of_output_file"
    [-validate_only]
    [-pre_script_loc]
    [-post_script_loc]
    [-pre_script_on_oms]
    [-post_script_on_oms]
    [-stage_location]
    [-job_name]
    [-override_credential]
    [-additional_parameters]

[ ]  indicates that the parameter is optional
```

Options

> **Note:**
>
> Either the -agents or -input_file parameter is mandatory. If you provide both, the union of both are taken, prerequisites are performed on the Agents, and an Agent upgrade job is submitted.
>
> You can pass all of these parameters in a response file. Usage: -input_file="response_file:/scratch/response_file.txt" . A file name with the full path must be provided, and each parameter should be specified in each line. If a parameter/flag is passed both in the command line and in a response file, the command-line option is given precedence. A parameter should be specified as a name-value pair in the response file. For example: job_name=UPGRADE_AGT_121020

- **agents**

  Checks whether the specified Agents specified are upgradable, and submits an Agent upgrade job.

- **input_file**

  Checks whether the Agents specified in file are upgradable, and submits an Agent upgrade job.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **validate_only**

  Checks only whether Agents specified are upgradable. An Agent upgrade job will not be submitted.

- **pre_script_loc**

  Executes this script before upgrading the Agent.

- **post_script_loc**

  Executes this script after upgrading the Agent.

- **pre_script_on_oms**

  Use if pre-script is treated to be on OMS.

- **post_script_on_oms**

  Use if post-script is treated to be on OMS.

- **stage_location**

  Passes a custom staging location used by the Agent upgrade job.

- **job_name**

  Submits the job with this name.

- **override_credential**

  Preferred credential of the Oracle home of the Agent used to run root.sh. Use this option to override this and use a named Oracle home credential.

- **additional_parameters**

   Passes additional parameters to the Agent upgrade job.

Examples

Example 1

This example checks whether the Agents matching pattern abc% and xyz.domain.com:1243 are upgradable, then submits the Agent upgrade job.

```
emcli upgrade_agents -agents="abc%,xyz.domain.com:1243"
```

Example 2

This example checks whether Agents in the file are upgradable, then submits the Agent upgrade job.

```
emcli upgrade_agents -input_file="agents_file:/scratch/agents_file.txt"
```

upgrade_database

Upgrades a database.

Format

```
emcli upgrade_database
      -dbTargetName="target_to_be_upgraded"
      -dbTargetType="oracle_database|rac_database"
      -newOracleHome="directory_full_path"
      -hostCreds="named_credentials"
      -sysdbaCreds="named_credentials"
      [-precheck="YES|NO|ONLY"
   [-ignoreWarnings]
      [-diagnosticDest="diagnostic_destination"]
      [-disableArchiveLogMode]
      [-recompileInvalidObjects]
   [[-restoreSettingsOnly] | [-backupLocation="backup_location_full_path"]]
      [-listeners=<name:port[:NEW]>
   [-scriptsFromSoftwareLibrary "scripts_from_software_library"]
      [-beforeUpgradeCustomScript="custom_SQL_file_name"]
   -continueOnScriptError
      [-afterUpgradeCustomScript="Custom_SQL_file_name_upgrade"]
      [-noBlackout]

[ ]  indicates that the parameter is optional
```

Options

- **dbTargetName**

   Enterprise Manager target name of the database to be upgraded. Versions 10.2.0.4 and above are supported for upgrade.

- **dbTargetType**

   Target type of the database — oracle_database for a single instance database, or rac_database for a cluster database.

- **newOracleHome**

New Oracle Home directory full path. Upgrade to 11g Release 2 and later is supported. Does not support a database downgrade.

- **hostCreds**

  Named host credentials of the user who owns the Oracle Home installation. Should have necessary privileges on the database files to be upgraded.

- **sysdbaCreds**

  Named database credentials having SYSDBA privileges on the database to be upgraded.

- **precheck**

  Option to run prerequisite checks during the upgrade job. Valid values are:

  YES — Run prerequisite checks and proceed to the database upgrade if there are no errors during prerequisite checks.

  NO — Proceed to the database upgrade directly. Do not run prerequisite checks.

  ONLY — Run prerequisite checks only. Do not upgrade the database.

- **ignoreWarnings**

  Ignores any warnings during prerequisite checking and proceeds with the upgrade. Used only when pre-check is set to YES, otherwise ignored. Does not ignore errors.

- **diagnosticDest**

  Full directory path for Oracle trace and diagnostic files for the upgraded database. By default, ORACLE_BASE is used as the location.

- **disableArchiveLogMode**

  Disable archive logging during the database upgrade.

- **recompileInvalidObjects**

  The upgrade process may invalidate the objects in the database. You can choose to recompile invalid objects at the end of the upgrade. This increases the upgrade time, but minimizes subsequent latencies caused by on-demand automatic recompilation at run time.

- **restoreSettingsOnly**

  Reverts only the configuration changes made during the upgrade if upgrade fails. You can restore the database outside the upgrade using your custom restore strategy. Choose this **option** if you already have a custom backup and restore strategy for this database. In case of an upgrade failure, this setting will be used.

- **backupLocation**

  Full directory path to back up the database. Performs a full backup of the database. A script will be created to restore the database. All files are placed in the specified backup location. Reverts all the changes made during the upgrade if the upgrade fails.

- **listeners**

  Comma-separated list of the listener name and port (name1:port1,name2:port2) to register the upgraded database. Specify at least one listener in the case of a single-instance database target. These listeners should be configured in the new Oracle home or TNS_ADMIN location. Additionally, you can choose to create a new listener in the new Oracle home by specifying :NEW (name1:port1:NEW).

- **scriptsFromSoftwareLibrary**

  Specify the custom scripts from the software library components. The parameters 'beforeUpgradeCustomScript' and 'afterUpgradeCustomScript' are interpreted as entity URNs of the components that contain the scripts.

- **beforeUpgradeCustomScript**

  Full file path of the custom SQL script to be run before the database upgrade.

- **continueOnScripteError**

  Ignores a non-zero exit code when executing a custom SQL script and continues the upgrade job.

- **afterUpgradeCustomScript**

  Full file path of the custom SQL script to be run after the successful database upgrade.

- **noBlackout**

  Suppresses a blackout of the database target. A blackout suspends monitoring of the database target from Enterprise Manager, which is the default behavior during a database upgrade.

Examples

```
emcli upgrade_database
   -dbTargetName=test1 -dbTargetType=oracle_database
   -newOracleHome=/u01/app/oracle/product/11.2.0/dbhome_2 -hostCreds=HOST_CREDS
   -sysdbaCreds=SYSDBA_CREDS -precheck=YES -ignoreWarnings -disableArchiveLogMode
   -beforeUpgradeCustomScript=/home/user1/sqlfiles/script1.sql
   -continueOnScriptError
   -afterUpgradeCustomScript=/home/user1/sqlfiles/script2.sql
   -diagnosticDest=/u01/app/oracle
   -recompileInvalidObjects -noBlackout
```

upload_ats_test_databank_file

Uploads a databank file for the specified ATS test.

Format

```
emcli upload_ats_test_databank_file
       -name=<target_name>
       -type=<target_type>
       -testname=<test_name>
       -testtype=<test_type>
       -databankAlias=<databank_alias>
       -input_file:databank=<databank_file>
       [-beaconName=<beacon_name>]
```

```
[ ]  indicates that the parameter is optional
```

Options

- **name**

  Name of the target.

- **type**

  Name of the target type.

- **testname**

  Name of the test.

- **testtype**

  Type of test.

- **databankAlias**

  Databank alias.

- **input_file**

  Databank file.

  For more information about the input_file parameter, see -input_file Syntax Guidelines.

- **beaconName**

  Beacon name.

Examples

Example 1

This example uploads the databank file corresponding to the specified test.

```
emcli upload_ats_test_databank_file
      -name="Service Name"
      -type="generic_service"
      -testname="Test Name"
      -testtype="OATS"
      -databankAlias="alias1"
      -input_file="databank:databankFile.csv"
```

Example 2

This example uploads the databank file corresponding to the specified test for the specified beacon.

```
emcli upload_ats_test_databank_file
      -name="Service Name"
      -type="generic_service"
      -testname="Test Name"
      -testtype="OATS"
      -databankAlias="alias1"
      -input_file="databank:databankFile.csv"
      -beaconName="Beacon Name"
```

upload_catalog

Uploads the catalog file 'em_catalog.zip' to the software library and submits a Refresh From My Oracle Support job to process the file. You can download the catalog file 'em_catalog.zip' from https://updates.oracle.com/download/em_catalog.zip. You cannot run this command when Enterprise Manager Cloud Control is running in online mode. When Enterprise Manager Cloud Control runs in online mode, the catalog file is downloaded by the Refresh From My Oracle Support job daily.

Format

```
emcli upload_catalog
      -from_host="host_name"
      -file="absolute_path_of_catalog_file"
      [-cred_name="credential_name" [-cred_owner="credential_owner"]]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- from_host

  Specifies the host where the catalog file 'em_catalog.zip' is located.

- file

  Specifies the absolute path of the catalog file 'em_catalog.zip' on the host specified for the -from_host option. The catalog file can be accessed by the Management Agent installed on the specified host.

- cred_name

  Specifies the named credentials that must be used to access the specified host. If you do not specify this option, the preferred normal credentials of the specified host will be used by default.

- cred_owner

  Specifies the owner of the named credentials that must be used to access the specified host. If you do not specify this option, the current login user is assumed to be the owner of the named credentials by default.

Examples

Example 1

The following example uploads the em_catalog.zip file, which is located at /scratch/em_catalog.zip on h1.example.com, to the software library:

```
emcli upload_catalog
      -file="/scratch/em_catalog.zip"
      -from_host=h1.example.com
```

Example 2

The following example uploads theem_catalog.zip, which is located at /scratch/em_catalog.zip on h1.example.com, to the software library, using the named credentials AIMECRED:

```
emcli upload_catalog
      -file="/scratch/em_catalog.zip"
      -from_host=h1.example.com
    -cred_name=AIMECRED
```

upload_compliance_standard

Uploads into Enterprise Manager SCAP XCCDF standards through an XML file.

Format

```
$ emcli upload_compliance_standard -file="<STANDARD_FILE>.xml"
```

Options

file
XML data stream file containing one or more standards.

Example

```
$ emcli upload_compliance_standard -file="ssg-ol8-ds.xml"
```

upload_jeeappcomp_file

Uploads one file to a Java EE Application component in the software library.

This command needs the verb create_jeeappcomp to function properly.

Format

```
emcli upload_jeeappcomp_file
        -entity_rev_id="entity_rev_id"
        -host="hostname"
        -filetype="filetype"
        -file="<absolute file path>[;<new file name>]"
        [-credential_set_name="setname"] | [-credential_name="name" -
credential_owner="owner"]
        [-upload_storage="<storage location name>;<storage type>"]
        [-use_latest_revision]

[ ]  indicates that the parameter is optional
```

Options

* **entity_rev_id**

  Identifier for the entity revision. You can view the entity ID by logging in to the
  Cloud Control console. The Software Library Home page exposes the identifier for
  folders and entities as a custom column called Internal ID. By default, this is
  hidden.

* **host**

  Target name of the host where the files are available.

* **filetype**

  The file type of the file specified with '-file'. Valid values include: "archive", "plan",
  "pre_deployment_script", "post_deployment_script", "target_execution_script",
  "additional_file", "zip".

* **file**

  Absolute path of the file to be uploaded. File name stored in the Software Library
  is overwritten with the name of the file being uploaded. Optionally, you can specify
  a different file name, separated by ";".

* **credential_set_name**

The name of the preferred credential stored inside the Management Repository for the host target. It can be one of the following:

HostCredsNormal - default unprivileged credential set.

HostCredsPriv - privileged credential set.

- **credential_name**

  Name of a named credential stored in the Management Repository. This option must be specified along with the -credential_owner option.

- **credential_owner**

  Owner of a named credential stored in the Management Repository. This option must be specified along with the -credential_name option.

- **upload_storage**

  Destination storage location and type of storage is passed as a parmeter for the upload, separated by ";". If no value is passsed, then the storage type and storage location are defaulted to the first upload location configured for Software Library. The storage type can be one of the following:

  OmsShared (OMS Shared File System)

  OmsAgent (OMS Agent File System)

  The storage location specified must be in "active" status.

- **use_latest_revision**

  Flag for indicating that the upload should happen to the latest revision of the entity instead of the revision identified by entity_rev_id.

Examples

Example 1

Uploads the file '/u01/downloads/file1.jar' to the entity revision identified by entity_rev_id. The file present on the host 'example.com' should be accessible using the preferred credential set for the "HostCredsNormal" credential set, for the user logged in to EMCLI.

The host must be a managed host target in Enterprise Manager and the agent on this host must be up and running.

```
emcli upload_jeeappcomp_file
-
entity_rev_id="oracle:defaultService:em:provisioning:1:cmp:COMP_Component:SUB_JavaEEApp
lication:B1B1880C6A8C62AAE040548C42832D14:0.1"
-file="/u01/downloads/file1.jar"
-filetype="archive"
-host="example.com"
-credential_set_name="HostCredsNormal"
```

Example 2

Uploads the file newfile.xml to the entity revision identified by entity_rev_id. The file present on the host 'example.com' should be accessible using the credential named'"MyExampleCreds" owned by "EXAMPLE_USER". The entity revision specified should contain a file of filetype 'archive' before uploading any other filetype. File '/u01/downloads/file2.xml', after upload, will be associated with the entity revision as 'newfile.xml'. A new revision will be created from the latest revision of the entity.

```
emcli upload_jeeappcomp_file
-
entity_rev_id="oracle:defaultService:em:provisioning:1:cmp:COMP_Component:SUB_Jav
aEEApplication:B1B1880C6A8C62AAE040548C42832D14:0.1"
-file="/u01/downloads/file2.xml;newfile.xml"
-filetype="plan"
-host="example.com"
-credential_name="MyExampleCreds"
-credential_owner="EXAMPLE_USER"
-use_latest_revision
```

upload_patches

Uploads patches to the software library.

Format

```
emcli upload_patches
        -from_host="host_name"
        -patch_files="metadata_file_path;ZIP_file_path"
        [-cred_name="name" -cred_owner="owner"]
```

```
[ ]  indicates that the parameter is optional
```

Options

* **from_host**

  Host from which to get the given patch files.

* **patch_files**

  List of patch file paths. A metadata file and a zip file must be provided.

* **cred_name**

  Named credential used to authenticate the host from which the given patch files
  are to be uploaded. If you do not provide this option, the normal preferred
  credential of the host from which the given patch files are to be uploaded will be
  used by default.

* **cred_owner**

  Owner of the named credentials used to authenticate the host from which the
  given patch files are to be uploaded.

Examples

```
emcli upload_patches -patch_files="/scratch/p13741363_112310_Linux-x86-64_M.xml;/
scratch/p13741363_112310_Linux-x86-64.zip" -from_host=h1.example.com
```

```
emcli upload_patches -patch_files="/scratch/p13741363_112310_Linux-x86-64_M.xml;/
scratch/p13741363_112310_Linux-x86-64.zip" -from_host=h1.example.com -
cred_name=AIMECRED -cred_owner=SYSMAN
```

Upload a multi-part patch

Patch 23274210 consists of Metadata split into different patch files. Download the
patch files to a location with enough space. Keep in mind the patch's preparation will
require about three times the total space of the zipped patch.

1. Upload the patch.

```
emcli upload_patches -from_host=<host_name>
-patch_files=/<PATCH_LOCATION>/p23274210_121020_American.txt;/<PATCH_LOCATION>/
p23274210_121020_Linux-x86-64_1of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_2of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_3of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_4of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_5of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_6of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_7of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_8of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_9of10.zip;\
/<PATCH_LOCATION>/p23274210_121020_Linux-x86-64_10of10.zip\
 -cred_name=<X_OMS_HOST_N> -cred_owner=<EM_USER>
```

The command will output a URL that will help track progress of the overall activity.

2. After the command is complete, go to the **Job Activity** page and **Submit** a job: "Download System Patch and Evaluate Patch Recommendations".

3. The job will generate the recommendations, navigate to **Software Update** page and search for Exadata Targets. **Select** QFSP. The patch will now be shown as recommended or applicable to the specified target.

See Also

create_patch_plan
delete_patches
describe_patch_plan_input
get_connection_mode
get_patch_plan_data
list_aru_languages
list_aru_platforms
list_aru_products
list_aru_releases
list_patch_plans
search_patches
set_connection_mode
set_patch_plan_data
show_patch_plan

submit_patch_plan

Also, see Patching Using EM CLI in the *Enterprise Manager Lifecycle Management Administrator's Guide*.

upload_swlib_entity_files

Uploads one or more files to an entity revision in the software library.

Format

```
emcli upload_swlib_entity_files
      -entity_rev_id="entity_rev_id"
      -file="<abs_file_path>[;<new_file_name>]" | [-removefile="<existing_
        file_name>"]
      -host="hostname"
      [-credential_set_name="setname"] | [-credential_name="name"
        -credential_owner="owner"]
```

```
[-upload_storage="<storage_location_name>;<storage_type>"]
[-use_latest_revision]
```

`[ ]` indicates that the parameter is optional

Options

- **entity_rev_id**

  Identifier of the entity revision. The software library home page exposes the identifier for folders and entities as a custom column (Internal ID), and is hidden by default.

- **file**

  Absolute path of the file to be uploaded. The file name stored in the software library on upload defaults to the name of the file being uploaded. You can optionally specify a different file name, separated by a semicolon (;).

- **removefile**

  Name of the file to be removed. This is an existing file carried forward from the specified entity revision.Alternatively, you can specify the following values:

  ALL — Remove all existing files (no carry forward).
  NONE — Retain all carried forward files.

  The default is NONE.

- **host**

  Target name of the host where the files are available.

- **credential_set_name**

  Set name of the preferred credential stored in the repository for the host target, which can be one of: HostCredsNormal — Default unprivileged credential set HostCredsPriv — Privileged credential set

- **credential_name**

  Name of a named credential stored in the repository. You must specify this option along with the credential_owner option.

- **credential_owner**

  Owner of a named credential stored in the repository. You must specify this option along with the credential_name option.

- **upload_storage**

  Destination storage location and type for the upload, separated by a semicolon (;). The location specified must be in 'active' status. This defaults to the storage type and location of the first upload location configured for the software library.

  The storage type can be one of:

  OmsShared — OMS shared file system

  OmsAgent — OMS Agent file system

- **use_latest_revision**

  Flag indicating that the upload should occur to the latest revision of the entity instead of the revision identified by entity_rev_id.

Examples

Example 1

This example uploads the file '/u01/acme_downloads/file1.zip' to the entity revision identified. The file present on the host 'fs1.us.acme.com' should be accessible using the preferred credential set for the 'HostCredsNormal' credential set for the user logged in to EM CLI. The host must be a managed host target in Enterprise Manager, and the Agent on this host must be up and running.

```
emcli upload_swlib_entity_files
        -entity_rev_id="oracle:defaultService:em:provisioning:1:
          cmp:COMP_Component:SUB_Generic:
          B1B1880C6A8C62AAE040548C42832D14:0.1"
        -file="/u01/acme_downloads/file1.zip"
        -host="fs1.us.acme.com"
        -credential_set_name="HostCredsNormal"
```

Example 2

This example uploads the files to the specified entity revision. The file present on the host 'fs1.us.acme.com' should be accessible using the credential named 'MyAcmeCreds' owned by 'ACME_USER'. File '/u01/acme_downloads/file1.zip' after upload will be associated with the entity revision as 'newfile1.zip'. A new revision will be created from the latest revision of the entity.

```
emcli upload_swlib_entity_files
        -entity_rev_id="oracle:defaultService:em:provisioning:1:
          cmp:COMP_Component:SUB_Generic:
          B1B1880C6A8C62AAE040548C42832D14:0.1"
        -file="/u01/acme_downloads/file1.zip;newfile1.zip"
        -file="/u01/acme_downloads/file2.zip"
        -host="fs1.us.acme.com"
        -credential_name="MyAcmeCreds"
        -credential_owner="ACME_USER"
        -use_latest_revision
```

use_target_properties_master_list

Enables or disables the Master List for a property.

Format

Standard Mode

```
    emcli use_target_properties_master_list
            -property_name="null"
            [-enable]
            [-disable]
            [-copy_from_targets]

[ ]  indicates that the parameter is optional.
```

Interactive or Script Mode

```
use_target_properties_master_list(
                property_name="null"
            [,enable=True/False]
                [,disable=True/False]
```

```
                        [,copy_from_targets=True/False]
                        )
```

Options

- -property_name

  Property name.

- -enable

  Enable master list.

- -disable

  Disable master list.

- -copy_from_targets

  If any of the targets have values set for the specified property previously, then you must use this option. If this situation occurs, this verb will raise an error and prompt you to use this option. All the values from the targets will be added to Master List by default.

Examples

Example 1

The following command enables the use of the Master List.

```
emcli use_target_properties_master_list
    -enable -property_name="orcl_gtp_location"
```

Example 2

The following command disables the use of the Master List.

```
emcli use_target_properties_master_list
      -disable -property_name="orcl_gtp_location"
```

validate_server_generated_alerts

Compares and synchronizes database server-generated alert metric thresholds between Enterprise Manager and target database(s). You can run the verb by specifying one or more specific database target(s) or an Enterprise Manager Group. If you specify a group, compare and synchronize operations are run for all database targets in the group and all of its sub-groups.

This verb generates one row for each server-generated alert metric threshold that is out-of-sync. Each row contains details, such as metric name, object name, critical, and warning thresholds in Enterprise Manager and the target database. Each of the values are separated by a comma.

Format

```
emcli validate_server_generated_alerts
      [-group="group_name" [-fix] [-verbose]]
      [-targets="<target_name1:target_type1;target_name2:target_type2>"
          [-fix]
          [-verbose]]
      [-help]
```

[ ] indicates that the parameter is optional.

Options

- group

  Name of the Enterprise Manager group.

- targets

  Target Name, Target Type pair(s). This option is mutually exclusive with the -group option.

- fix

  Synchronizes server-generated alert metric thresholds between Enterprise Manager and the specified database target(s). Server -generated alert metric thresholds in Enterprise Manager are pushed to the target database(s) during the synchronize operation.

- verbose

  Provides a detailed report of out-of-sync thresholds.

- help

  Shows help messages.

Notes

- This verb may not push some metrics to the target database due to non-existent Object(s). For example: non-existent tablespace.

- This verb's default output format is comma-separated values (CSV).

Examples

Example 1

This example compares thresholds between Enterprise Manager and 'ProdDb' (including RAC instances) and 'TestDb' individually.

```
emcli validate_server_generated_alerts
-targets="ProdDb:rac_database;TestDb:oracle_database"
```

Example 2

This example compares and fixes thresholds between Enterprise Manager and target databases 'ProdDb' and 'TestDb' individually.

```
emcli validate_server_generated_alerts
-targets="ProdDb:rac_database;TestDb:oracle_database" -fix
```

verify_adm

Submits a Verify Application Data Model job for the given Application Data Model and target.

Format

```
emcli verify_adm
    -adm_name=<application_data_model_name>
    -target_name=<target_name>
    -target_type=<target_type>
    [-job_name=<job_name>]
```

```
[-db_cred_name=<database_named_credentials>]
[-db_pref_creds_name=<database_preferred_credentials>]
[-job_description=<job_description>]
```

```
[ ]  indicates that the parameter is optional.
```

Options

- adm_name

  Application Data Model name for which the job will be submitted for verification.

- target_name

  Name of the target for which the verify ADM job will be submitted. This can either be the source database or any of the associated databases for the ADM.

- target_type

  Type of target for which the verify aDM job will be submitted for the Application Data Model.

- job_name

  Name of the job to be submitted.

- db_cred_name

  Name of named database credentials stored in the Enterprise Manager repository.

- db_pref_creds_name

  Name of preferred database credentials stored in the Enterprise Manager repository. The valid values for this option are:

  – DBCredsNormal — Default normal credential set for a database target.

  – DBCredsSYSDBA — SYSDBA credential set for a database target.

  You must provide a value of either db_pref_creds_name or db_cred_name for the successful submission of the job.

- job_description

  Description of the job to be submitted.

Output

Success/error messages.

Example

This example submits a job with name 'verify adm' on the target test_database for the application data model Sample_ADM.

```
emcli verify_adm
      -adm_name=Sample_ADM
      -target_name=test_database
      -target_type=oracle_pdb
      -job_name="verify adm"
      -db_cred_name=NC_testdb

emcli verify_adm
      -adm_name=Sample_ADM
      -target_name=test_database
      -target_type=oracle_pdb
```

```
        -job_name="verify adm"
        -db_cred_name=NC_testdb
        -job_description="verify adm job on test_database"

emcli verify_adm
        -adm_name=Sample_ADM
        -target_name=test_database
        -target_type=oracle_database
        -job_name="verify adm"
        -db_pref_creds_name=NC_testdb_pref
        -job_description="verify adm job on test_database"
```

verify_config (deprecated)

Verifies the health of the Data Guard configuration of a primary database and performs the following checks:

- Obtains detailed status information from Data Guard broker for each database.

- Verifies that the redo transport is functioning between the primary and standby databases.

- Verifies the proper configuration of the standby redo log files.

- Verifies that the Data Guard broker properties are consistent with underlying database properties.

- Verifies the status of the Agents for all databases.

- Verifies that the preferred credentials are set for all databases.

When the verification process ends, the following fixes are performed automatically:

- Resolves inconsistencies between broker and database properties.

- Creates standby redo log files for any databases when needed.

> **✎ Note:**
>
> Since verify_config is depracated, use dg_verify_config instead.

Format

```
emcli verify_config
        -primary_target_name="<primary target name>"
        -primary_target_type="oracle_database|rac_database"
        [-primary_db_creds_name="<primary database credential name>"]
        [-primary_host_creds_name="<primary database host credential name>"]
        [-reset_inconsistent_props="broker|database"]
        [-create_srls]
        [-verify_only]
```

Options

• primary_target_name

  Primary database Enterprise Manager target name. Can be either a single-instance database or a cluster database.

• primary_target_typ

Primary database target type. Specify oracle_database for single instance, or rac_database for cluster.

- primary_db_creds_name

    Primary database named credential for a user with SYSDBA or SYSDG role. Default is to use preferred credential.

- primary_host_creds_name

    Primary database host named credential for an operating system user who can access the primary database Oracle Home. Default is to use preferred credential.

- reset_inconsistent_props

    Resets the inconsistent properties to the broker or the database values. Allowed values:

    - broker

    - database

    Default is broker.

- create_srls

    Creates standby redo log files for any database that either have none or do not have the proper number and/or sizes. The files will be created as Oracle-managed files (OMF) for any databases that are configured with OMF.

- verify_only

    Runs a verification. Neither resolves automatically inconsistent properties nor creates standby redo log files.

Examples

Example 1

The following command verifies the health and resolves inconsistent properties of the cluster primary database named "database".

```
emcli verify_config
      -primary_target_name="database"
      -primary_target_type="oracle_database"
```

Example 2

The following command only verifies the health of the primary single-instance database named "primary".

```
emcli verify_config
      -primary_target_name="primary"
      -primary_target_type="rac_database"
      -verify_only
```

verify_swlib

Verifies and reports the state of the Software Library.

Format

```
emcli verify_swlib
         [-report_type="storage|entity|uploadjobs|all"]
```

```
        [-verbose]
```

[ ] indicates that the parameter is optional.

Options

- **report_type**

   Type of report to be generated, which can be one of the following:

   – **storage** — Reports accessibility of upload storage locations.

   – **entity** — Reports sanity of entities w.r.t associated files in upload storage locations

   – **uploadjobs** — Reports active jobs uploading files to storage.

   – **all** — For all reports.

- **verbose**

   Gives detailed output with the Software Library status.

Example

This example generates the storage, upload jobs, and entities verification reports.

```
emcli verify_swlib
      -report_type="all" -verbose

Logged in user: SYSMAN

Software Library Storage Verification Report: Start


OMS Shared File System storage is configured.
    ..........................................................
    [OMS Shared File System Storage Location : EM_SWLIB]

    Location 'EM_SWLIB' is accessible.

                 - Disk Usage Stats : EM_SWLIB -

    Free Disk Space : 456.05 GB
    Total Disk Space : 492.03 GB
    Disk Space taken up by Deleted Entities : 0 Bytes

                 - Credential Details : EM_SWLIB -

    Associated Credential Type : System Credential
    Credential Name : SWLIB15
    Credential Owner : <SYSTEM>
    Credential User : oracle
    Privileged : false
    ..........................................................
Software Library Storage Verification Report: End
Software Library Active Upload Jobs Report: Start
    ..........................................................
    No active upload jobs were found.
    ..........................................................
Software Library Active Upload Jobs Report: End

Continued in the next comment...
mmuthukr: Software Library Entity Verification Report: Start
OMS Shared File System storage is configured.
```

```
        ...............................................................
        [OMS Shared File System Storage Location : EM_SWLIB]
              --- List of entities with file issues ---
                      - User Owned Entities -

        Entity : /Database Provisioning CVU Downloads/ASM Disk Discovery/
linux_ppc64/ASM Disk Discovery Utility (0.1)
        File : kfodprereq_227.zip
        Storage Path : /EM12c_shared_library/EM_SWLIB/
        File Size (bytes) : NA
        File Status : Pending
        Issue : File pending upload.
        Modified Date : 2018-07-01 00:00:08.0
        Owner : SYSMAN


        ...... Summary ............................................

        - Found 1 user-owned entity revision(s) with missing files.

        - Found 1 entity revision(s) with mismatched file size between repository
and storage.
        ...............................................................

Software Library Entity Verification Report: End
```

verify_updates

Checks for archives that are missing in the software library, and prints steps to
download them and re-import them to the software library.

Format

```
emcli verify_updates
```

Options

None.

version

Lists EM CLI verb versions or the EM CLI client version.

Format

```
emcli version
    [-verb_name=<verb_name_filter>]
    [-exact_match]
    [-noheader]
    [-script | -format=
            [name:"pretty|script|csv"];
            [column_separator:"column_sep_string"];
            [row_separator:"row_sep_string"];
    ]

[ ]  indicates that the parameter is optional
```

Options

• **verb_name**

Verb name filter. Selects matching EM CLI verb names. When you specify this , an output table shows the version for each verb whose name matches <verb_name_filter>. The EM CLI client version is displayed when you do not specify this **option**.

Verb filters use regular expression pattern matching unless you specify -exact_match. A zero-length filter matches everything.

> **Note:**
>
> For Unix csh, use single quotes around a filter value containing '$' .

- **exact_match**

  Uses exact matching for filters.

- **noheader**

  Displays tabular information without column headers.

- **script**

  This is equivalent to `-format="name:script"`.

- **format**

  Format specification (default is `-format="name:pretty"`).

  - `format="name:pretty"` prints the output table in a readable format not intended to be parsed by scripts.

  - `format="name:script"` sets the default column separator to a tab and the default row separator to a newline. The column and row separator strings can be specified to change these defaults.

  - `format="name:csv"` sets the column separator to a comma and the row separator to a newline.

  - `format=`column_separator:"column_sep_string" column-separates the verb output by <column_sep_string>. Rows are separated by the newline character.

  - row_separator:"row_sep_string" row-separates the verb output by <row_sep_string>. Rows are separated by the tab character.

Output Columns

Verb, Version (when -verb_name is specified)

Examples

Example 1

This example shows the version for all verbs with names that contain a substring matching "elp" or with names that begin with "ver" or "lo", contains "i", and ends with "n:"

```
emcli version -verb_name="elp|^(ver|lo).*i.*n$"
```

Example 2

This example shows the version for all verbs with names that exactly match the string "setup." Alternatively, you could use the -exact_match .

```
emcli version -verb_name="^setup$"
```

ORACLE®

view_redundancy_group

Shows the present configuration of the redundancy group.

Format

```
emcli view_redundancy_group       -redundancyGroupName="redGrpName"
```

Parameters

- **redundancyGroupName**

  You must specify a single redundancy group name. The target name should be the same as present in the repository, and it should be of target type= "generic_redundancy_group".

Examples

This example shows the details for the 'redGrp1' Redundancy Group.

```
emcli view_redundancy_group -redundancyGroupName='redGrp1'
```

6
Error Code Reference

This chapter documents errors and associated codes returned by EM CLI. You can use EM CLI return codes to manage the control flow in a workflow/scripting environment. EM CLI return codes for Verb errors are positive integers. A Verb returns either 0 (successful execution) or an error number.

The following sections provide reference tables for these types of errors:

- EM CLI infrastructure
- OMS connection
- File-fed option
- Built-in verb

EM CLI Infrastructure Errors

Any execution of the EM CLI client could result in the following errors.

Table 6-1   Infrastructure Errors

Error Code	Description
242	A Verb has encountered a problem with a dependency specific to the implementation of the Verb (INSIDE of its abstraction barrier) unrelated to the Verb's semantics.
248	Configuration files are corrupt or inaccessible.
253	The command name is not recognized.
254	Internal system error.

OMS Connection Errors

Verbs that execute at the OMS return these error codes as indicated in the listing for each applicable verb.

Table 6-2   OMS Connection Errors

Error Code	Description
243	License has not been accepted by the current user.
249	Cannot connect to the OMS.
250	Wrong credentials for sign in to the OMS.

File-fed Option Errors

Verbs that allow for file-fed options (rather than options where the values are explicitly defined on the command line) can return the following error codes.

Table 6-3   File-Fed Option Errors

Error Code	Description
244	Cannot find an option value file.
245	Cannot read in an option value file.
246	An option value file is too big.

Built-in Verb Errors

The following error codes are returned by each verb (not including EM CLI infrastructure errors that apply to all verbs).

Table 6-4   Built-In Verb Errors

Verb	Error Code
add_beacon	0—Beacon added successfully.
	129— Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	173—Beacon does not exist.
	201—Beacon is already in the monitoring beacons list.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
add_group_to_mpa	2—I/O error occurred while writing to the MPA file.
	3—The specified MP already exists in the MPA.
	4—The group name is empty or not specified.
	223—The supplied options are syntactically incorrect.
add_mp_to_mpa	1—File does not exist, is unreadable, or an I/O error occurred.
	2—I/O error occurred while writing to the MPA file.
	3—The specified MP already exists in the MPA.
	4—The target-type definition file cannot be parsed.
	5—The MPA filename is not between 1 and 255 characters.
	6 —A file of a particular file type is required for another file.
	223—The supplied options are syntactically incorrect.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
add_target	1—The supplied target type does not exist. Unable to retrieve target metadata from the specified host's Management Agent.
	2—Host does not exist.
	3—Agent does not exist.
	4—Group does not exist.
	5—No monitoring credentials set found for target in the repository.
	6—Target instance already exists in the repository.
	7—The supplied target properties are incomplete.
	8—One or more of the supplied target properties are invalid.
	15—Target deletion in progress.
	20—Unable to connect to the specified host's Agent.
	21—Unable to save the target instance to the specified host's Agent.
	22—Cannot add more than one Agent target for a single Agent URL.
	23—Unable to add an instance of an Agent target without a URL.
	219—Insufficient privileges to add the target to the group.
	223—Unable to parse command line correctly. Invalid argument value.
	File-Fed Option Errors—The errors associated with file-fed options.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
apply_privilege_delegation_setting	0—Setting successfully applied.2—Setting does not exist.3—All or some of the targets are invalid.129—Syntax error. The displayed message indicates which argument is syntactically incorrect.
apply_template_tests	1—Error processing input XML file.4—Insufficient privileges for apply template.6—Target does not exist.7—Incompatible template and target types during apply.8—Test(s) specified for overwriteExisting do not exist in the template.9—Key test(s) specified as disabled for apply.10—Stepgroup contains a step that does not exist in the file.11—Some text property in file does not conform to valid syntax.12—Some text property contains variable but variable value is missing.13—Some transaction property/threshold/collection setting does not conform to required restrictions.50—Generic error.
argfile	Possible return error codes consist of the following list plus all of the errors returned by the Verb specified in the command line file for execution.
	244—The file does not exist.
	245—There is a problem reading in the file or it does not exist.
	246—The file ends inside a quoted token.
	247—The argfile options are specified incorrectly.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
assign_test_to_target	0—Test assigned to target type successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	190—Test or target type invalid.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
change_service_system_assoc	0—Service system changed successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	171—System <system> does not exist.
	172—Key component does not exist.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
clear_credential	1—Target type does not exist.
	2—Target does not exist.
	3—Credential set does not exist.
	4—Insufficient privileges.
	5—Credential column does not exist.
create_aggregate_service	1—Target does not exist.
	2—Target exists.
create_blackout	1—Blackout X already exists.
	2—Only Super Administrators are allowed to add a new reason (use get_ blackout_reasons).
	3—Agent targets cannot be directly blacked out.
	217—The blackout end_time cannot be in the past.
	The dates specified will never cause this blackout to take effect.
	The difference between the end_time and the start_time must be equal to the duration.
	The difference between the repeat interval and the duration must be at least X minutes.
	The duration must be -1 (for indefinite blackouts) or positive.
	The duration must be at least X minutes.
	219—Current user does not have OPERATOR privilege over all blackout targets.
	220—Target X does not exist.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
create_group	1—Group X already exists.
	2—Cannot add target X to typed group of base type Y.
	218—Group X is currently in the process of being deleted.
	219—Current user does not have privilege X over all member targets.
	220—Member target X does not exist.
	223—Unable to parse command line correctly.
	Invalid argument value.
	Group type is invalid.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
create_pluggable_database	2– Invalid -sourcePDBArchiveLocation (Fully qualified archive location of the source PDB)
	3–Caused due to one of the following conditions: • Specified source PDB is not open • Unable to identify CDB for source PDB • Source PDB name is not provided for PDB clone operation • Source CDB credentials not provided
	4–Location of new PDB datafiles (-newPDBFileLocation) not provided.
	5–PDB File Set location not provided (-sourcePDBMetadataFile and -sourcePDBDataBackup)
	6–Internal error (user must check the logs for more details)
	7–Invalid host credentials specified
	8–Invalid CDB (source and test)/ PDB Admin credentials specified
	9–Invalid privileged host credentials specified
	10–Internal error (user must check the logs for more details)
	11–Internal error (user must check the logs for more details)
	12–Internal error (user must check the logs for more details)
	13–Invalid count of PDBs specified
	14–Corresponds to errors that occur while validating Pluggable Database (PDB) Name. The new PDB name should not: • be same as the destination Container Database (CDB) name • have invalid characters (the PDB name should be at least 1 character and maximum 30 characters, start with an alphabetical character and must include only alphanumeric characters and the '_' character) • match with any existing PDBs in the destination CDB
	15–PDB Admin credentials not provided (-sourceType=DEFAULT")
	16–Corresponds to the following errors when cloning a PDB using profile: • Invalid profile URN (-profileURN) specified • The input -sourcePDBTempStagingLocation is not specified
	17–Corresponds to the following errors: • The -sourcePDBMetadataFile option not specified • Multiple pluggable databases cannot be created if -moveDatafiles option is specified • The -excludeStandbys option is applicable only if -moveDatafiles option is specified
	18–Source CDB not using OMF. -useOMF option specified when the source CDB does not use Oracle Managed Files (OMF)
	19–-sameAsSource option specified for multiple Pluggable Database creation

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
	21–Corresponds to the following errors: • The option -unpluggedPDBType is mandatory if UNPLUGGED_PDB is from file system • Invalid value specified for -unpluggedPDBType (Allowed values: ARCHIVE | RMAN | XML)
	22–The -mountPointPrefix option not specified (Applicable for PDB Snap Clone | PDB Clone using Profile)
	23–The -writableSpace option not specified (Applicable for PDB Snap Clone | PDB Clone using Profile)
	24–The -privHostCreds option not specified (Applicable for PDB Snap Clone | PDB Clone using Profile)
	25–Corresponds to the following errors: • The version of the source PDB does not match with the target CDB version • Cannot clone the specified PDB using Snap Clone since it is not enabled for Snap Clone
	26–The -srcCDBHostCreds option not specified
	27–Issue with Profile details: • The -profileName option not specified • The -profileLocation option not specified • The -profileURN option not specified • Given profile name already exists in Software Library • Invalid profile URN • Given profile location is not writable

> **Note:**
>
> Error codes 1 and 20 are not used.

Verb	Error Code
create_privilege_delegation_setting	0—Setting successfully created. 129—Syntax error. The displayed message indicates which argument is syntactically incorrect.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
create_red_group	0—Redundancy Group "<red_group_name>" created successfully.
	1—Redundancy Group "<red_group_name>" of target type <red_group_type> already exists.
	2—Cannot add target "<member_target_type>" to typed group of base type "<red_group_type>".
	3—Time Zone Region <timezone_region> does not exist.
	4—Redundancy Group Type "<red_group_type>" is invalid.
	218—Redundancy Group "<red_group_name>:<red_group_type>" is currently in the process of being deleted.
	220—Target "<member_target_name>:<member_target_type>" does not exist.
	223—Redundancy Group name "<red_group_name>" is not valid. It may contain only alphanumeric characters, multi-byte characters, a space, "-", "_", ".", ":", and have a maximum length of 256 characters.
	223—User name "<owner>" is not valid. It must begin with an alphabetic character, contain only alphanumeric characters, underscores (\"_\"), or periods (\".\"), and have a maximum length of 256 characters.
	223—Invalid value for parameter "add_targets": "<add_targets>". Reason: "<add_targets>" is not a name-value pair.
	223—Member Targets not of same type.
	223—"<generic_redundancy_group>" does not support member of type "<member_target_type>" .
create_role	1—Role by same name already exists.
	2—User with same name as role already exists.
	4—Privilege is invalid or nonexistent.
	5—Target specified in one of the privileges is invalid.
	6—The Super Administrator privilege cannot be granted to a role.
	7—Role does not exist.
	8—Group specified in one of the privileges is invalid.
	9—Job in privilege is invalid or nonexistent.
	10—Creating a role that you are assigning to the new role.
	11—The specified user does not exist.
	219—User is unauthorized to perform this action.
	223—Unable to parse command line correctly.
	Invalid argument value.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
create_service	0—Web application created successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	130—Missing key components.
	151—Test validation failed.
	171—System <system> does not exist.
	172—Key component does not exist.173—Beacon does not exist.
	181—No key tests defined.
	182—No key beacons defined.
	200—Service <target_name> already exists.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
create_system	0—System "<system_name:system_type>" created successfully.
	110—System "<system_name:system_type>" already exists.
	120—Member target "<member_target_name>:<member_target_type>" does not exist.
	122—Type "<system_type>" is not a valid System type.
	123—Time Zone Region "<timezone_region>" does not exist.
	130—Type meta version "<type_meta_ver>" is invalid.
	223—System name "<system_name>" is not valid. It must begin with an alphabetic char, contain only alphanumeric chars or any of "- _.:", and have a maximum length of 256 chars.
	223—Type meta version "<type_meta_ver>" is invalid. It must contain only numeric and "." characters, and have a maximum length of 8 chars.
	223—Timezone_region cannot be null or blank.
	223—Invalid value for parameter "add_members": "<add_members>". Reason: "<add_members>" is not a name-value pair.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
create_user	1—Target specified in one of the privileges is invalid.
	2—Group specified in one of the privileges is invalid.
	3—Job specified in one of the privileges is invalid.
	4—One of the specified privileges is invalid.
	5—Such user already exists.
	6—One or more roles to be granted to the new user does not exist.
	7—A role with the same name as the new user already exists.
	218—A delete is pending against this user until all blackouts and jobs submitted by this user are stopped.
	219—User has insufficient privileges to perform this operation.
	223—Unable to parse command line correctly:
	Invalid argument value.
	User name is somehow invalid.
	Supplied password does not have the proper format. Example: Password left empty.
	File-Fed Option Errors—The errors associated with file-fed options.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
delete_blackout	1—Blackout X created by user Y does not exist.
	2—Cannot delete a blackout that has not ended or was not stopped.
	219—You (X) do not have the SUPER_USER privilege needed to stop, delete, or modify blackout Y created by user Z.
	Only the blackout owner can stop, delete, or modify the blackout.
	Current user does not have OPERATOR privilege over all blackout targets.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
delete_group	1—Group X does not exist.
	218—Group X is currently in the process of being deleted.
	219—Current user does not have sufficient privileges to perform this action.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
delete_job	1—Specified job is invalid or non-existent.
	219—User has insufficient privileges to perform this operation.
	218—Some executions are not stopped when delete happens.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
delete_metric_promotion	0—SUCCESS
	223—SYNTAX_ERRNUM: Input is malformed.
	255—VERB_FAILED_ERRNUM: Back-end validation fails.
delete_pluggable_database	0–Setting successfully deleted.
	1–Deployment procedure has not been submitted because of validation errors.
delete_privilege_delegation_settings	0—Setting successfully deleted.2—All or some of the names are invalid.129—Syntax error. The displayed message indicates which argument is syntactically incorrect.
delete_role	1—Role does not exist.
	219—User is unauthorized to perform this action.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
delete_system	0—System "<system_name:system_type>" deleted successfully.
	121—System "<system_name:system_type>" does not exist.
	122—Type "<system_type>" is not a valid System type.
	219—Current user does not have sufficient privileges to perform this action.
	223—System name "<system_name>" is not valid. It must begin with an alphabetic character, contain only alphanumeric characters or any of "- _.:", and have a maximum length of 256 chars.
delete_target	15—Target deletion in progress.
	219—Insufficient privileges to delete specified target.
	220—Target does not exist.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
delete_test	0—Test deleted successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	174—Test does not exist.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.

ORACLE®

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
delete_user	1—Cannot delete the repository owner.
	2—Specified user does not exist.
	3—Cannot delete the current user.
	218—A delete is pending against this user until all blackouts and jobs submitted by this user are stopped.
	219—User has insufficient privileges to perform this operation.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
disable_audit	223—Syntax Error.
disable_test	0—Test disabled successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	174—Test does not exist
	203—Test already disabled.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
enable_audit	223—Syntax Error.
enable_test	0—Test enabled successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	174—Test does not exist
	202—Test already enabled.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
execute_hostcmd	0—Command execution succeeded for all targets.
	2—Command execution failed for one or more targets. Detailed errors will be displayed for each failed target.
	3—Invalid or unknown targets in the targets list.
	4—Preferred credentials are missing for one or more targets.
	5—Invalid credential set name.
	223—Unable to parse the command line properly.
execute_sql	0—Command execution succeeded for all targets.
	2—Command execution failed for one or more targets. Detailed errors will be displayed for each failed target.
	3—Invalid or unknown targets in the targets list.
	4—Preferred credentials are missing for one or more targets.
	5—Invalid credential set name.
	223—Unable to parse the command line properly.
export_template	223—Unable to parse command line correctly, or an exception was thrown during SQL handling.
	245—There is a problem writing to the file.

Table 6-4 (Cont.) Built-In Verb Errors

Verb	Error Code
extract_template_tests	2—Error serializing XML output.3—Insufficient privileges for extract template.5—Template does not exist in repository.50—Generic error.
get_aggregate_service_info	1—Target does not exist.
	2—Target exists.
get_aggregate_service_members	1—Target does not exist.
	2—Target exists.
get_blackout_details	1—Blackout X created by user Y does not exist.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
get_blackout_reasons	OMS Connection Errors—The errors associated with connecting to the executing OMS.
get_blackout_targets	1—Host X does not exist.223—Unable to parse command line correctly.220—Target X does not exist.
get_blackouts	1—Host X does not exist.
	220—Target X does not exist.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
get_group_members	1—Group X does not exist.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
get_groups	Other than the confirmation message, the get_groups verb only generates syntax errors. The SQL invoked by get_groups does not throw any exception.
	0—All groups (TargetName, TargetType) in the repository are displayed.
	223—Syntax Error: Argument -script cannot be specified with a value.
	223—Syntax Error: -format argument "name" value must match one of these strings: "script|pretty|csv".
	223—Syntax Error: Invalid value for parameter "format": "name:<format_name>;column_separator=<column_separator _char>". Reason: "column_separator=column_separator_char" is not a name-value pair.
	223—Syntax Error: -format argument contains an unrecognized key name <key_name>
get_instance_status	0—Success.
	223—Syntax Error.
get_jobs	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
get_procedures	0—Success.
	223—Syntax Error.
get_system_members	121—System "<system_name:system_type>" does not exist.
get_targets	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
grant_privs	2—User does not exist.
	3—Invalid privilege.
	4—Invalid target privilege.
	5—Invalid globally unique identifier (GUID).
	6—One or more targets are not groups.
	7—Specified job does not exist.
	8—Privilege grant failed.
grant_roles	2—User does not exist.
	7—Role does not exist.
help	1—There is no help available.
	223—Unable to parse the command line correctly.
import_template	21—Occurs if one of the templates has an OMS version specified in it that does not match the version of the OMS you are importing it into, and there are no other errors.
	22—Occurs if one of the template files cannot be parsed, and there are no other errors.
	99—More than one of the templates to be imported had errors during processing.
	223—Unable to parse command line correctly, or an exception was thrown during SQL handling.
	245—There is a problem reading in the file, or it does not exist.
list	0—Success.
	1—When list service fails to process the request.
login	0—Verb success exit value.
	1—Cannot establish an OMS connection storage area, or a corrupt area already exists.
	2—A connection with the OMS cannot be established.
	3—The login with the credentials provided failed at the OMS.
	4— The Enterprise Manager license was not accepted by the current user.
	5—The user is already logged in Enterprise Manager.
	223—Command syntax error Verb exit value.
	241—Custom attribute error handling.
	255—Error code for browser-related errors.
	248—Error:Either set EMCLI_CERT_LOC or EMCLI_TRUSTALL using set_client_property.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
logout	0—Verb success exit value.
	1—Cannot establish an OMS connection storage area, or a corrupt area already exists.
	2—A connection with the OMS cannot be established.
	3—The login with the credentials provided failed at the OMS.
	4— The Enterprise Manager license was not accepted by the current user.
	249—OMS connection error verb exit value.
	255—Error code for browser-related errors.
modify_aggregate_service	1—Target does not exist.
	2—Target exists.
modify_group	1—Group X does not exist.
	2—Cannot add target X to typed group of base type Y.
	3—Group X contains itself as a sub-group at some level.
	219—Current user does not have sufficient privileges to perform this action:
	Current user does not have privilege X over all member targets. Current user does not have sufficient privileges on target X to add it to the group.
	220—Target X does not exist.
	223—Unable to parse command line correctly. Group type is invalid.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
modify_red_group	0—Redundancy Group ""<red_group_name>" modified successfully.
	1—Redundancy Group ""<red_group_name>:<red_group_type>" does not exist.
	2—Cannot add target "<member_target_type>" to typed group of base type "<red_group_type>".
	4—Redundancy Group Type "<red_group_type>" is invalid.
	218—Redundancy Group "<red_group_name>:<red_group_type>" is currently in the process of being deleted.
	220—Target "<member_target_name>:<member_target_type>" does not exist.
	223—Redundancy Group name "<red_group_name>" is not valid. It may contain only alphanumeric characters, multi-byte characters, a space, "-", "_", ".", ":", and have a maximum length of 256 characters.
	223—User name "<owner>" is not valid. It must begin with an alphabetic character, contain only alphanumeric characters, underscores (\"_\"), or periods (\".\"), and have a maximum length of 256 characters.
	223—Invalid value for parameter "add_targets": "<add_targets>". Reason: "<add_targets>" is not a name-value pair.
	223—Member Targets not of same type.
	223—"Generic redundancy group" does not support member of type "<member_target_type>" .
modify_role	4—Privilege is invalid or nonexistent.
	5—Target specified in one of the privileges is invalid.
	6—The Super Administrator privilege cannot be granted to a role.
	7—Role does not exist.
	8—Group specified in one of the privileges is invalid.
	9—Job in privilege is invalid or nonexistent.
	10—Cannot have a circular chain of role grants.
	11—The specified user does not exist.
	219—User is unauthorized to perform this action.
	223—Unable to parse command line correctly. Invalid argument value.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
modify_system	0—System "<system_name:system_type>" modified successfully.
	101—System <system_name:system_type> contains itself as a sub-system at some level.
	120—Member target "<member_target_name>:<member_target_type>" does not exist.
	121—System "<system_name:system_type>" does not exist.
	122—Type "<system_type>" is not a valid System type.
	219—Current user does not have sufficient privileges on target <member_target_name> to add it to the system.
	219—Current user does not have sufficient privileges to perform this action.
	223—Invalid value for parameter "add_members": "<add_members>". Reason: "<add_members>" is not a name-value pair.
modify_target	8—One or more of the supplied target properties are invalid.
	15—Target deletion in progress.
	219—Insufficient privileges to modify target.
	220—Target does not exist.
	223—Unable to parse command line correctly.
	File-Fed Option Errors—The errors associated with file-fed options.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
modify_user	1—Target specified in one of the privileges is invalid.
	2—Group specified in one of the privileges is invalid.
	3—Job specified in one of the privileges is invalid.
	4—One of the specified privileges is invalid.
	5—Specified user does not exist.
	6—One or more roles to be granted to the new user does not exist.
	218—A delete is pending against this user until all blackouts and jobs submitted by this user are stopped.
	219—User has insufficient privileges to perform this operation.
	223—Unable to parse command line correctly: Invalid argument value or user name is somehow invalid.
	File-Fed Option Errors—The errors associated with file-fed options.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

ORACLE®

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
provision	1—An Internal error occurred. Could not get an Instance of the Assignment Manager. Exception occurred when getting URN from path.
	2—Could not provision. Exception occurred either in getting editable ProvisioningAssigment object, or during call to Initiate Provisioning.
	3—Could not get one or more URNs. Returned if any of imageUrn, bootServerUrn, stageServerUrn, networkProfileUrn, targetUrn retrieved is null.
	4—Could not create assignment state. Failed to create an AssignmentState object.
	5—Could not set assignment properties. Failed to set the assignment properties in the assignment state object.
	Since this verb uses the FileArgRemoteVerb, the following errors are also possible:
	• This Verb posts Verb.SYNTAX_ERRNUM if a specified option/file mapping on the command line is not properly formatted.
	• This Verb posts Verb.LOGIN_SYSTEM_ERRNUM if it cannot sign in to the OMS.
	• This Verb posts Verb.OMS_CONNECTION_SYSTEM_ERRNUM if it cannot connect to the OMS.
	• This Verb posts Verb.CONFIGURATION_SYSTEM_ERRNUM if the configuration files are corrupt or inaccessible.
	• This Verb posts Verb.MISSING_FILE_SYSTEM_ERRNUM if it cannot find an option value file.
	• This Verb posts Verb.FILE_READ_SYSTEM_ERRNUM if it cannot read in an option value file.
	• This Verb posts Verb.FILE_SYNTAX_SYSTEM_ERRNUM.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
relocate_targets	0—Moved all targets from Source Agent to Destination Agent.
	1—Target relocation has failed. The following errors are possible:
	• SQL exception when relocating targets : <Database-specific error message>.
	• Communication exception when relocating targets: < communication exception message >.
	• Verb usage error:
	`` emcli relocate_targets -src_agent=<source agent target name> - dest_agent=<dest agent target name> {-target_name=<name of the target to be relocated> - target_type=<type of the target to be relocated>}	{-input_file=dupTargets:<complete path to file>} {-force=yes}; " ```
	• Errors relocating targets from Source Agent to Destination Agent:
	< error message >
	< error message >
	• Exception in parsing targets from the command line argument <message>.
remove_beacon	0—Beacon removed successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	173—Beacon does not exist.
	225—Beacon not in monitoring beacons list.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
remove_service_system_assoc	0—System removed from service successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	180—System does not exist.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
retry_job	1—Cannot restart job of a non-restartable type.
	2—Specified job execution does not exist or has not failed.
	3—The specified job execution has already been restarted and failed on restart.
	219—User has insufficient privileges to perform this operation.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
revoke_roles	2—User does not exist.
	7—Role does not exist.
revoke_privs	2—User does not exist.
	3—Invalid privilege.
	4—One or more targets are invalid.
	5—Invalid globally unique identifier (GUID) privilege.
	6—One or more targets are not groups.
	7—Specified job does not exist.
	8—Privilege grant failed.
set_availability	0—Availability set successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	180—No system defined.
	181—No key tests defined.
	182—No key beacons defined.
	230—Insufficient privileges.
	231—Availability not changed.
	255—Back-end error. Verb failed.
set_credential	1—Target type does not exist.
	2—Target (of given target type) does not exist.
	3—Credential set does not exist.
	4—Insufficient privileges.
	5—Credential column does not exist.
	6—Credential column number mismatch.
set_key_beacons_tests	0—Key beacons and tests set successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	135—Must specify at least one key beacon and test.
	170—Service does not exist.
	173—Beacon does not exist.
	175—Beacon not in list of monitoring beacons.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
set_metric_promotion	0—SUCCESS
	223—SYNTAX_ERRNUM: Input is malformed.
	255—VERB_FAILED_ERRNUM: Back-end validation fails.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
set_properties	0—Properties set successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	132—Invalid property.
	133—Invalid property value.
	170—Service does not exist.
	173—Beacon does not exist.
	175—Beacon not in list of monitoring beacons.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
set_target_property_value	0—Success.
	6—Invalid target property.
	7—Invalid target type.
	8—Invalid target.
	9—Error saving target property.
	219—Privilege error.
	223—Syntax error.
setup	1—The Verb cannot establish a configuration area, or a corrupt area already exists.
	2—A connection with the OMS cannot be established.
	3—The login with the provided credentials fails at the OMS.
	4—The supplied "url" option is malformed or is not http/https.
	5—The configuration directory is not local as determined by the user in non-trustall HTTPS mode.
	6—The Verb cannot collect the user password safely.
	7—License is not been accepted by the user.
	223—Unable to parse command line correctly.
stop_blackout	1—Blackout X created by user Y does not exist.
	2—The blackout has already ended or stopped.
	3—Agent-side blackouts cannot be edited or stopped.
	218—The start of the blackout is currently being processed.
	The blackout is already pending stop.
	The last set of edits to the blackout have not yet been committed.
	219—You (X) do not have the Super Administrator privilege needed to stop, delete, or modify blackout Y created by user Z.
	Only the blackout owner can stop, delete, or modify the blackout.
	Current user does not have OPERATOR privilege over all blackout targets.
	223—Unable to parse command line correctly.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
stop_job	1—Specified job is invalid or non-existent.219—User has insufficient privileges to perform this operation.223—Unable to parse command line correctly.OMS Connection Errors—The errors associated with connecting to the executing OMS.
submit_job	1—Supplied job type is invalid or non-existent.
	2—Job with the same name already exists.
	3—One or more specified targets are invalid.
	4—Missing job parameter.
	5—Invalid job parameters, possibly including the security parameters such as "pwd".
	217—Specified job schedule is invalid.
	219—User has insufficient privileges to perform this operation.
	223—Unable to parse command line correctly.
	Invalid argument value.
	File-Fed Option Errors—The errors associated with file-fed options.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
submit_procedure	0— Success
	223— Syntax Error
subscribeto_rule	1—Rule with name X and owner Y does not exist.2—EM user X does not exist.3—EM user X has no email addresses set up (see console tab Preferences->General).4—Outgoing Mail (SMTP) Server not set up (see console tab Setup->Notification Methods).219—You (X) do not have the SUPER_USER or MANAGE_ANY_ USER privilege needed to add email addresses for user Y.You (X) do not have the SUPER_USER or MANAGE_ANY_ USER privilege needed to subscribe Y to the rule owned by Z.223—Unable to parse command line correctly.
	Invalid argument value.
	OMS Connection Errors—The errors associated with connecting to the executing OMS.
sync	1—The Verb cannot establish a configuration area or a corrupt area already exists.
	2—A connection with the OMS cannot be established.
	3—The login with the provided credentials fails at the OMS.
	4—The license has not been accepted by the current user.
	223—Unable to parse the command line correctly.

Table 6-4   (Cont.) Built-In Verb Errors

Verb	Error Code
sync_beacon	0—Beacon synced successfully.
	129—Syntax Error. The displayed message indicates which argument is syntactically incorrect.
	170—Service does not exist.
	173—Beacon does not exist.
	175—Beacon not in list of monitoring beacons.
	230—Insufficient privileges.
	255—Back-end error. Verb failed.
update_audit_settings	223—Syntax error, which could be an invalid directory name or invalid audit settings.
update_db_password	1—Invalid target.
	2—Invalid key value parameter.
	3—Invalid old password.
	4—Invalid privilege.
	223—Syntax error.
update_host_password	1—Invalid target.
	2—Invalid key value parameter.
	3—Invalid old password.
	4—Invalid privilege.
	223—Syntax error.
update_password	4—Target (of given target type) does not exist.
	5—Credential type does not exist for given target.
	6—Key value (that is, user name) does not exist.
	7—Non-operator cannot change credentials.
	8—Wrong value for old password.
	9—Old and new passwords match.
	10—No such non_key_column name.

ORACLE®

A

Sample Scripts

This appendix provides the sample scripts that were discussed in Using EM CLI in a format that you can use to copy the desired lines into your Jython code.

Sample Scripts

Example A-1   Script That Retrieves All Targets and Prints Their Names

```
#emcli_get_targets.py

#Import all emcli verbs to current program
from emcli import *

def print_target_details(target):
    '''
    print the target name and target type given a target tuple.
    '''
    print target['Target Name'] + ' ' + target['Target Type']

#Set the OMS URL to connect to
set_client_property('EMCLI_OMS_URL','https://host1.example.com:1234/em')
#Accept all the certificates
set_client_property('EMCLI_TRUSTALL','true')

#Login to the OMS
login(username='adminuser')

#Invoke get_targets and loop over the targets array
targets_array = get_targets().out()['data']
for target in targets_array:
        #Call print_target_details function to print the target details
    print_target_details(target)
```

Example A-2   Script that Incorporates Functions in the get_targets Verb

```
#emcli_introspect_response.py

#Import all emcli verbs to current program
from emcli import *

#Set the OMS URL to connect to
set_client_property('EMCLI_OMS_URL','https://host1.example.com:1234/em')
#Accept all the certificates
set_client_property('EMCLI_TRUSTALL','true')

#Login to the OMS
login(username='sysman')

res = get_targets()

print 'Number of targets:'+str(len(res.out()['data']))
print 'Errors           :'+res.error()
```

```
print 'Exit code         :'+str(res.exit_code())
print 'IsJson            :'+str(res.isJson())
```

Example A-3   Script that Incorporates Custom SQL with the list() Function

```
#emcli_json_processing.py
#Import all EM CLI verbs to current program
from emcli import *
def format(str):
   '''
   Given a string argument returns it back or returns
   a blank string if it is of None type
   '''
   if str is None:
     return ""
   return str

def get_targets_with_props(p_prop_name, p_prop_val):
'''
Returns targets with given property name and its value. Uses list verb.
'''
l_sql = "select target_name, target_type, property_value " \
        "from mgmt$target_properties " \
        "where property_name = '" + p_prop_name + "' " + "    " \
        "and property_value like '" + p_prop_val + "'"
   obj =
list(sql=l_sql)




   return obj
#Set the OMS URL to connect to
set_client_property('EMCLI_OMS_URL','https://host1.example.com:1234/em')
#Accept all the certificates
set_client_property('EMCLI_TRUSTALL','true')
#Log in to the OMS
login(username='sysman')
#Find all the targets that have Version property set to release 12
l_targets = get_targets_with_props('Version', '12%')
for target in l_targets.out()['data']:
     tn =  target['TARGET_NAME']
     tt =  target['TARGET_TYPE']
     pv =  target['PROPERTY_VALUE']
     print "Name "+tn + " Type =" + tt + " value=" + pv
```

Example A-4   Script that Incorporates Exception Handling

```
#emcli_error_exception_handling.py

#import all emcli verbs to current program
from emcli import *
#import the verbexecutionerror
```

```
from emcli.exception import VerbExecutionError

#Set the OMS URL to connect to
set_client_property('EMCLI_OMS_URL','https://host1.example.com:1234/em')
#Accept all the certificates
set_client_property('EMCLI_TRUSTALL','true')

#Login to the OMS
login(username='sysman')

#Create a group
res = create_group(name='Jan_Doe_Group')

print res.out()

#Try to create the same group again
try:
  #This will trigger an exception as the group exist already
  create_group(name='Jan_Doe_Group')
except VerbExecutionError , e:
   print e.error()
   print 'Exit code:'+str(e.exit_code())
```

Example A-5   LifeCyclePropertyChange.py

```
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS. ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A) THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE OF,
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.
from emcli import *

search_list = ['PROPERTY_NAME=\'DBVersion\'','TARGET_TYPE=
\'oracle_database\'','PROPERTY_VALUE LIKE \'11.2%\'']

if len(sys.argv) == 2:

      print login(username=sys.argv[0])
      l_prop_val_to_set = sys.argv[1]
      l_targets = list(resource="TargetProperties", search=search_list,
columns="TARGET_NAME,TARGET_TYPE,PROPERTY_NAME")
      for target in l_targets.out()['data']:
            t_pn =  'LifeCycle Status'
            print "INFO: Setting Property name " + t_pn + " to value " +
l_prop_val_to_set
            print set_target_property_value(property_records=target['TARGET_NAME']
+":"+target['TARGET_TYPE']+":"+t_pn+":"+l_prop_val_to_set)
 else: ]
```

```
          print "\n ERROR: Property value argument is missing"
          print "\n INFO: Format to run this file is filename.py <username>
<Database Target LifeCycle Status Property Value>"
```

Example A-6   dbPasswordChange.py

```python
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED
ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS.
ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY,
FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A)
THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON
AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE
SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE
OF,
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.

from emcli  import *
from emcli.exception import VerbExecutionError
import sys
import time

def check_job_status(job):
  count=0
  while (count < 10):
   count = count + 1
   obj = emcli.get_jobs(job_id=job)
   #print obj.out()
   for entry in obj.out()['data']:
       l_status = entry['Status ID']
       l_exec_id = entry['Execution ID']
       #print entry['Status ID']
       if (l_status == '5'):
           print "Job completed successfully"
           count=100
       elif (l_status == '4'):
           l_resp = get_job_execution_detail(execution=l_exec_id,
showOutput=True, xml=True)
           print "Job failed, error details "
           print "Output " + str(l_resp.out())
           count=100
       else:
           time.sleep(2)

def update_db_pwd_for_target(p_target_name, p_target_type, p_old_password,
p_new_password):
```

```
      l_target_name = p_target_name
      l_target_type = p_target_type
      print "Changing the password for member : name = " + l_target_name + " type = " +
l_target_type
      try :
          l_resp = update_db_password (target_name=l_target_name,
                                       target_type = l_target_type,
                                       change_at_target="yes",
                                       user_name="dbsnmp",
                                       old_password=p_old_password,
                                       new_password=p_new_password,
                                       retype_new_password=p_new_password)
          l_job_submitted = l_resp.out()['JobId']
          check_job_status(l_job_submitted)
      except emcli.exception.VerbExecutionError, e:
          print "ERROR : Change Password failed for name = " + l_target_name + " type = "
+ l_target_type
          print "ERROR : " + e.error()

def update_db_pwd_for_group(p_group, p_old_password, p_new_password):
    print "Changing the password for group - " + p_group + " from " + p_old_password +
" to " + p_new_password
    members = get_group_members(name=p_group).out()['data']
    for member in members:
        l_target_name = member['Target Name']
        l_target_type = member['Target Type']
        update_db_pwd_for_target(l_target_name, l_target_type, p_old_password,
p_new_password)


#Set the OMS URL to connect to
set_client_property('EMCLI_OMS_URL','https://myoms.com/em')
#Accept all the certificates
set_client_property('EMCLI_TRUSTALL','true')

login(username=sys.argv[0])


l_grp_name = 'maurGroup'

l_group_members = ['db1:oracle_database','db2:oracle_database','db3:rac_database']



res = create_group(name = l_grp_name, add_targets = l_group_members)

print "Listing members for group " + l_grp_name

for member in get_group_members(name=l_grp_name).out()['data']:
    print member


y_n_input = raw_input('Now lets change the password for all the members in this
group(y/n)')
if y_n_input != 'y':
 exit(0)

l_tgt_username = "dbsnmp"
l_old_password = "secret1"
l_new_password = "secret2"
```

```
update_db_pwd_for_group(l_grp_name, l_old_password, l_new_password)
```

Example A-7   promote_discovered_dbs.py

```
#Disclaimer
#EXCEPT WHERE EXPRESSLY PROVIDED OTHERWISE, THE SITE, AND ALL CONTENT PROVIDED
ON
#OR THROUGH THE SITE, ARE PROVIDED ON AN "AS IS" AND "AS AVAILABLE" BASIS.
ORACLE
#EXPRESSLY DISCLAIMS ALL WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED,
#INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY,
FITNESS
#FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT WITH RESPECT TO THE SITE AND ALL
#CONTENT PROVIDED ON OR THROUGH THE SITE. ORACLE MAKES NO WARRANTY THAT: (A)
THE
#SITE OR CONTENT WILL MEET YOUR REQUIREMENTS; (B) THE SITE WILL BE AVAILABLE ON
AN
#UNINTERRUPTED, TIMELY, SECURE,OR ERROR-FREE BASIS; (C) THE RESULTS THAT MAY BE
#OBTAINED FROM THE USE OF THE SITE OR ANY CONTENT PROVIDED ON OR THROUGH THE
SITE
#WILL BE ACCURATE OR RELIABLE; OR (D) THE QUALITY OF ANY CONTENT PURCHASED OR
#OBTAINED BY YOU ON OR THROUGH THE SITE WILL MEET YOUR EXPECTATIONS.
#ANY CONTENT ACCESSED, DOWNLOADED OR OTHERWISE OBTAINED ON OR THROUGH THE USE OF
#THE SITE IS USED AT YOUR OWN DISCRETION AND RISK. ORACLE SHALL HAVE NO
#RESPONSIBILITY FOR ANY DAMAGE TO YOUR COMPUTER SYSTEM OR LOSS OF DATA THAT
#RESULTS FROM THE DOWNLOAD OR USE OF CONTENT.
#ORACLE RESERVES THE RIGHT TO MAKE CHANGES OR UPDATES TO, AND MONITOR THE USE
OF,
#THE SITE AND CONTENT PROVIDED ON OR THROUGH THE SITE AT ANY TIME WITHOUT NOTICE.

 from emcli.exception import VerbExecutionError
 import sys

 alltargets=False
 targetparms=0
 uname=''
 pword=''
 url=''
 monitor_pw=''

 def helpUsage():
    print 'Usage: promote_discovered_dbs.py [-help]'
    print '[-all] Add all discovered Single Instance DBs'
    print '[-targets <target1:target2:...] Add only targets listed'
    sys.exit()

 for i in range(len(sys.argv)):
    if sys.argv[i] in ("-help"):
            helpUsage()
    elif sys.argv[i] in ("-targets"):
            if i+1 < len(sys.argv):
                    targetparms = sys.argv[i+1]
    else:
                    print 'Usage: promote_discovered_dbs.py [-help]'
                    print '[-all] Add all discovered Single Instance DBs'
                    print '[-targets <target1:target2:...] Add only targets
listed'
                    sys.exit()
    elif sys.argv[i] in ("-url"):
            if i+1 < len(sys.argv):
```

```
                                        url = sys.argv[i+1]
        elif sys.argv[i] in ("-username"):
                if i+1 < len(sys.argv):
                        uname = sys.argv[i+1]
        elif sys.argv[i] in ("-password"):
                if i+1 < len(sys.argv):
                        pword = sys.argv[i+1]
        elif sys.argv[i] in ("-monitor_pw"):
                if i+1 < len(sys.argv):
                        monitor_pw = sys.argv[i+1]
        elif sys.argv[i] in ("-all"):
                        alltargets = True

 # Make sure user did not specify target list and all targets.
 if alltargets<>0 and targetparms <>0:
     print 'Cannot specify target list and all switch'
     print 'Usage: promote_discovered_dbs.py -url <EM URL> -username <username> -
password <password> -monitor_pw <password>'
     print '[-all] Add all discovered SI Databses'
     print '[-targets <target1:target2:...] Add only list targets'
     print '[-help]'
     sys.exit()

 if len(uname)==0 or len(pword)==0 or len(url)==0:
         print 'Missing required arguments (-url, -username, -password)'
         print 'Usage: promote_discovered_dbs.py -url <EM URL> -username <username> -
password <password> -monitor_pw <password>'
         print '[-all] Add all discovered SI Databses'
         print '[-targets <target1:target2:...] Add only list targets'
         print '[-help]'
         sys.exit()

 # Set Connection properties and logon
 set_client_property('EMCLI_OMS_URL',url)
 set_client_property('EMCLI_TRUSTALL','true')
 login(username=uname,password=pword)

 cred_str = "UserName:dbsnmp;password:" + monitor_pw + ";Role:Normal"

 if targetparms <> 0:
     targetparms = targetparms.replace(":",":oracle_database;")+":oracle_database"
     target_array =
get_targets(unmanaged=True,properties=True,targets=targetparms).out()['data']
 elif alltargets:
     target_array =
get_targets(targets="oracle_database",unmanaged=True,properties=True ).out()['data']
 else:
     print 'Missing required arguments (-targets or -all)'
     helpUsage()

 if len(target_array) > 0:
     for target in target_array:
             print 'Adding target ' + target['Target Name'] + '...',

             for host in str.split(target['Host Info'],";"):
                 if host.split(":")[0] == "host":]
                     print host.split(":")[1]
             try:
                 res1 = add_target(type='oracle_database',name=target['Target
Name'],host=host.split(":")[1], credentials=cred_str,properties=target['Properties'])
                 print 'Succeeded'
```

```
            except VerbExecutionError, e:
                print 'Failed'
                print e.error()
                print 'Exit code:'+str(e.exit_code())
 else:
     print 'INFO: There are no targets to be promoted. Please verify the targets
in Enterprise Manager webpages.'
```