Mobile Labs

deviceConnect™ CLI User Guide

Contents

1 Requirements

1.1 Software Requirements

Various components will require different software. The deviceConnect UI is a browser based application and browser based deviceViewer. These can run on many platforms and browsers. Mobile Labs Trust is a software download that only runs on Windows. Below are the software requirements needed to use deviceConnect.

1.1.1 deviceConnect Management

Browsers	Chrome, Edge, IE 8 or later, Firefox, Safari
OS X	10.8 or later
Windows	Windows 7 or later

2 deviceConnect Command Line Interface (CLI)

deviceConnect comes with a Command Line Interface (CLI). The CLI can be used to create scripts that automate most of the deviceConnect management functions that can be performed manually using the web user interface. The CLI can be scripted in Continuous Integration systems to deploy apps to the testing devices. It can even be used in automated tests to retain a device, install and launch an application, and then connect to the device to begin automation without user interaction.

2.1 Installation

To install the CLI tool, simply download it from deviceConnect. In deviceConnect, in the user dropdown at the upper right corner, click on the Download deviceConnect CLI option. This will download a zip file. You can extract the zip file anywhere you wish, however, we recommend putting it in the following locations:

- Windows: C:\Program Files (x86)\Mobile Labs\Cli
- Mac: \usr\local\deviceconnect\cli

2.2 Connecting to deviceConnect

Note
If using Mac OS X, use "mono" as the precursor to each call.

The CLI executable requires that you enter in the authentication information into each call that is used. There are a few options and formats for authenticating with the CLI.

```
MobileLabs.DeviceConnect.Cli.exe <host> <username> <password> [options]
```

```
MobileLabs.DeviceConnect.Cli.exe <host> <username> <api key> [options]
```

```
MobileLabs.DeviceConnect.Cli.exe <username>:<password>@<host> [options]
```

```
MobileLabs.DeviceConnect.Cli.exe <username>:<api key>@<host> [options]
```

The host is the IP address or DNS entry for the deviceConnect server, username and password/API token are of a user in deviceConnect. This information must be present in every call unless specified in the configuration file.

Tip
The API token allows a user to authenticate without storing a password in plain text. The token can be found on your user details page by expanding the user menu and selecting manage your account.

2.2.1 CLI Configuration File

The CLI configuration file can be used to store the connection information to deviceConnect so that it does not need to be specified for every call.

On Windows systems, the following file will need to be created: %HOMEPATH%\dccli.ini. On OS X systems, the following file will need to be created: ~/.dcclirc. The contents of the file will be:

```
[connection]
server=host
user=username
token=api token
```

Tip
To override the values stored in the config file use the -host -username -password (or -api) options. See the connection options below.

2.3 Running A Command

Running a command is as simple as using the connection items before and adding the options for the commands you want to use. There are options for the various features of deviceConnect. To see a list of features, use the "-help" option or do not specify an option.

Note
All values that contain spaces must be marked within quotations.

```
> MobileLabs.DeviceConnect.Cli.exe
> MobileLabs.DeviceConnect.Cli.exe -help
```

For example, to get a list of devices use the following command:

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    devicelist
```

The result of the command would be:

```
Online:
  Blue iPhone (iPhone 5c IOS 7.0.4 FXXXXXXXXXXG Enabled)
  iPad 2 iOS 7 (iPad 2 IOS 7.0.2 DXXXXXXXXXXW Enabled)
  iPhone 3GS (iPhone 3GS IOS 6.1.3 QXXXXXXXXXXP Enabled)
  iPhone 5s (iPhone 5s IOS 7.0.3 FXXXXXXXXXXV Enabled)
  ML Nexus 4 (Nexus 4 Android 4.3 01XXXXXXXXX1c Enabled)
  ML_Nexus 5 (Nexus 5 Android 4.4 03XXXXXXXXXce Enabled) [Tester Name]
  New Nexus 7 (Nexus 7 Android 4.3 01XXXXXXXXX11 Enabled)
  Old Nexus 7 (Nexus 7 Android 4.3 01XXXXXXXXX02 Enabled)
```

2.4 Options

There are options for the various versions of deviceConnect features. To see a list of features, use the "-help" option or do not specify an option. Remember that the host, username, and password are required for each call.

2.4.1 CLI options

The options for the CLI tool will allow the user to understand how to use the CLI tool or change the way it behaves.

-help

Displays a list of commands and options, this is also called without providing any option.

```
> MobileLabs.DeviceConnect.Cli.exe
> MobileLabs.DeviceConnect.Cli.exe -help
```

-continue

Allows the command to continue processing options after printing an error message from a previous command that has failed.

The below example will continue to upload the application if the -deleteapp option failed because the application is not present in deviceConnect.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    continue -deleteapp MobileLabsApp -upload "C:\applications\MobileLabsApp.ipa"
```

-os [android,ios]

Adds an iOS or Android OS filter to application and device options.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -os  ↩
    ios -dl -al

Online:
 Blue iPhone (iPhone 5c IOS 7.0.4 FXXXXXXXXXXG Enabled)
 iPad 2 iOS 7 (iPad 2 IOS 7.0.2 DXXXXXXXXXXW Enabled)
 iPhone 3GS (iPhone 3GS IOS 6.1.3 QXXXXXXXXXXP Enabled)
 iPhone 5s (iPhone 5s IOS 7.0.3 FXXXXXXXXXXV Enabled)
IOS:
 deviceControl 4.5/43 (min 6.0, uploaded 05/05/15 1:23:45 PM) 3eacd9ed-5d98-4855-9a06 ↩
    -f6b56fedffbf
 f5ba9eb5bb8d
 PhoneLookup 4.0/1 (min 6.0, uploaded 05/05/15 1:23:45 PM) c0d26509-a00a-413e-8906-30 ↩
    d263993209
 Trust Browser 1.3.4/8 (min 5.0, uploaded 05/05/15 1:23:45 PM) c3d87f35-4c66-43da-85 ↩
    a1-75cd7f3ad510
```

-ping

This option will test to see if the host is reachable. If the host is contacted a message will be printed, otherwise it will exit with a nonzero code.

2.4.2 Connection options

-host

Allows a user to change the deviceConnect host in a string of options. The following shows how to get two device lists from two different deviceConnect units. The first server is displayed followed by the following server.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -dl - ↩
    host 192.168.1.100 -dl
```

Note
username and password must be the same for both servers.

-username

Allows a user to use a different deviceConnect username than provided in the connection parameters. The following shows how to get a device list from another deviceConnect unit.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -host ↩
    192.168.1.100 -username your.email@example.com -password yourPassword -dl
```

-password

Allows a user to use a different deviceConnect password than provided in the connection parameters. The following shows how to get a device list from another deviceConnect unit.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -host ↩
    192.168.1.100 -username your.email@example.com -password yourPassword -dl
```

-api

Allows a user to use a different deviceConnect API token than provided in the connection parameters. The following shows how to get a device list from another deviceConnect unit.

```
MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com ceba3df1-b9d1
-415f-a704-be29a3838899 -host 192.168.1.100 -username your.email@example.com -api  ↩
    apiToken -dl
```

2.4.3 Application Options

-al or -applist [android,ios]

This will return a list of all applications and the builds uploaded to deviceConnect. Passing "android" or "ios" will filter the list to those particular operating systems. The default is without parameters. The output is a list of the applications along with version, build, unique ID, and minimum operating system. The output is also grouped by operating system.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    applist

IOS:
 deviceControl 4.5/43 (min 6.0, uploaded 05/05/15 1:23:45 PM) 3eacd9ed-5d98-4855-9a06 ↩
    -f6b56fedffbf
 PhoneLookup 4.0/1 (min 6.0, uploaded 05/05/15 1:23:45 PM) c0d26509-a00a-413e-8906-30 ↩
    d263993209
 Trust Browser 1.3.4/8 (min 5.0, uploaded 05/05/15 1:23:45 PM) c3d87f35-4c66-43da-85 ↩
    a1-75cd7f3ad510
Android:
 deviceControl 1.0/1 (min 2.3.1, uploaded 05/05/15 1:23:45 PM) cce59724-1369-48d7 ↩
    -8197-5ba8e4943002
 Phone Lookup 1.0/1 (min 2.3.1, uploaded 05/05/15 1:23:45 PM) 0b6643c7-c318-4d05-ad88 ↩
    -c1c302b1a251
 Trust Browser 123/270 #2 of 2 (min 2.2, uploaded 05/05/15 1:23:45 PM) a6a191a5 ↩
    -1955-4051-8309-83d3655aa670
```

-deleteapp <application>

This will delete an application from deviceConnect by passing in the name of the application or the ID. If there are multiple uploads of the application with that name, you can use the unique ID of the application to delete the single instance. If a unique ID or version is not specified, then the latest upload will be deleted. Use the "-version" option to delete a particular version of the application. If more than one application is found using the version option then a message of "# applications start with that name." will appear and the application will not be deleted.

> **Note**
> When using the version option for deleteapp, the build number cannot be specified alone and must include the version number.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    deleteapp MobileLabsApp
```

-version <version>

This will allow the version to be specified for an application. This option must be placed before the other options that use an application. The information can be specified by the version, build, or upload # that can come from an application.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    version 1.0 -deleteapp MobileLabsApp
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    version "#2" -deleteapp MobileLabsApp
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    version "1.0/3#2" -deleteapp MobileLabsApp
```

-upload <filename>

This will upload an application to deviceConnect

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    upload "C:\applications\MobileLabsApp.ipa"
```

2.4.4 Device Options

-devicelist/-dl [online|offline|all],[android|ios],[available|inuse],[simple|delimited],[enabled,disabled][-format csv|json|list] [-o file]

This will list the devices that are in deviceConnect. The optional parameters filter the list by online or offline devices, Android or iOS devices, available displays devices that can be retained with reservation information, if any, and inuse will show a list of devices retained by another user. Using the simple parameter will output the list with only a device name per line. Using the delimited parameter will print out a tab separated list of all of the device details per line making it easy to export to a file and parse for scripting. The default is to display only online (connected) devices. The output displays: Name, Model, Platform, Operating System, Serial Number, and Username. Do not put spaces between the options. File formatting options are also accepted as CSV, JSON, or list by passing in the -format option. Using the -o option allows the device list to be saved to a file specified as the parameter.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword  ↩
    -devicelist

Online:
 Blue iPhone (iPhone 5c IOS 7.0.4 FXXXXXXXXXXG Enabled)
 iPad 2 iOS 7 (iPad 2 IOS 7.0.2 DXXXXXXXXXXW Enabled)
 iPhone 3GS (iPhone 3GS IOS 6.1.3 QXXXXXXXXXXP Enabled)
 iPhone 5s (iPhone 5s IOS 7.0.3 FXXXXXXXXXXV Enabled)
 ML Nexus 4 (Nexus 4 Android 4.3 01XXXXXXXXX1c Enabled)
 ML_Nexus 5 (Nexus 5 Android 4.4 03XXXXXXXXXce Enabled) [Test User]
 New Nexus 7 (Nexus 7 Android 4.3 01XXXXXXXXX11 Enabled)
 Old Nexus 7 (Nexus 7 Android 4.3 01XXXXXXXXX02 Enabled)
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    devicelist online,ios,simple

Blue iPhone
iPad 2 iOS 7
iPad First Gen
iPhone 3GS
iPhone 5s
```

-d or -device [name]

This will allow you to specify a device for items such as install, launch, retain, release, reboot, log, forward, set USB port mode. The name parameter can be the device name or serial number including prefixes. If the name of the device is not specified, then the first online device will be used. If more than one device has the same name, the first match against the database will be used. The CLI will search in the following manner: exact match by serial number, name, and then it'll try prefix matches

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -log

> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device Blue -log

> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "FXXXXXXXXXXG" -log
```

-install <application>

This will install an application to a particular device when used with the device command. The name or unique ID of the application can be used.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -install MobileLabsApp
```

-launch <application>

This will launch an application on a particular device when used with the device command. The name or unique ID of the application can be used. The application must be installed and the device must be retained before running this command.

Note
It is not recommended to use the -launch command if you are using -autoconnect, as both commands will launch the application.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -retain -launch MobileLabsApp
```

-uninstall <application>

This will uninstall an application from a particular device when used with the device command. The name or unique id of the application is required.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -uninstall MobileLabsApp
```

-preserve

This will flag to allow data to be preserved if an application is uninstalled with the uninstall command. This command will come before the uninstall command.

Note
This option is only available for Android devices.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "ML Nexus 4" -preserve -uninstall MobileLabsApp
```

-retain

This will retain the device and show as in use for the user passed in the connection command.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -retain
```

-release

This will release the device from the user and show as Available.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -release
```

-reboot

This will reboot the device if needed, or if the device is not responding.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -reboot
```

-renamedevice <new name>

This will rename a particular device giving it a new alias. The new name is required.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -renamedevice "Red iPhone"
```

-log

This will return the log for a particular device. You can save this to a file for reporting within a script or to send to a developer.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -log

> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -log > C:\DeviceLogs\BlueiPhoneLog.txt
```

-forward <local port> <remote port>

This will forward a local port to the device specified at the remote port. If only one port is specified then it will be used for both the local and remote ports. This is useful when using third party testing frameworks that will allow you to run a test against the real device giving you the benefits of device management with deviceConnect. The port parameter is required.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -forward 88 80
```

-enable <device name>

This option will enable a device bringing it into the pool of licensed devices. The Device name is required.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    enable "Blue iPhone"
```

-disable

This option will remove a device from the license pool freeing up a license for another device to be used.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -disable
```

-reset [-uninstallAll] [-reboot]

The -reset command will reset the state of a device and -reboot reboots a device. The reset will default to the system-wide preference unless specified by the -uninstallAll and -reboot options.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -reset -uninstallAll
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -reset -uninstallAll -reboot
```

2.4.5 Viewer Options

Note
only available on Windows machines in which deviceViewer is installed.

-connect <application>

This option when called will launch the Trust deviceViewer application, retain the device as well as launch the application on the device. The deviceViewer will be ready to accept scale and rotation and the user can then click the Connect button to connect. The parameter for application is required and the name or unique ID of the application can be used.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -retain -install "MobileLabsApp" -connect "MobileLabsApp"
```

-autoconnect <application>

This option when called will launch the Trust deviceViewer application, reserve the device as well as launch the application on the device. The deviceViewer will connect immediately without any user prompts. The parameter for application is required and the name or unique ID of the application can be used.

Note
It is not recommended to use the -launch command if you are using -autoconnect as both commands will launch the application.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -retain -install "MobileLabsApp" -autoconnect "MobileLabsApp ↩
    "
```

-r or -run <application>

This option is a shortcut to a few commands -retain, -install, and -autoconnect within the viewer. The parameter for application is required and the name or unique ID of the application can be used.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -r "MobileLabsApp"
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -run "MobileLabsApp"
```

-scale < % >

This option is used with the –run, -connect, and -autoconnect options to scale the viewer window of the device to a particular percentage. The values are entered in integers and must be placed before the –run, -connect, and -autoconnect options. For example, 25 for 25%. This cannot be used to scale the viewer window after the viewer is launched, use the viewer menu to scale the screen.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -scale 50 -r "MobileLabsApp"
```

-orientation

This option is used with the –run, -connect, and -autoconnect options to put the device in a particular orientation. The default is portrait orientation if no parameter is entered. The valid options are: Portrait, LandscapeLeft, LandscapeRight, UpsideDown and must be placed before the –run, -connect, and -autoconnect options. If an invalid option is used, then it will default back to portrait. This cannot be used to orient the application after the viewer is launched, use the viewer menu to change the orientation.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -orientation LandscapeLeft -r "MobileLabsApp"
```

-client [web,desktop,https]

This option specifies which client will be used to connect to the device. The default client is the desktop version on Windows, or the web client everywhere else. If the web argument is used, it will open within the default browser selected on the computer. The scale option is not used when launching with the web viewer. If https is used as the argument, then the link to the server is generated using HTTPS. The https option can only be used if the server has been set up to use HTTPS.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -orientation LandscapeLeft -client web -r "MobileLabsApp"
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -orientation LandscapeLeft -client desktop -r "MobileLabsApp ↩
    "
```

2.4.6 User Options

With the CLI, a user with the proper entitlements can interact with various user options within deviceConnect.

-user <id or email> [-details],[-usage],[-history],[-password]

This option specifies which deviceConnect user to return -details, -usage, -history and -password information for.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -user ↩
    my.user@example.com -usage
```

-details [-format csv|json|list][-o file]

This option provides user details as a screen output or a CSV, JSON, or list by passing in the -format option. Using the -o option allows the output to be saved as a specified -format file. By default, -details output is returned in a list format.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -user ↩
    my.user@example.com -details
```

The output of the command will be:

```
        Email:  my.user@example.com
        IsActive:       True
        FirstName:      My
        MiddleName:     M
        LastName:       User
        Notes:  Sample line note
        Organization:   Example
        Title:  Senior Software Engineer
        Location:       Atlanta Office
        Address1:       123 Test St
        Address2:
        City:   Atlanta
        Region: GA
        PostalCode:     30000
        Country:        US
        HomePhone:      404-555-5555
        MobilePhone:    404-555-6666
        OfficePhone:    404-555-7777
        FaxPhone:
        Password:
        PasswordHash:
        PasswordSalt:
```

```
        PasswordHashType:       0
        Roles:  "Administrator"
```

Other example commands:

To save user details as a CSV file:

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -user ↩
    my.user@example.com -details -format csv -o "C:\DeviceConnect\Users\details.csv"
```

json format will return two additional user details, "Id" and "CreatedDate":

```
MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -user  ↩
    admin -details -format json
```

"CreatedDate" will only display the date of newly created users, the return value for existing users will be "null."

-password <new password>

Supplying the -password option after the -user password will allow the password for the specified user to be changed. This will not change a password for a user account that uses LDAP authentication.

```
MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com password -user  ↩
    username -password newpassword
```

-userimport </path/to/csv/file.csv>

This option allows an entitled user to import users into the system with a CSV file. By using the user template, the users will be added within the system. The user password will be set to what text is contained in the Password field, otherwise the PasswordHash and PasswordSalt columns will be used.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    userimport "C:\DeviceConnect\Users\import.csv"
```

Tip
It is useful to use the "-userexport" option to get a template for the user import CSV file.

-userexport </path/to/csv/file.csv> [-format csv|json]

This option allows an entitled user to export deviceConnect users into a CSV or JSON file. The file will be created when saved. If the format option is not set, the default is CSV file.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    userexport "C:\DeviceConnect\Users\export.csv"
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    userexport "C:\DeviceConnect\Users\export.json" -format json
```

Note
A JSON file format cannot be used for user import.

2.4.7 Gateway Options

-gateways [-format csv|json|list] [-o file]

View the USB hub port status for the attached device gateways.

To save USB hub port status as a CSV file:

```
MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    gateways -format csv -o "C:\DeviceConnect\GatewayDetails\GatewayDetails.csv"
```

-usbhubportmode <data|charge|off> [-usbhub <usb hub serial>] [-usbhubport <usb hub port number>]

Option to change a device's port mode to data, charge or off. Can be changed by specifying device or by the USB hub serial number and port number.

To set port mode to data by device id:

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword -d "7 ↩
    e3a6b6b-ad5b-40e2-9fd7-0d18dd46832a" -usbhubportmode data
```

To set port mode to off by USB hub serial number and port number:

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usbhubportmode off -usbhub DJ008SQD -usbhubport 5
```

-usbhubreboot <usb hub serial number>

Will reboot the designated USB hub.

To reboot USB hub DJ008SQD:

```
MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usbhubreboot DJ008SQD
```

2.4.8 Time Options

-timezones

This option lists all of the timezones in Olsen location name (https://en.wikipedia.org/wiki/Tz_database). This can be used to find timezones to add to the usage and history reports.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    timezones
```

2.4.9 Report Options

History and Usage data can be retrieved via the CLI, allowing the data to be saved into a file or just viewed.

-history [arguments]

This option can be run by itself to return the whole history for every device. This option returns the data in a CSV file. When used in conjunction with the -device or -d options (which will come before using the -history option) then only the history for this device is returned. The arguments passed to history can narrow down the criteria of the history data or even specify a file to save the data to. All arguments are optional.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    history
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -history
```

-usage [arguments]

This option can be run by itself to return the whole usage data for every device. This option returns the data in a CSV file. When used in conjunction with the -device or -d options (which will come before using the -usage option) then only the usage for this device is returned. The arguments passed to usage can narrow down the criteria of the usage data or even specify a file to save the data to. All arguments are optional.

Note
The duration column is in a format of HH:MM:SS.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usage
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    device "Blue iPhone" -usage
```

Report Arguments:

-from <YYYY-MM-DD>

This argument is used to specify the date for the report to start from. This does not have to be used with the "-to" argument.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    history -from 2015-01-01
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usage -from 2015-01-01
```

-to <YYYY-MM-DD>

This argument specifies the date for the report to end on. This does not have to be used with the "-from" argument.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    history -from 2015-01-01 -to 2015-05-01
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usage -from 2015-01-01 -to 2015-05-01
```

-days <N>

This argument specifies the number of previous days the report should include.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    history -days 7
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usage -days 7
```

-timezone <locations>

This argument allows additional date and time columns to be displayed for a particular time zone. The argument specifies space delineated list of timezone locations for timestamps to appear in. "local" will reference the machine's timezone. Multiple locations require the values to be enclosed in quotes.

Note
When the report is CSV formatted, it will appear as additional columns. When the report is JSON formatted, it will appear inside a singular object.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    history -days 7 -timezone local
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    history -days 7 -timezone "America/New_York Asia/Calcutta local"
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usage -days 7 -timezone local
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usage -days 7 -timezone "America/New_York Asia/Calcutta local"
```

-format <csv|json>

This argument specifies the output format. Valid options are CSV or JSON. The options are case sensative and if no argument is used "CSV" is the default option.

-o <filename>

This argument specifies the output file (including path) to save the data to. The data is returned in a CSV format. Whithout this option, the default behavior is to output the data to the console.

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    history -days 7 -o "C:\DeviceConnect\History\Previous7Days.csv"
```

```
> MobileLabs.DeviceConnect.Cli.exe 192.168.1.50 my.email@example.com myPassword - ↩
    usage -days 7 -o "C:\DeviceConnect\Usage\Previous7Days.csv"
```

3 Legal Notices

Confidential and Proprietary

The information contained in this document is confidential and proprietary to Mobile Labs, LLC. No part of this guide may be distributed or disclosed in any form to any third party without written permission of Mobile Labs, LLC.

Disclaimers

The information contained herein is subject to change without notice.

Mobile Labs shall not be liable for technical or editorial errors or omissions contained herein.

Copyrights

Trademarks

Mobile Labs is a trademark of Mobile Labs, LLC.
deviceConnect is a trademark of Mobile Labs, LLC.
deviceBridge is a trademark of Mobile Labs, LLC.
Mobile Labs Trust is a trademark of Mobile Labs, LLC.

iOS is a trademark or registered trademark of Cisco in the U.S. and other countries and is used under license.

App Store is a service mark of Apple Inc.
Xcode is a registered trademark of Apple Inc.
iTunes is a registered trademark of Apple Inc.

Android is a trademark of Google Inc.

Amazon, Kindle, Fire and all related logos are trademarks of Amazon.com, Inc. or its affiliates.

Microsoft, Windows, Windows XP are U.S registered trademarks of Microsoft Corporation in the United States and other countries.

HP Unified Functional Testing is a registered trademark of Hewlett-Packard Development Company. HP and the names of any other HP products referenced herein are trademarks and/or service marks or registered trademarks and/or service marks of Hewlett-Packard Development Company, L.P., and/or its subsidiaries.