

# UCI® Command Line Interface Reference

**Unified Configuration Interface (UCI®)** is an API of OpenWrt® which is also a utility to intend and to centralize the whole configuration of a device running on OpenWrt®.

## Using UCI

1. Open an SSH terminal connection and connect to the device
2. Type the following command to execute UCI:

```
#uci
```

3. Then step ito the OpenWrt/UCI system configuration files directory, where all configuration items of the UCI are located in the `/etc/config/`

```
#cd /etc/config/
```

4. Then check the files here:

```
#ls
```

```
root@M2M-Pro4:/etc/config# ls -l
-rw-r--r--  1 root    root      947 Apr 20 13:08 dhcp
-rw-r--r--  1 root    root       62 Apr 20 12:05 dropbear
-rw-----  1 root    root      670 Apr 12 12:39 easycwmp
-rw-r--r--  1 root    root    2849 Apr 19 09:55 firewall
-rw-r--r--  1 root    root     484 Apr 12 12:39 fstab
-rw-r--r--  1 root    root     822 Jan  1 1970 luci
-rw-r--r--  1 root    root   3785 Apr 17 12:52 luci_statistics
-rw-r--r--  1 root    root     662 Apr 20 13:08 network
-rw-----  1 root    root     392 Apr 20 13:16 ntpclient
-rw-----  1 root    root      97 Mar  7 16:01 rpcd
-rw-----  1 root    root     809 Mar  7 16:01 ser2net
-rw-r--r--  1 root    root    1267 Apr 20 12:05 system
-rw-r--r--  1 root    root     860 Apr 20 12:06 ucitrack
-rw-----  1 root    root     752 Apr 12 12:39 uhttpd
-rw-r--r--  1 root    root     328 Apr 18 15:03 users
root@M2M-Pro4:/etc/config#
```

## Syntax of UCI

Before continuing to use the UCI, let's see its syntax:

```
#uci [options] commands arguments
```

*\*Using options is optional, it is not obligatory to use.*

## Commands

Command	Target	Description
<b>batch</b>	-	Executes a multi-line UCI script which is typically wrapped into a here document syntax
<b>export</b>	[<config>]	Exports the configuration in a machine readable format. It is used internally to evaluate configuration files as shell scripts
<b>import</b>	[<config>]	Imports configuration files in UCI syntax
<b>changes</b>	[<config>]	Lists staged changes to the given configuration file or if none given, all configuration files
<b>commit</b>	[<config>]	Writes changes of the given configuration file, or if none is given, all configuration files, to the filesystem. All "uci set", "uci add", "uci rename" and "uci delete" commands are staged into a temporary location until they are written to flash with the "uci commit" command. This is used exclusively for UCI commands and is not needed after editing configuration files with a text editor
<b>add</b>	<config> <section-type>	Adds an anonymous section of type <i>section-type</i> to the given configuration
<b>add_list</b>	<config>.<section>.<option>=<string>	Adds the given <i>string</i> to an existing list option
<b>del_list</b>	<config>.<section>.<option>=<string>	Removes the given <i>string</i> from an existing list option

<b>show</b>	[<config>[.<section>[.<option>]]]	Shows the given option, section or configuration in compressed notation. If no option is given, shows all configuration files
<b>get</b>	<config>.<section>[.<option>]	Gets the value of the given option or the type of the given section
<b>set</b>	<config>.<section>[.<option>]=<value>	Sets the value of the given option, or add a new section with the type set to the given <i>value</i>
<b>delete</b>	<config>[.<section>[[.<option>][=<id>]]]	Deletes the given section or option
<b>rename</b>	<config>.<section>[.<option>]=<name>	Renames the given option or section to the given name
<b>revert</b>	<config>[.<section>[.<option>]]	Reverts the given option, section or configuration file. Used to undo any changes performed with UCI and not yet committed with <i>uci commit</i>
<b>reorder</b>	<config>.<section>=<position>	Moves the specified section to the given <i>position</i> . Used for easier management purposes

## Options

```

-c <path> set the search path for config files (default: /etc/config)
-d <str>  set the delimiter for list values in uci show
-f <file> use <file> as input instead of stdin
-m       when importing, merge data into an existing package
-n       name unnamed sections on export (default)
-N       don't name unnamed sections
-p <path> add a search path for config change files
-P <path> add a search path for config change files and use as default
-q       quiet mode (don't print error messages)
-s       force strict mode (stop on parser errors, default)
-S       disable strict mode
-X       do not use extended syntax on 'show'

```

## Checking by UCI

The commands are listed, but for arguments you have to know the useable configuration elements.

1. Then the following command will list the config files:

```
#ls /etc/config
```

2. There you will find the following list, then you can use these configuration elements at arguments:

```
root@M2M-Pro4:~# ls /etc/config
dhcp          fstab          ntpclient     ucitrack
dropbear     luci           rpcd          uhttpd
easycwmp     luci_statistics ser2net       users
firewall     network       system
```

If you are new at using UCI, we offer to check the configuration of some services first. The **SHOW** command is for that.

1. For example choose the „**dhcp**“ from the list by UCI.
2. Then to check „**dhcp**“ arguments and settings, use the following command:

```
#uci show dhcp
```

3. Then the current dhcp settings will be listed

```
root@M2M-Pro4: ~
dhcp.@dnsmasq[0].nonwildcard='1'
dhcp.@dnsmasq[0].localservice='1'
dhcp.lan=dhcp
dhcp.lan.interface='lan'
dhcp.lan.start='100'
dhcp.lan.limit='150'
dhcp.lan.lease_time='12h'
dhcp.lan.dhcpv6='server'
dhcp.lan.ra='server'
dhcp.lan.ra_management='1'
dhcp.wan=dhcp
dhcp.wan.interface='wan'
dhcp.wan.ignore='1'
dhcp.odhcpd=odhcpd
dhcp.odhcpd.maindhcp='0'
dhcp.odhcpd.leasefile='/tmp/hosts/odhcpd'
dhcp.odhcpd.lease_trigger='/usr/sbin/odhcpd-update'
dhcp.odhcpd.loglevel='4'
dhcp.usblan=dhcp
dhcp.usblan.start='100'
dhcp.usblan.lease_time='12h'
dhcp.usblan.limit='150'
dhcp.usblan.interface='usblan'
root@M2M-Pro4:~#
```

Another example to check your TR-069 settings:

```
#uci show easycwmp
```

Then the current TR-069 arguments and settings will be listed

```
root@M2M-Pro4: ~
easycwmp.@local[0].username='easycwmp'
easycwmp.@local[0].logging_level='3'
easycwmp.@acs[0]=acs
easycwmp.@acs[0].url='http://192.168.1.110:8080/openacs/acs'
easycwmp.@acs[0].username='easycwmp'
easycwmp.@acs[0].periodic_enable='1'
easycwmp.@acs[0].periodic_interval='100'
easycwmp.@acs[0].periodic_time='0001-01-01T00:00:00Z'
easycwmp.@acs[0].ssl_cert='easycwmp.cert'
easycwmp.@device[0]=device
easycwmp.@device[0].oui='FFFFFF'
easycwmp.@device[0].serial_number='FFFFFF123456'
easycwmp.@device[0].manufacturer='OpenWrt
http://lede-project.org/'
easycwmp.@device[0].product_class='Generic'
easycwmp.@device[0].hardware_version='v0'
easycwmp.@device[0].software_version='r6395-6c19407'
firewall.@defaults[0]=defaults
firewall.@defaults[0].syn_flood='1'
firewall.@defaults[0].input='ACCEPT'
firewall.@defaults[0].output='ACCEPT'
firewall.@defaults[0].forward='REJECT'
firewall.@zone[0]=zone
firewall.@zone[0].name='lan'
```

To view a configuration file such as `/etc/config/network`:

```
#uci export network
```

When you checked the configuration of the service, then you will be able to get detailed information on that by the **GET** command. The structure of the **GET** command is the following:

```
# uci get <config>.<section>[.<option>]
```

## Configuration files

File	Description
<code>/etc/config/dhcp</code>	Stores Dnsmasq configuration and DHCP settings
<code>/etc/config/dropbear</code>	Stores List Of Blocked Addresses configuration settings
<code>/etc/config/easycwmp</code>	Stores EasyCwmp configuration settings
<code>/etc/config/firewall</code>	Stores Firewall rules configuration settings
<code>/etc/config/fstab</code>	Stores network Shares configuration settings
<code>/etc/config/luci</code>	Base LuCI config
<code>/etc/config/luci_statistics</code>	LuCI statistics
<code>/etc/config/network</code>	Stores Network (LAN, WAN, Mobile interface) interface configuration settings
<code>/etc/config/ntpclient</code>	Stores NTP configuration settings

<b>/etc/config/rpcd</b>	Stores Login settings
<b>/etc/config/ser2net</b>	RS485 serial proxy settings
<b>/etc/config/system</b>	Stores various system settings (e.g., modem settings, reset button settings, device's hostname, etc.)
<b>/etc/config/ucitrack</b>	Stores <i>init</i> script information
<b>/etc/config/uhttpd</b>	Stores RMS configuration settings
<b>/etc/config/uhttpd</b>	http daemon settings
<b>/etc/config/users</b>	User settings

### Setting and configuring by UCI for the modem

For example, to setup TR-069 certification file use the following commands, setup the ***url*** (with the ACS server and http(s) URL, and ***ssl\_cert*** (with path and filename of the certification file on the local device) entries.

Then you have to commit the changes to apply.

```
#uci set easycwmp.@acs[0].url='192.168.1.110:8080/openacs/acs'

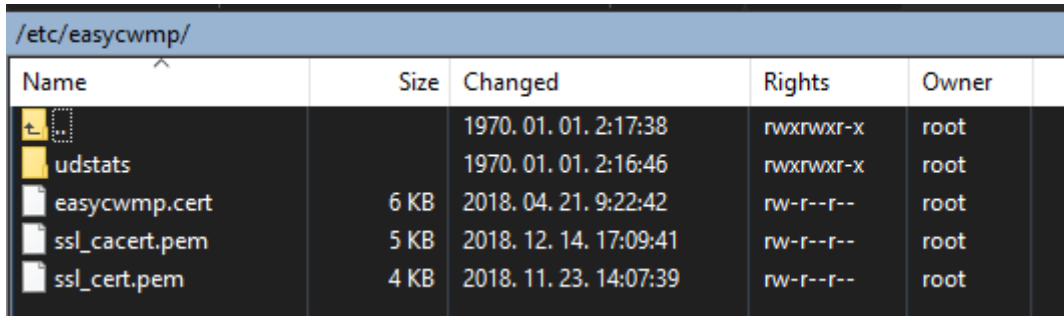
#uci set easycwmp.@acs[0].ssl_cert='/etc/easycwmp/easycwmp.cert'

#uci commit
```

### ***Important!***

*For using the certification file, you have to copy to the path you were given.*

Copy the .cert file to the **/etc/easycwmp/ directory** to the modem. You can use e.g. the **WinSCP** tool for that (**SCP** protocol, **port 22**, by defining an **account** and **password** for the connection).



Name	Size	Changed	Rights	Owner
udstats		1970. 01. 01. 2:17:38	rw-rw-r-x	root
easycwmp.cert	6 KB	2018. 04. 21. 9:22:42	rw-r--r--	root
ssl_cacert.pem	5 KB	2018. 12. 14. 17:09:41	rw-r--r--	root
ssl_cert.pem	4 KB	2018. 11. 23. 14:07:39	rw-r--r--	root