NAME

mmcli − Control and monitor the ModemManager

SYNOPSIS

mmcli [*OPTION*...]

DESCRIPTION

ModemManager is a DBus-powered Linux daemon which provides a unified high level API for communicating with (mobile broadband) modems. It acts as a standard RIL (Radio Interface Layer) and may be used by different connection managers, like NetworkManager. Thanks to the built-in plugin architecture, ModemManager talks to very different kinds of modems with very different kinds of ports. In addition to the standard AT serial ports, Qualcomm-based QCDM and QMI ports are also supported.

HELP OPTIONS

−h, −−help
Show summary of options by group.

−−help−all
Show all groups and options.

−−help−manager
Show manager specific options.

−−help−common
Show common options. These are used for defining the device an option operates on. For example, modems, bearers, SIMs, SMS', etc.

−−help−modem
Show modem specific options.

−−help−3gpp
Show 3GPP specific options.

−−help−cdma
Show CDMA specific options.

−−help−simple
Show simple options. These are useful for getting connected or disconnected and understanding the state of things as fast as possible without worrying so much about the details.

−−help−location
Show location or positioning specific options.

−−help−messaging
Show messaging specific options. See also **−−help−sms** which is related.

−−help−time
Show time specific options.

−−help−firmware
Show firmware specific options.

−−help−oma
Show OMA specific options.

−−help−sim
Show SIM card specific options.

−−help−bearer
Show bearer specific options.

−−help−sms
Show SMS specific options. See also **−−help−messaging** which is related.

MANAGER OPTIONS

−B, −−get−daemon−version
Retrieve the version of the currently running ModemManager daemon.

−G, −−set−logging=[ERR|WARN|INFO|DEBUG]
Set the logging level in ModemManager daemon. For debugging information you can supply **DEBUG**. Each value above **DEBUG** provides less detail. In most cases **ERR** (for displaying errors) are the important messages.

The default mode is **ERR**.

−L, −−list−modems
List available modems.

−M, −−monitor−modems
List available modems and monitor modems added or removed.

−S, −−scan-modems
Scan for any potential new modems. This is only useful when expecting pure RS232 modems, as they are not notified automatically by the kernel.

−I, −−inhibit−device=[UID]
Inhibit the specific device from being used by ModemManager. The **UID** that should be given is the value of the **Device** property exposed by a given modem (i.e. equal to the **ID_MM_PHYSDEV_UID** if one set, or otherwise equal to the full device sysfs path).

This command will not exit right away, as that would implicitly remove the inhibition. The user must make sure to stop the mmcli process hitting Ctrl+C in order to un-inhibit the device.

When a device is inhibited via this method, ModemManager will disable the modem (therefore stopping any ongoing connection) and will no longer use it until it is uninhibited.

−−report−kernel−event=['KEY1=VALUE1,KEY2=VALUE2,...']
Manually report kernel events, instead of relying on udev (e.g. if the daemon is running with **−−no−auto−scan** or if the system was built without udev support).

The supported **KEY**s are:

'action'
Action to report, one of 'add' or 'remove'. Required.

'subsystem'
Subsystem of the specific port being reported, e.g. 'tty' (for serial ports),

'name' Name of the port being reported, e.g. 'ttyACM0', 'wwan0' or 'cdc-wdm0'.

'uid' The specific UID of the device, equivalent to the **ID_MM_PHYSDEV_UID** udev tag. All ports reported with the same 'UID' value will be considered part of the same device, which may be useful for e.g. modems with multiple platform TTYs.

−−report−kernel−event−auto−scan
When built with udev support but the daemon is running with **−−no−auto−scan**, this method may be used to automatically report kernel events based on udev.

This command will not exit right away. The user must make sure to stop the mmcli process hitting Ctrl+C in order to stopping monitoring for new events.

COMMON OPTIONS

All options below take a **PATH** or **INDEX** argument. If no action is provided, the default information about the modem, bearer, etc. is shown instead.

The **PATH** and **INDEX** are created automatically when the modem is plugged in. They can be found using **mmcli −L**. This produces something like (for modems only):

```
Found 1 modems:
  /org/freedesktop/ModemManager1/Modem/4
```

In this case, the **INDEX** is **4** and the **PATH** is the entire string above.

However, for the bearers, SIMs and SMS cases, the **PATH** is slightly different. The *Modem* is replaced with the object name in use, like *Bearer*. For example:

```
/org/freedesktop/ModemManager1/Bearer/4
```

−m, −−modem=[PATH|INDEX]
Specify a modem.

−b, −−bearer=[PATH|INDEX]
Specify a bearer.

−i, −−sim=[PATH|INDEX]
Specify a SIM card.

−s, −−sms=[PATH|INDEX]
Specify an SMS.

MODEM OPTIONS

All of the modem options below make use of the **−−modem** or **−m** switch to specify the modem to act on.

Some operations require a **MODE**. **MODE** can be any combination of the modes actually supported by the modem. In the perfect case, the following are possible:

'2G' - 2G technologies, e.g. EDGE, CDMA1x
'3G' - 3G technologies, e.g. HSPA, EV-DO
'4G' - 4G technologies, e.g. LTE
'ANY' - for all supported modes.

−w, −−monitor−state
Monitor the state of a given modem.

−e, −−enable
Enable a given modem.

This powers the antenna, starts the automatic registration process and in general prepares the modem to be connected.

−d, −−disable
Disable a given modem.

This disconnects the existing connection(s) for the modem and puts it into a low power mode.

−r, −−reset
Resets the modem to the settings it had when it was power cycled.

−−factory−reset=CODE
Resets the modem to its original factory default settings.

The **CODE** provided is vendor specific. Without the correct vendor code, it's unlikely this operation will succeed. This is not a common user action.

−−command=COMMAND
Send an AT **COMMAND** to the given modem. For example, **COMMAND** could be 'AT+GMM' to probe for phone model information. This operation is only available when ModemManager is run in debug mode.

−−create−bearer=['KEY1=VALUE1,KEY2=VALUE2,...']
Create a new packet data bearer for a given modem. The **KEY**s and some **VALUE**s are listed below:

'apn' Access Point Name. Required in 3GPP.

'ip-type'
Addressing type. Given as a MMBearerIpFamily value (e.g. 'ipv4', 'ipv6', 'ipv4v6'). Optional in 3GPP and CDMA.

'allowed-auth'
Authentication method to use. Given as a MMBearerAllowedAuth value (e.g. 'none|pap|chap|mschap|mschapv2|eap'). Optional in 3GPP.

'user' User name (if any) required by the network. Optional in 3GPP.

'password'
Password (if any) required by the network. Optional in 3GPP.

'allow-roaming'
Flag to tell whether connection is allowed during roaming, given as a boolean value (i.e 'yes' or 'no'). Optional in 3GPP.

'rm-protocol'
Protocol of the Rm interface, given as a MMModemCdmaRmProtocol value (e.g. 'async', 'packet-relay', 'packet-network-ppp', 'packet-network-slip', 'stu-iii'). Optional in CDMA.

'number'
Telephone number to dial. Required in POTS.

−−delete−bearer=[PATH|INDEX]
Delete bearer from a given modem.

−−set−allowed−modes=[MODE1|MODE2|...]
Set allowed modes for a given modem. For possible modes, see the beginning of this section.

−−set−preferred−mode=MODE
Set the preferred **MODE** for the given modem. The **MODE** *MUST* be one of the allowed modes as set with the **−−set−allowed−modes** option. Possible **MODE** arguments are detailed at the beginning of this section.

−−set−current−bands=[BAND1|BAND2|...]
Set bands to be used for a given modem. These are frequency ranges the modem should use. There are quite a number of supported bands and listing them all here would be quite extensive. For details, see the MMModemBand documentation.

An example would be: 'egsm|dcs|pcs|g850' to select all the GSM frequency bands.

−−set−primary−sim−slot=[SLOT]
Request to switch the primary SIM slot.

The given **SLOT** must be a valid slot number in the [1,N] range, where N is the amount of SIM slots available in the system.

−−inhibit
Inhibit the specific modem from being used by ModemManager. This method is completely equivalent to **−−inhibit−device**, with the only difference being that in this case, the modem must be managed by the daemon at the time the inhibition is requested.

This command will not exit right away, as that would implicitly remove the inhibition. The user must make sure to stop the mmcli process hitting Ctrl+C in order to un-inhibit the device.

When a device is inhibited via this method, ModemManager will disable the modem (therefore stopping any ongoing connection) and will no longer use it until it is uninhibited.

3GPP OPTIONS

The 3rd Generation Partnership Project (3GPP) is a collaboration between groups of telecommunications associations. These options pertain to devices which support 3GPP.

Included are options to control USSD (Unstructured Supplementary Service Data) sessions.

All of the 3GPP options below make use of the **−−modem** or **−m** switch to specify the modem to act on.

−−3gpp−scan
Scan for available 3GPP networks.

−−3gpp−register−home
Request a given modem to register in its home network.

This registers with the default network(s) specified by the modem,

−−3gpp−register−in−operator=MCCMNC
Request a given modem to register on the network of the given **MCCMNC** (Mobile Country Code, Mobile Network Code) based operator. This code is used for GSM/LTE, CDMA, iDEN, TETRA and UMTS public land mobile networks and some satellite mobile networks. The ITU-T Recommendation E.212 defines mobile country codes.

−−3gpp−ussd−status
Request the status of *ANY* ongoing USSD session.

−−3gpp−ussd−initiate=COMMAND
Request the given modem to initiate a USSD session with **COMMAND**.

For example, **COMMAND** could be '*101#' to give your current pre-pay balance.

−−3gpp−ussd−respond=RESPONSE
When initiating an USSD session, a RESPONSE may be needed by a network-originated request. This option allows for that.

−−3gpp−ussd−cancel
Cancel an ongoing USSD session for a given modem.

−−3gpp−disable−facility−lock=FACILITY,CONTROL_KEY
Disable selected facility lock using provided control key.

'FACILITY'
One of the following types of lock:
'net-pers' - network personalization
'net-sub-pers' - network subset personalization
'provider-pers' - provider personalization
'corp-pers' - corporate personalization

'CONTROL_KEY'
Alphanumeric code to unlock facility.

CDMA OPTIONS

All CDMA (Code Division Multiple Access) options require the **--modem** or **-m** option.

--cdma-activate=CARRIER
Activate the given modem using OTA (Over the Air) settings. The **CARRIER** is a code provided by the network for the default settings they provide.

SIMPLE OPTIONS

All simple options must be used with **--modem** or **-m**.

--simple-connect=['KEY1=VALUE1,KEY2=VALUE2,...']
Run a full connection sequence using **KEY** / **VALUE** pairs. You can use the **--create-bearer** options, plus any of the following ones:

'pin' SIM-PIN unlock code.

'operator-id'
ETSI MCC-MNC of a network to force registration.

--simple-disconnect
Disconnect *ALL* connected bearers for a given modem.

LOCATION OPTIONS

These options detail how to discover your location using Global Positioning System (GPS) or directly from your mobile network infrastructure (either 3GPP or 3GPP2).

All location options must be used with **--modem** or **-m**.

--location-status
Show the current status for discovering our location.

--location-get
Show all location information available.

--location-enable-3gpp
Enable location discovery using the 3GPP network.

--location-disable-3gpp
Disable location discovery using the 3GPP network.

--location-enable-agps-msa
Enable A-GPS (MSA) support. This command does not implicitly start the GPS engine, it just specifies that A-GPS should be enabled when the engine is started. Therefore, the user should request enabling A-GPS before the raw or NMEA outputs are enabled with **--location-enable-gps-raw** or **--location-enable-gps-nmea**.

--location-disable-agps-msa
Disable A-GPS (MSA) support.

--location-enable-agps-msb
Enable A-GPS (MSB) support. This command does not implicitly start the GPS engine, it just specifies that A-GPS should be enabled when the engine is started. Therefore, the user should request enabling A-GPS before the raw or NMEA outputs are enabled with **--location-enable-gps-raw** or **--location-enable-gps-nmea**.

−−location−disable−agps-msb
Disable A-GPS (MSB) support.

−−location−enable−gps−nmea
Enable location discovery using GPS and reported with NMEA traces.

This command will start the GPS engine, if it isn't started already.

−−location−disable−gps−nmea
Disable location discovery using GPS and NMEA traces.

If the raw output is not enabled at the same time, the GPS engine will be stopped.

−−location−enable−gps−raw
Enable location discovery using GPS and reported with raw (i.e. longitude/latitude) values.

This command will start the GPS engine, if it isn't started already.

−−location−disable−gps−raw
Disable location discovery using GPS and raw values.

If the NMEA output is not enabled at the same time, the GPS engine will be stopped.

−−location−enable−cdma-bs
Enable location discovery using the 3GPP2 network.

−−location−disable−cdma-bs
Disable location discovery using the 3GPP2 network.

−−location−enable−gps−unmanaged
Enable location discovery using GPS but without taking control of the NMEA tty port. This allows other programs, e.g. gpsd, to use the NMEA tty once the GPS engine has been enabled.

−−location−disable−gps−unmanaged
Disable location discovery using GPS and unmanaged port.

−−location−set−gps−refresh−rate=SEC
Set the location refresh rate on the DBus interface to SEC seconds. If set to 0, the new location is published on the DBus interface as soon as ModemManager detects it.

−−location−set−supl−server=[IP:PORT] or −−location−set−supl−server=[FQDN:PORT]
Configure the location of the A−GPS SUPL server, either specifying the IP address (**IP:PORT**) or specifyng a fully qualified domain name (**[FQDN:PORT]**).

−−location−inject−assistance−data=[PATH]
Inject assistance data into the GNSS module, loaded from a local file at **PATH**. The assistance data should be in a format expected by the device, e.g. downloaded from the URLs exposed by the 'AssistanceDataServers' property.

−−location−set−enable−signal
Enable reporting location updates via DBus property signals. This is required if applications rely on listening to 'Location' property updates, instead of explicit queries with the policy-protected 'GetLocation' method.

This DBus property signal updates are by default disabled.

−−location−set−disable−signal
Disable reporting location updates via DBus property signals.

MESSAGING OPTIONS

All messaging options must be used with **−−modem** or **−m**.

−−messaging−status
Show the status of the messaging support.

−−messaging−list-sms
List SMS messages available on a given modem.

−−messaging−create-sms=['KEY1=VALUE1,...']
Create a new SMS on a given modem. **KEY**s can be any of the following:

'number'
Number to which the message is addressed.

'text' Message text, in UTF-8. When sending, if the text is larger than the limit of the technology or modem, the message will be broken into multiple parts or messages. Note that text and data are never given at the same time.

'smsc' Indicates the SMS service center number.

'validity'
Specifies when the SMS expires in the SMSC.

'class' 3GPP message class (0..3).

'delivery-report-request'
Specifies whether delivery report is requested when sending the SMS ('yes' or 'no')

'storage'
Specifies the storage where this message is kept. Storages may be 'sm', 'me', 'mt', 'sr', 'bm', 'ta'.

−−messaging−create−sms−with−data=PATH
Use **PATH** to a filename as the data to create a new SMS.

−−messaging−delete−sms=[PATH|INDEX]
Delete an SMS from a given modem.

TIME OPTIONS

All time operations require the **−−modem** or **−m** option.

−−time Display the current network time from the operator. This includes the timezone which is usually of importance.

VOICE OPTIONS

All voice operations require the **−−modem** or **−m** option.

−−voice−list−calls
List calls managed (initiated, received, ongoing) on a given modem.

−−voice−create-call=['KEY1=VALUE1,...']
Create a new outgoing call on a given modem. **KEY**s can be any of the following:

'number'
Number to call.

−−voice−delete−call=[PATH|INDEX]
Delete a call from a given modem.

FIRMWARE OPTIONS

All firmware options require the **−−modem** or **−m** option.

−−firmware−status
Show firmware update specific details and properties.

−−firmware−list
List all the firmware images installed on a given modem.

−−firmware−select=ID
Select a firmware image from those installed on a given modem. A list of available firmware images can be seen using the **−−firmware−list** option.

The **ID** provided is a *UNIQUE* identifier for the firmware.

SIGNAL OPTIONS

All signal options require the **−−modem** or **−m** option.

−−signal−setup=[Rate]
Setup extended signal quality information retrieval at the specified rate (in seconds).

By default this is disabled (rate set to 0).

−−signal−get
Retrieve the last extended signal quality information loaded.

OMA OPTIONS

All OMA options require the **−−modem** or **−m** option.

−−oma−status
Show the status of the OMA device management subsystem.

−−oma−start−client−initiated−session=[SESSION TYPE]
Request to start a client initiated session.

The given session type must be one of:
'client−initiated−device−configure'
'client−initiated−prl−update'
'client−initiated−hands−free−activation'

−−oma−accept−network−initiated−session=[SESSION ID]
Request to accept a network initiated session.

−−oma−reject−network−initiated−session=[SESSION ID]
Request to reject a network initiated session.

−−oma−cancel−session
Request to cancel current OMA session, if any.

SIM OPTIONS

−−pin=PIN
Send **PIN** code to a given SIM card.

−−puk=PUK
Send **PUK** code to a given SIM card. This must be used *WITH* **−−pin**.

−−enable−pin
Enable PIN request for a given SIM card. This must be used *WITH* **−−pin**.

−−disable−pin
Disable PIN request for a given SIM card. This must be used *WITH* **−−pin**.

−−change−pin=PIN
Change the PIN for a given SIM card. It will be set to **PIN**. This must be used *WITH* **−−pin** to supply the old PIN number.

BEARER OPTIONS

All bearer options require the **−−bearer** or **−b** option.

−c, −−connect
Connect to a given bearer.

−x, −−disconnect
Disconnect from a given bearer.

SMS OPTIONS

All SMS options require the **−−sms** or **−s** option.

−−send Send an SMS.

−−store
This option will store the SMS in the default storage defined by the modem, which may be either modem-memory or SMS-memory. To know what the existing default storage is, see the **−−messaging−status** option.

−−store−in−storage=STORAGE
This option states which **STORAGE** to use for SMS messages. Possible values for **STORAGE** include:

- **'sm'** SIM card storage area.
- **'me'** Mobile equipment storage area.
- **'mt'** Sum of SIM and Mobile equipment storages
- **'sr'** Status report message storage area.
- **'bm'** Broadcast message storage area.
- **'ta'** Terminal adaptor message storage area.

−−create−file−with−data=PATH
This option takes an SMS that has *DATA* (not *TEXT*) and will create a local file described by **PATH** and store the content of the SMS there.

CALL OPTIONS

−−start
Initiate an outgoing call.

−−accept
Accept an incoming call.

−−hangup
Reject an incoming call or hangup an ongoing one.

−−send−dtmf=[0−9A−D*#]
Send a DTMF sequence through an ongoing call.

APPLICATION OPTIONS

−J, −−output−json
Run action with machine-friendly JSON output, to be used e.g. by shell scripts that rely on mmcli operations.

−K, −−output−keyvalue
Run action with machine-friendly key-value output, to be used e.g. by shell scripts that rely on mmcli operations.

−v, −−verbose
Perform actions with more details reported and/or logged.

−V, −−version
Returns the version of this program.

−a, −−async
Use asynchronous methods. This is purely a development tool and has no practical benefit to most user operations.

−−timeout=SECONDS
Use **SECONDS** for the timeout when performing operations with this command. This option is useful when executing long running operations, like **−−3gpp−scan**.

EXAMPLES

Send the PIN to the SIM card

You'll need first to know which the proper path/index is for the SIM in your modem:

```
$ mmcli -m 0 -K | grep "modem.generic.sim" | awk -F ": " '{ print $2 }'
/org/freedesktop/ModemManager1/SIM/0
```

And after that, you can just use the SIM index:

```
$ sudo mmcli -i 0 --pin=1234
successfully sent PIN code to the SIM
```

Simple connect and disconnect

You can launch the simple connection process like:

```
$ sudo mmcli -m 0 --simple-connect="pin=1234,apn=internet"
successfully connected the modem
```

Then, you can disconnect it like:

```
$ sudo mmcli -m 0 --simple-disconnect
successfully disconnected all bearers in the modem
```

3GPP network scan

Scanning for 3GPP networks may really take a long time, so a specific timeout must be given:

```
$ sudo mmcli -m 0 --3gpp-scan --timeout=300
 ----------------------
 3GPP scan | networks: 21403 - Orange SP (gprs, unknown)
           |           21407 - Movistar (gprs, unknown)
           |           21404 - YOIGO (gprs, unknown)
           |           21401 - vodafone ES (gprs, unknown)
```

Creating a new SMS message & storing it

Using the "sm" (SIM), you can do this using:

```
$ sudo mmcli -m 0 --messaging-create-sms="text='Hello world',number='+1234567890'"
Successfully created new SMS:
```

```
    /org/freedesktop/ModemManager1/SMS/21 (unknown)

$ sudo mmcli -s 21 --store-in-storage="sm"
successfully stored the SMS

$ sudo mmcli -s 21
  --------------------------------
  General    |       dbus path: /org/freedesktop/ModemManager1/SMS/21
  --------------------------------
  Content    |          number: +1234567890
             |            text: Hello world
  --------------------------------
  Properties |         PDU type: submit
             |           state: stored
             |            smsc: unknown
             |         validity: 0
             |           class: 0
             |         storage: sm
             |   delivery report: not requested
             | message reference: 0

$ sudo mmcli -m 0 --messaging-status
  ----------------------------
  Messaging | supported storages: sm, me
            |    default storage: me
```

Sending binary SMS messages from files

As you can see below, the important part is the **−−messaging−create−sms−with−data** and the **PATH** provided.

```
$ sudo mmcli -m 0 \
     --messaging-create-sms="number='+1234567890'" \
     --messaging-create-sms-with-data=/path/to/your/file
Successfully created new SMS:
   /org/freedesktop/ModemManager1/SMS/22 (unknown)

$ sudo mmcli -s 22 --send
successfully sent the SMS
```

Listing SMS messages

When the receiver gets all the parts of the message, they can now recover the sent file with another **mmcli** command in their ModemManager setup:

```
$> sudo mmcli -m 0 --messaging-list-sms
   /org/freedesktop/ModemManager1/SMS/0 (received)

$> sudo mmcli -s 0 --create-file-with-data=/path/to/the/output/file
```

GPS location status

You first need to check whether the modem has GPS-specific location capabilities. Note that we'll assume the modem is exposed as index 0; if you have more than one modem, just use --list-modems to check the proper modem index:

```
$ mmcli -m 0 --location-status
  ----------------------------
  Location | capabilities: 3gpp-lac-ci, gps-raw, gps-nmea
           |     enabled: none
           |     signals: no
```

The output says that the modem supports 3GPP Location area code/Cell ID, GPS raw and GPS-NMEA location sources. None is enabled yet, as we didn't enable the modem, which we can do issuing:

```
$ sudo mmcli -m 0 --enable
successfully enabled the modem

$ mmcli -m 0 --location-status
  ----------------------------
  Location | capabilities: 3gpp-lac-ci, gps-raw, gps-nmea
           |     enabled: 3gpp-lac-ci
           |     signals: no
```

GPS location technology enabling

We can start the GPS engine by enabling the RAW or NMEA GPS location sources:

```
$ sudo mmcli -m 0 \
          --location-enable-gps-raw \
          --location-enable-gps-nmea
successfully setup location gathering
```

If we do check again the status, we'll see the GPS-specific locations are enabled:

```
$ mmcli -m 0 --location-status
  --------------------------------
  Location | capabilities: 3gpp-lac-ci, gps-raw, gps-nmea
           |     enabled: 3gpp-lac-ci, gps-raw, gps-nmea
           |     signals: no
```

GPS location retrieval

You can query all location information at the same time with a single command. If any of the specific outputs is not available, the corresponding section will be omitted from the output.

```
$ sudo mmcli -m 0 --location-get
  -------------------------
  3GPP location   | Mobile country code: 214
                  | Mobile network code: 3
                  |  Location area code: 21071
                  |            Cell ID: 7033737
  -------------------------
  GPS NMEA traces | $GPGGA,,,,,,0,,,,,,,,*66
                  | $GPRMC,,V,,,,,,,,,,N*53
                  | $GPGSA,A,1,,,,,,,,,,,,,,,*1E
                  | $GPGSV,4,1,16,24,,,,29,,,,05,,,,18,,,*7A
                  | $GPGSV,4,2,16,22,,,,14,,,,11,,,,17,,,*7B
                  | $GPGSV,4,3,16,03,,,,12,,,,30,,,,13,,,*78
                  | $GPGSV,4,4,16,23,,,,15,,,,27,,,,07,,,*79
                  | $GPVTG,,T,,M,,N,,K,N*2C
```

A-GPS support

If A-GPS is enabled before starting the GPS engine, and if a data connection is available in the modem, the configured SUPL servers may be used to obtain a faster initial position fix.

Note that the GPS engine will not be started when just A-GPS capability is enabled. An explicit output (RAW or NMEA) is required to be enabled in order to start the GPS engine.

```
$ mmcli -m 0 --location-status
  --------------------------------
  Location |      capabilities: 3gpp-lac-ci, gps-raw, gps-nmea, agps-msa, agps-msb
           |           enabled: 3gpp-lac-ci
           |           signals: no
  ----------------------------
  GPS      |      refresh rate: 30 seconds
           | a-gps supl server: supl.google.com:7276

$ sudo mmcli -m 0 --location-enable-agps-msa
successfully setup location gathering

$ sudo mmcli -m 0 --location-enable-gps-nmea
successfully setup location gathering

$ sudo mmcli -m 0 --location-enable-gps-raw
successfully setup location gathering
```

Injecting assistance data

If the modem device does not have an ongoing connection (e.g. no mobile network coverage) but the system has other means to access the Internet (e.g. WiFi), the user may be able to download location assistance data and inject it in the module.

E.g. If the device supports XTRA assistance data, the user may download it from one of the servers listed by ModemManager and manually inject it afterwards. The XTRA assistance data is usually valid for several days.

```
$ mmcli -m 0 --location-status
  --------------------------------
  Location |        capabilities: 3gpp-lac-ci, gps-raw, gps-nmea, agps-msa, agps-msb
           |             enabled: 3gpp-lac-ci
           |             signals: no
  --------------------------------
  GPS      |         refresh rate: 30 seconds
           |     a-gps supl server: supl.google.com:7276
           | supported assistance: xtra
           |   assistance servers: https://xtrapath3.izatcloud.net/xtra3grcej.bin
           |                       https://xtrapath1.izatcloud.net/xtra3grcej.bin
           |                       https://xtrapath2.izatcloud.net/xtra3grcej.bin

$ wget -q https://xtrapath3.izatcloud.net/xtra3grcej.bin

$ sudo mmcli -m 0 --location-inject-assistance-data=./xtra3grcej.bin
successfully injected assistance data

$ sudo mmcli -m 0 --location-enable-gps-nmea
successfully setup location gathering
```

```
$ sudo mmcli -m 0 --location-enable-gps-raw
successfully setup location gathering
```

Key-Value output

Writing shell scripts that use mmcli to perform operations with the modem is easy when using the **−−output−keyvalue** option. For example, you could gather all the main status information of the modem with a single call and then parse it to read single fields:

```
$ STATUS=$(mmcli -m 0 --output-keyvalue)
$ echo "${STATUS}" | grep "modem.generic.state " | awk -F ": " '{ print $2 }'
failed
$ echo "${STATUS}" | grep "modem.generic.state-failed-reason " | awk -F ": " '{ print $2 }'
sim-missing
```

AUTHORS

Written by Martyn Russell <martyn@lanedo.com> and Aleksander Morgado <aleksander@aleksander.es>

SEE ALSO

ModemManager(8), **NetworkManager**(8)

AT (http://en.wikipedia.org/wiki/AT_commands).

3GPP (http://en.wikipedia.org/wiki/3GPP).

MCCMNC (http://en.wikipedia.org/wiki/Mobile_Network_Code).

USSD (http://en.wikipedia.org/wiki/Unstructured_Supplementary_Service_Data).

CDMA (http://en.wikipedia.org/wiki/Code_division_multiple_access).

OTA (http://en.wikipedia.org/wiki/Over-the-air_programming).

GPS (http://en.wikipedia.org/wiki/Global_Positioning_System)

NMEA (http://en.wikipedia.org/wiki/NMEA_0183)