Telit

FN980M Appzone Linux

API Reference Guide

1VV0301723 Rev. 2 – 2022-06-17

Telit Technical Documentation

APPLICABILITY TABLE

Products	SW Versions	Modules
FN980M Series	38.02.xx1	5G

CONTENTS

1. INTRODUCTION

1.1. Scope

This document describes the FN980 Family TLB, which is part of the complete FN980 Family Development Kit (Dev-Kit).

1.2. Audience

This document is intended for system integrators using the Telit FN980 family module in their products.

1.3. Contact Information, Support

For technical support and general questions please e-mail:

- *TS-EMEA@telit.com*
- *TS-AMERICAS@telit.com*
- *TS-APAC@telit.com*
- *TS-SRD@telit.com*
- *TS-ONEEDGE@telit.com*

Alternatively, use:

https://www.telit.com/contact-us

Product information and technical documents are accessible 24/7 on our website:

https://www.telit.com

1.4. Symbol Conventions

Danger: This information MUST be followed, or catastrophic equipment failure or personal injury may occur.

Warning: Alerts the user on important steps about the module integration.

Note/Tip: Provides advice and suggestions that may be useful when integrating the module.

Electro-static Discharge: Notifies the user to take proper grounding precautions before handling the product.

All dates are in ISO 8601 format, that is YYYY-MM-DD.

2. FUNCTIONAL OVERVIEW

The Appzone Linux is a high-level connectivity framework that makes available a rich set of functions that can be called by a client application running in the Linux user space. The MCM API offers multi-client architecture in the Linux user space, allowing multiple processes to concurrently leverage resources exported via IoE interfaces. A client of the IoE may be any Linux user-space process, such as an application or daemon.

MCM APIs are message based. To perform a desired command (for example, to establish a data connection), an IoE client sets applicable parameters within an MCM message and invokes the function API to send the message. The response to the command is sent back to the client after the command is processed. The client may also register to receive indications via a callback mechanism. When invoked, the callback receives an indication along with a payload. The payload contains the description of the event received.

There are three message types – requests, responses, and indications. The payload format for each message type is represented in a dedicated structure, defined in a C header file, which is available within the MCM API. For example, mcm_data_start_data_call_req_msg is the message type for parameters for setting up a data connection. Note that types usually have a version suffix, such as _v01, to facilitate versioning for MCM APIs.

Each message can be sent either synchronously or asynchronously, with the difference being that the IoE client's thread, sending an MCM message (request) synchronously, is blocked until a response is received (or a timeout occurs). For MCM messages (requests) being sent asynchronously, the response message is communicated via a callback mechanism.

Before an application can send any messages to the MCM, it is required to perform client initialization.

This can be done by calling the mcm_client_init function. Also, once the application is done using MCM, the client can be released by calling the mcm_client_release function. The below figure illustrates the client initialization and release call flow.

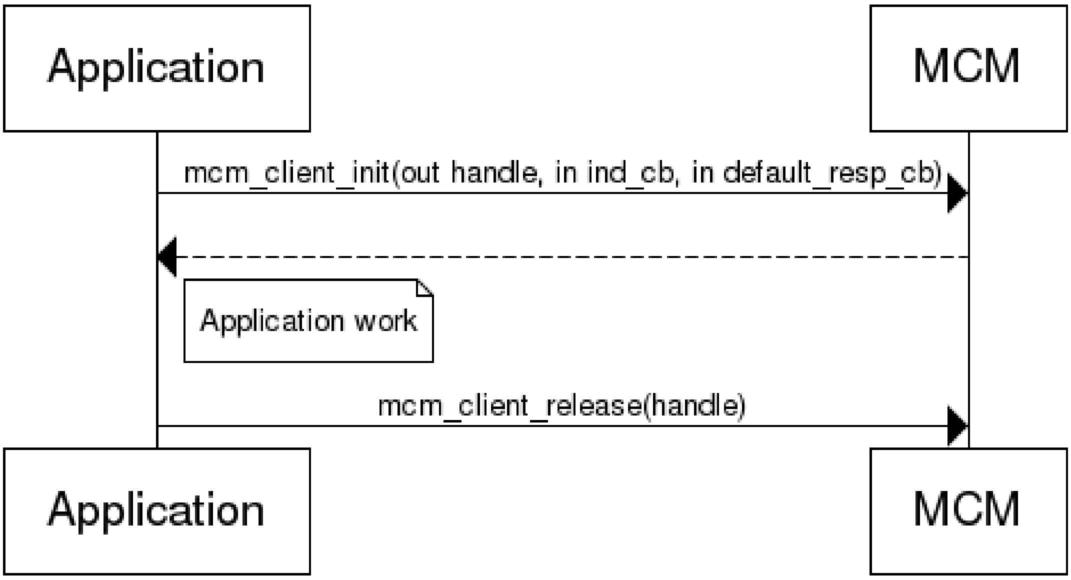
Application
MCM
mcm_client_init(out handle, in ind_cb, in default_resp_cb)
Application work
mcm_client_release(handle)
Application
MCM

Figure 1: Client initialization and release call flow

As part of the client initialization call, the application provides the MCM with a couple of callbacks that can be invoked by the MCM to inform the application that an indication has been received, or when a request is completed and a response has been received.

The following type is used to invoke an asynchronous request callback.

```
typedef void (* mcm_client_async_cb)
(
mcm_client_handle_type hndl,
uint32 msg_id,
void *resp_c_struct,
uint32 resp_len,
void *token_id
);
```

The following type is used to invoke an indication callback.

```
typedef void (* mcm_client_ind_cb)
(
mcm_client_handle_type hndl,
uint32 msg_id,
void *ind_c_struct,
uint32 ind_len
);
```

In the case of a response, the handle (hndl) indicates the client that sent the request, while for an

indication, it is the client to which the indication is being sent.

The parameter msg_id identifies the type of indication or response being sent, and the next parameter will be interpreted based on this type. For example, when receiving a callback for the DIAL request (MCM_VOICE_DIAL_REQ_V01), msg_id is set to MCM_VOICE_DIAL_RESP_V01, and resp_c_struct (ind_c_struct for indications) points to an mcm_voice_dial_resp_msg_v01 structure, the length of which is provided in the resp_len parameter (ind_len for indications).

Note that all callbacks are called in the context of a receiver thread created by the MCM, and some restrictions apply as to what can be done by the application in that context:

- The application should abstain from blocking too long in this context, as any indications or
- outstanding responses are not delivered until the callback has returned
- The application should not send any requests in this context, as that can lead to deadlock
- Sending request messages is achieved through the use of one of the following:
- mcm_client_execute_command_async
- mcm_client_execute_command_sync
- mcm_client_execute_command_sync_ex

mcm_client_execute_command_async sends a request message but does not wait for the response.

Instead, it uses either the supplied callback (if not NULL) or the default callback provided when client initialization was performed. The application must provide a pointer to a request structure, as well as a pointer to a response structure, that the MCM fills with the response received. The response structure is passed back to the application when the callback is invoked.

mcm_client_execute_command_sync and mcm_client_execute_command_sync_ex both

block until a response is received, or in the case of mcm_client_execute_command_sync_ex,

when a timeout expires, if a response is not received before then.

As a way of simplifying the invocation of these functions, since, for every structure pointer parameter, the corresponding length must be passed, the following macros are provided that allow for shorter invocations:

- MCM_CLIENT_EXECUTE_COMMAND_ASYNC
- MCM_CLIENT_EXECUTE_COMMAND_NO_PAYLOAD_ASYNC
- MCM_CLIENT_EXECUTE_COMMAND_SYNC
- MCM_CLIENT_EXECUTE_COMMAND_NO_PAYLOAD_SYNC
- MCM_CLIENT_EXECUTE_COMMAND_SYNC_EX
- MCM_CLIENT_EXECUTE_COMMAND_NO_PAYLOAD_SYNC_EX

The below figure illustrates how to send an MCM_VOICE_DIAL_REQ_V01 message, both synchronously and asynchronously

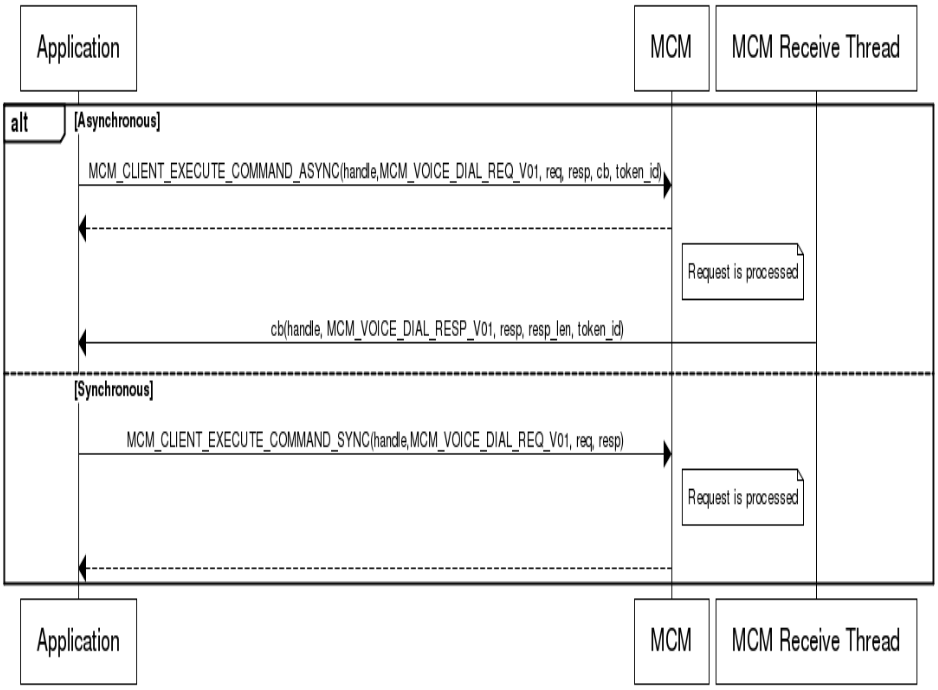
Application
MCM
MCM Receive Thread
alt
[Asynchronous]
MCM_CLIENT_EXECUTE_COMMAND_ASYNC(handle,MCM_VOICE_DIAL_REQ_V01, req, resp, cb, token_id)
Request is processed
cb(handle, MCM_VOICE_DIAL_RESP_V01, resp, resp_len, token_id)
[Synchronous]
MCM_CLIENT_EXECUTE_COMMAND_SYNC(handle,MCM_VOICE_DIAL_REQ_V01, req, resp)
Request is processed
Application
MCM
MCM Receive Thread

Figure 2: Asynchronous and synchronous voice dialing call flow

3. APPZONE LINUX API'S

This chapter contains detailed information about the Mobile Connection Manager (MCM) APIs.

These APIs are provided by Qualcomm and are maintained by Telit.

The available API's are listed below:

- Client Domain
- Device Management
- Network Registration
- Data Calls
- SMS
- Mobile Access Point
- Voice
- Subscriber Identity Module
- Access Terminal Command Process

3.1. Client Domain

This section contains the client-facing interfaces and common types for the MCM.

- Client Functions
- Client Message Identifiers
- Client Message Structures
- Common Enumerations
- Common Data Structures

3.1.1. Client Functions

This section contains the MCM client functions.

3.1.1.1. Function Documentation

3.1.1.1.1. uint32 mcm_client_init (mcm_client_handle_type hndl, mcm_client_ind_cb ind_cb, mcm_client_async_cb default_resp_cb)

Initializes the MCM client library.

This function is to be called before any other functions are called.

Parameters

Out	hndl	Client handle associated with this instance.
In	ind_cb	Indication callback.
In	default_resp_cb	Default response callback for async methods.

Returns

MCM_SUCCESS – 0 is a success.

MCM_ERROR_ – check section 4.1.4 for possible error values.

3.1.1.1.2. uint32 mcm_client_execute_command_async(mcm_client_handle_type hndl, int msg_id, void req_c_struct, int req_c_struct_len, void resp_c_struct, int resp_c_struct_len, mcm_client_async_cb async_resp_cb, void token_id)

Sends a command asynchronously to the service.

Parameters

in	hndl	Client handle associated with this instance.
in	msg_id	Message ID.
in	req_c_struct	Command request structure.
in	req_c_struct_len	Command request structure length.
in	resp_c_struct	Command response structure.
in	resp_c_struct_len	Command response structure length.
in	async_resp_cb	Asynchronous response callback.
in	token_id	Token ID.

Returns

MCM_SUCCESS – 0 is success.

MCM_ERROR – possible error values.

3.1.1.1.3. uint32 mcm_client_execute_command_sync (mcm_client_handle_type hndl, int msg_id, void req_c_struct, int req_c_struct_len, void resp_c_struct, int resp_c_struct_len)

Sends a command synchronously to the service.

Parameters

in	hndl	Client handle associated with this instance.
in	msg_id	Message ID.
in	req_c_struct	Command request structure.
in	req_c_struct_len	Command request structure length.
in	resp_c_struct	Command response structure.
in	resp_c_struct_len	Command response structure length.

Returns

MCM_SUCCESS – 0 is success.

MCM_ERROR_ – check section 4.1.4 possible error values.

Warning: This function uses a hardcoded timeout to execute the command specified by the Message ID. Some Messages can take longer than this fixed timeout to execute, resulting in failure and returning an error. Because of this, we advise using mcm_client_execute_command_sync_ex with a timeout that is large enough.

3.1.1.1.4. uint32 mcm_client_execute_command_sync_ex (mcm_client_handle_type hndl, int msg_id, void req_c_struct, int req_c_struct_len, void resp_c_struct, int resp_c_struct_len, uint32 timeout)

Sends a command synchronously to the service with the specified timeout value.

Parameters

in	hndl	Client handle associated with this instance.
in	msg_id	Message ID.
in	req_c_struct	Command request structure.
in	req_c_struct_len	Command request structure length.
in	resp_c_struct	Command response structure.
in	resp_c_struct_len	Command response structure length.
in	timeout	Timeout in milliseconds.

Returns

MCM_SUCCESS – 0 is success.

MCM_ERROR_ – check section 4.1.4 for possible error values.

3.1.1.1.5. uint32 mcm_client_release (mcm_client_handle_type hndl)

Releases the subscription to the client.

Parameters

in	hndl	Handle associated with this instance and event queue.

Returns

MCM_SUCCESS – 0 is success.

MCM_ERROR_ – check section 4.1.4 for possible error values.

3.1.2. Client Message Identifiers

This section contains the MCM client message identifiers.

- #define MCM_CLIENT_REQUIRE_REQ_V01 0x0800
- #define MCM_CLIENT_REQUIRE_RESP_V01 0x0800
- #define MCM_CLIENT_NOT_REQUIRE_REQ_V01 0x0801
- #define MCM_CLIENT_NOT_REQUIRE_RESP_V01 0x0801

3.1.3. Client Message Structures

This section contains the MSM client message structures.

3.1.3.1. Data Structure Documentation

3.1.3.1.1. struct mcm_client_require_req_msg_v01

Request message; Required client services.

Data fields

Type	Parameter	Description
uint16_t	require_service	Preferred services to be loaded on the device; a bitmask of mcm_client_service_type.

3.1.3.1.2. struct mcm_client_require_resp_msg_v01

Response message; Required client services.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.1.3.1.3. struct mcm_client_not_require_req_msg_v01

Request message; Optional client services.

Data fields

Type	Parameter	Description
uint16_t	not_require_-service	Preferred services to be loaded on the device; a bitmask of mcm_client_service_type.

3.1.3.1.4. struct mcm_client_not_require_resp_msg_v01

Response message; Optional client services.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.1.4. Common Enumerations

This section contains the MCM common enumerations.

3.1.4.1. Enumeration Type Documentation

3.1.4.1.1. enum mcm_result_t_v01

Enumerator:

MCM_RESULT_SUCCESS_V01 Success.

MCM_RESULT_FAILURE_V01 Failure.

3.1.4.1.2. enum mcm_error_t_v01

Possible return values.

Enumerator:

MCM_SUCCESS_V01 Success.

MCM_SUCCESS_CONDITIONAL_SUCCESS_V01 Conditional success.

MCM_ERROR_MCM_SERVICES_NOT_AVAILABLE_V01 MCM services not available.

MCM_ERROR_GENERIC_V01 Generic error.

MCM_ERROR_BADPARM_V01 Bad parameter.

MCM_ERROR_MEMORY_V01 Memory error.

MCM_ERROR_INVALID_STATE_V01 Invalid state.

MCM_ERROR_MALFORMED_MSG_V01 Malformed message.

MCM_ERROR_NO_MEMORY_V01 No memory.

MCM_ERROR_INTERNAL_V01 Internal error.

MCM_ERROR_ABORTED_V01 Action was aborted.

MCM_ERROR_CLIENT_IDS_EXHAUSTED_V01 Client IDs have been exhausted.

MCM_ERROR_UNABORTABLE_TRANSACTION_V01 Unabortable transaction.

MCM_ERROR_INVALID_CLIENT_ID_V01 Invalid client ID.

MCM_ERROR_NO_THRESHOLDS_V01 No thresholds.

MCM_ERROR_INVALID_HANDLE_V01 Invalid handle.

MCM_ERROR_INVALID_PROFILE_V01 Invalid profile.

MCM_ERROR_INVALID_PINID_V01 Invalid PIN ID.

MCM_ERROR_INCORRECT_PIN_V01 Incorrect PIN.

MCM_ERROR_NO_NETWORK_FOUND_V01 No network found.

MCM_ERROR_CALL_FAILED_V01 Call failed.

MCM_ERROR_OUT_OF_CALL_V01 Out of call.

MCM_ERROR_NOT_PROVISIONED_V01 Not provisioned.

MCM_ERROR_MISSING_ARG_V01 Missing argument.

MCM_ERROR_ARG_TOO_LONG_V01 Argument is too long.

MCM_ERROR_INVALID_TX_ID_V01 Invalid Tx ID.

MCM_ERROR_DEVICE_IN_USE_V01 Device is in use.

MCM_ERROR_OP_NETWORK_UNSUPPORTED_V01 OP network is not supported.

MCM_ERROR_OP_DEVICE_UNSUPPORTED_V01 OP device is not supported.

MCM_ERROR_NO_EFFECT_V01 No effect.

MCM_ERROR_NO_FREE_PROFILE_V01 No free profile.

MCM_ERROR_INVALID_PDP_TYPE_V01 Invalid PDP type.

MCM_ERROR_INVALID_TECH_PREF_V01 Invalid technical preference.

MCM_ERROR_INVALID_PROFILE_TYPE_V01 Invalid profile type.

MCM_ERROR_INVALID_SERVICE_TYPE_V01 Invalid service type.

MCM_ERROR_INVALID_REGISTER_ACTION_V01 Invalid register action.

MCM_ERROR_INVALID_PS_ATTACH_ACTION_V01 Invalid PS attach action.

MCM_ERROR_AUTHENTICATION_FAILED_V01 Authentication failed.

MCM_ERROR_PIN_BLOCKED_V01 PIN is blocked.

MCM_ERROR_PIN_PERM_BLOCKED_V01 PIN is permanently blocked.

MCM_ERROR_SIM_NOT_INITIALIZED_V01 SIM is not initialized.

MCM_ERROR_MAX_QOS_REQUESTS_IN_USE_V01 Maximum QoS requests are in use.

MCM_ERROR_INCORRECT_FLOW_FILTER_V01 Incorrect flow filter.

MCM_ERROR_NETWORK_QOS_UNAWARE_V01 Network QoS is unaware.

MCM_ERROR_INVALID_ID_V01 Invalid ID.

MCM_ERROR_INVALID_QOS_ID_V01 Invalid QoS ID.

MCM_ERROR_REQUESTED_NUM_UNSUPPORTED_V01 Requested number is not supported.

MCM_ERROR_INTERFACE_NOT_FOUND_V01 Interface was not found.

MCM_ERROR_FLOW_SUSPENDED_V01 Flow is suspended.

MCM_ERROR_INVALID_DATA_FORMAT_V01 Invalid data format.

MCM_ERROR_GENERAL_V01 General error.

MCM_ERROR_UNKNOWN_V01 Unknown error.

MCM_ERROR_INVALID_ARG_V01 Invalid argument.

MCM_ERROR_INVALID_INDEX_V01 Invalid index.

MCM_ERROR_NO_ENTRY_V01 No entry.

MCM_ERROR_DEVICE_STORAGE_FULL_V01 Device storage is full.

MCM_ERROR_DEVICE_NOT_READY_V01 Device is not ready.

MCM_ERROR_NETWORK_NOT_READY_V01 Network is not ready.

MCM_ERROR_CAUSE_CODE_V01 Cause code error.

MCM_ERROR_MESSAGE_NOT_SENT_V01 Message was not sent.

MCM_ERROR_MESSAGE_DELIVERY_FAILURE_V01 Message delivery failure.

MCM_ERROR_INVALID_MESSAGE_ID_V01 Invalid message ID.

MCM_ERROR_ENCODING_V01 Encoding error.

MCM_ERROR_AUTHENTICATION_LOCK_V01 Authentication lock error.

MCM_ERROR_INVALID_TRANSITION_V01 Invalid transition.

MCM_ERROR_NOT_A_MCAST_IFACE_V01 Not an MCast interface.

MCM_ERROR_MAX_MCAST_REQUESTS_IN_USE_V01 Maximum MCast requests are in use.

MCM_ERROR_INVALID_MCAST_HANDLE_V01 Invalid MCast handle.

MCM_ERROR_INVALID_IP_FAMILY_PREF_V01 Invalid IP family preference.

MCM_ERROR_SESSION_INACTIVE_V01 Session is inactive.

MCM_ERROR_SESSION_INVALID_V01 Session is invalid.

MCM_ERROR_SESSION_OWNERSHIP_V01 Session ownership error.

MCM_ERROR_INSUFFICIENT_RESOURCES_V01 Insufficient resources.

MCM_ERROR_DISABLED_V01 Disabled.

MCM_ERROR_INVALID_OPERATION_V01 Invalid operation.

MCM_ERROR_INVALID_CMD_V01 Invalid command.

MCM_ERROR_TPDU_TYPE_V01 Transfer Protocol data unit type error.

MCM_ERROR_SMSC_ADDR_V01 Short message service center address error.

MCM_ERROR_INFO_UNAVAILABLE_V01 Information is not available.

MCM_ERROR_SEGMENT_TOO_LONG_V01 Segment is too long.

MCM_ERROR_SEGMENT_ORDER_V01 Segment order error.

MCM_ERROR_BUNDLING_NOT_SUPPORTED_V01 Bundling is not supported.

MCM_ERROR_OP_PARTIAL_FAILURE_V01 OP partial failure.

MCM_ERROR_POLICY_MISMATCH_V01 Policy mismatch.

MCM_ERROR_SIM_FILE_NOT_FOUND_V01 SIM file was not found.

MCM_ERROR_EXTENDED_INTERNAL_V01 Extended internal error.

MCM_ERROR_ACCESS_DENIED_V01 Access is denied.

MCM_ERROR_HARDWARE_RESTRICTED_V01 Hardware is restricted.

MCM_ERROR_ACK_NOT_SENT_V01 Acknowledgement was not sent.

MCM_ERROR_INJECT_TIMEOUT_V01 Inject timeout error.

MCM_ERROR_INCOMPATIBLE_STATE_V01 Incompatible state.

MCM_ERROR_FDN_RESTRICT_V01 Fixed dialing number restrict error.

MCM_ERROR_SUPS_FAILURE_CAUSE_V01 SUPS failure cause.

MCM_ERROR_NO_RADIO_V01 No radio.

MCM_ERROR_NOT_SUPPORTED_V01 Not supported.

MCM_ERROR_NO_SUBSCRIPTION_V01 No subscription.

MCM_ERROR_CARD_CALL_CONTROL_FAILED_V01 Card call control failed.

MCM_ERROR_NETWORK_ABORTED_V01 Network was aborted.

MCM_ERROR_MSG_BLOCKED_V01 Message was blocked.

MCM_ERROR_INVALID_SESSION_TYPE_V01 Invalid session type.

MCM_ERROR_INVALID_PB_TYPE_V01 Invalid phonebook type.

MCM_ERROR_NO_SIM_V01 No SIM was found.

MCM_ERROR_PB_NOT_READY_V01 Phonebook not ready.

MCM_ERROR_PIN_RESTRICTION_V01 PIN restriction.

MCM_ERROR_PIN2_RESTRICTION_V01 PIN2 restriction.

MCM_ERROR_PUK_RESTRICTION_V01 PIN unlocking key restriction.

MCM_ERROR_PUK2_RESTRICTION_V01 PIN unlocking key2 restriction.

MCM_ERROR_PB_ACCESS_RESTRICTED_V01 Phonebook access is restricted.

MCM_ERROR_PB_DELETE_IN_PROG_V01 Phonebook delete is in progress.

MCM_ERROR_PB_TEXT_TOO_LONG_V01 Phonebook text is too long.

MCM_ERROR_PB_NUMBER_TOO_LONG_V01 Phonebook number is too long.

MCM_ERROR_PB_HIDDEN_KEY_RESTRICTION_V01 Phonebook hidden key restriction.

MCM_ERROR_PB_NOT_AVAILABLE_V01 Phonebook is not available.

MCM_ERROR_DEVICE_MEMORY_ERROR_V01 Device memory error.

MCM_ERROR_SIM_PIN_BLOCKED_V01 SIM PIN is blocked.

MCM_ERROR_SIM_PIN_NOT_INITIALIZED_V01 SIM PIN is not initialized.

MCM_ERROR_SIM_INVALID_PIN_V01 SIM PIN is invalid.

MCM_ERROR_SIM_INVALID_PERSO_CK_V01 SIM invalid personalization CK.

MCM_ERROR_SIM_PERSO_BLOCKED_V01 SIM personalization blocked.

MCM_ERROR_SIM_PERSO_INVALID_DATA_V01 SIM personalization contains invalid data.

MCM_ERROR_SIM_ACCESS_DENIED_V01 SIM access is denied.

MCM_ERROR_SIM_INVALID_FILE_PATH_V01 SIM file path is invalid.

MCM_ERROR_SIM_SERVICE_NOT_SUPPORTED_V01 SIM service is not supported.

MCM_ERROR_SIM_AUTH_FAIL_V01 SIM authorization failure.

MCM_ERROR_SIM_PIN_PERM_BLOCKED_V01 SIM PIN is permanently blocked.

3.1.5. Common Data Structures

This section contains the MCM common data structures.

3.1.5.1. Data Structure Documentation

3.1.5.1.1. struct mcm_response_t_v01

Data fields

Type	Parameter	Description
mcm_result_t_-v01	result	Result code: MCM_RESULT_SUCCESS MCM_RESULT_FAILURE
mcm_error_t_-v01	error	Error code. Possible error code values are described in the error codes section of each message definition.

3.2. Device Management

This section contains the Device Management (DM) data types for managing a mobile connection device, such as a modem, a fusion of modems, or multiple modems, using MCM.

- DM Message Identifiers
- DM Message Structures
- DM Constants
- DM Enumerations

3.2.1. DM Message Identifiers

This section contains the MCM DM message identifiers.

- #define MCM_DM_GET_RADIO_MODE_REQ_V01 0x0201
- #define MCM_DM_GET_RADIO_MODE_RESP_V01 0x0201
- #define MCM_DM_SET_RADIO_MODE_REQ_V01 0x0202
- #define MCM_DM_SET_RADIO_MODE_RESP_V01 0x0202
- #define MCM_DM_EVENT_REGISTER_REQ_V01 0x0203
- #define MCM_DM_EVENT_REGISTER_RESP_V01 0x0203
- #define MCM_DM_RADIO_MODE_CHANGED_EVENT_IND_V01 0x0204
- #define MCM_DM_GET_MODEL_NAME_REQ_V01 0x0280
- #define MCM_DM_GET_MODEL_NAME_RESP_V01 0x0280
- #define MCM_DM_GET_SOFTWARE_VERSION_REQ_V01 0x0281
- #define MCM_DM_GET_SOFTWARE_VERSION_RESP_V01 0x0281
- #define MCM_DM_GET_SERIAL_NUMBER_REQ_V01 0x0282
- #define MCM_DM_GET_SERIAL_NUMBER_RESP_V01 0x0282

3.2.2. DM Message Structures

This section contains the MCM DM message structures.

3.2.2.1. Data Structure Documentation

3.2.2.1.1. struct mcm_dm_get_radio_mode_resp_msg_v01

Response message; Gets the radio mode of the device.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	radio_mode_-valid	Must be set to TRUE if radio_mode is being passed.
mcm_dm_-radio_mode_t_- v01	radio_mode	Current radio mode; must be one of the modes in mcm_dm_radio_mode_t_v01.

3.2.2.1.2. struct mcm_dm_set_radio_mode_req_msg_v01

Request message; Sets the device radio mode.

Data fields

Type	Parameter	Description
mcm_dm_-radio_mode_t_- v01	radio_mode	Radio mode to set.

3.2.2.1.3. struct mcm_dm_set_radio_mode_resp_msg_v01

Response message; Sets the device radio mode.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	no_change_-valid	Must be set to TRUE if no_change is being passed.
uint8_t	no_change	No change.

3.2.2.1.4. struct mcm_dm_event_register_req_msg_v01

Request message; Registers for an indication of events.

Data fields

Type	Parameter	Description
uint8_t	register_radio_-mode_changed-_event_valid	Must be set to TRUE if register_radio_mode_changed_event is being passed.
uint8_t	register_radio_-mode_changed-_event	Radio mode changed event.

3.2.2.1.5. struct mcm_dm_event_register_resp_msg_v01

Response message; Registers for an indication of events.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Response.

3.2.2.1.6. struct mcm_dm_radio_mode_changed_event_ind_msg_v01

Indication message; Indication when the radio mode is changed.

Data fields

Type	Parameter	Description
uint8_t	radio_mode_-valid	Must be set to TRUE if radio_mode is being passed.
mcm_dm_-radio_mode_t_- v01	radio_mode	Radio mode.

3.2.3. DM Constants

This section contains the MCM DM constants.

3.2.4. Define Documentation

#define MCM_MAX_ARRAY_LIMIT_V01 252

3.2.5. DM Enumerations

This section contains the MCM DM enums.

3.2.5.1. Enumeration Type Documentation

3.2.5.1.1. enum mcm_dm_radio_mode_t_v01

Enumerator:

MCM_DM_RADIO_MODE_OFFLINE_V01 Radio power off or unknown.

MCM_DM_RADIO_MODE_ONLINE_V01 Radio online

MCM_DM_RADIO_MODE_UNAVAILABLE_V01 Radio unavailable.

3.3. Network Registration

This section contains the messages, constants, data structures, and enumerations for managing and reporting the mobile Network (NW) connections, their status, and statistics, using MCM.

- Network Message Identifiers
- Network Message Structures
- NW Constants
- NW Enumerations
- NW Data Structures

3.3.1. Network Message Identifiers

This section contains the MCM network message identifiers.

- #define MCM_NW_SET_CONFIG_REQ_V01 0x0500
- #define MCM_NW_SET_CONFIG_RESP_V01 0x0500
- #define MCM_NW_GET_CONFIG_REQ_V01 0x0501
- #define MCM_NW_GET_CONFIG_RESP_V01 0x0501
- #define MCM_NW_GET_REGISTRATION_STATUS_REQ_V01 0x0502
- #define MCM_NW_GET_REGISTRATION_STATUS_RESP_V01 0x0502
- #define MCM_NW_SCAN_REQ_V01 0x0503
- #define MCM_NW_SCAN_RESP_V01 0x0503
- #define MCM_NW_GET_OPERATOR_NAME_REQ_V01 0x0504
- #define MCM_NW_GET_OPERATOR_NAME_RESP_V01 0x0504
- #define MCM_NW_SCREEN_ON_OFF_REQ_V01 0x0505

- #define MCM_NW_SCREEN_ON_OFF_RESP_V01 0x0505
- #define MCM_NW_SELECTION_REQ_V01 0x0506
- #define MCM_NW_SELECTION_RESP_V01 0x0506
- #define MCM_NW_GET_SIGNAL_STRENGTH_REQ_V01 0x0507
- #define MCM_NW_GET_SIGNAL_STRENGTH_RESP_V01 0x0507
- #define MCM_NW_GET_CELL_ACCESS_STATE_REQ_V01 0x0508
- #define MCM_NW_GET_CELL_ACCESS_STATE_RESP_V01 0x0508
- #define MCM_NW_GET_NITZ_TIME_INFO_REQ_V01 0x0509
- #define MCM_NW_GET_NITZ_TIME_INFO_RESP_V01 0x0509
- #define MCM_NW_EVENT_REGISTER_REQ_V01 0x050A
- #define MCM_NW_EVENT_REGISTER_RESP_V01 0x050A
- #define MCM_NW_VOICE_REGISTRATION_EVENT_IND_V01 0x050B
- #define MCM_NW_DATA_REGISTRATION_EVENT_IND_V01 0x050C
- #define MCM_NW_SIGNAL_STRENGTH_EVENT_IND_V01 0x050D
- #define MCM_NW_CELL_ACCESS_STATE_CHANGE_EVENT_IND_V01 0x050E
- #define MCM_NW_NITZ_TIME_IND_V01 0x050F

3.3.2. Network Message Structures

This section contains the MCM network message structures.

3.3.2.1. Data Structure Documentation

3.3.2.1.1. struct mcm_nw_set_config_req_msg_v01

Request message; Configures the settings that define the MCM network interface.

Data fields

Type	Parameter	Description
uint8_t	preferred_nw_-mode_valid	Must be set to TRUE if preferred_nw_mode is being passed.
uint64_t	preferred_nw_-mode	Preferred network mode for connections; a bitmask of mcm_nw_mode.
uint8_t	roaming_pref_-valid	Must be set to TRUE if roaming_pref is being passed.

mcm_nw_- roam_state_t_- v01	roaming_pref	Roaming preference.

3.3.2.1.2. struct mcm_nw_set_config_resp_msg_v01

Response message; Configures the settings that define the MCM network interface.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.3.2.1.3. struct mcm_nw_get_config_resp_msg_v01

Response message; Gets the configuration status for this network interface.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	preferred_nw_-mode_valid	Must be set to TRUE if preferred_nw_mode is being passed.
uint64_t	preferred_nw_-mode	Preferred network mode for connections; a bitmask of mcm_nw_mode.
uint8_t	roaming_pref_-valid	Must be set to TRUE if roaming_pref is being passed.
mcm_nw_-roam_state_t_- v01	roaming_pref	Roaming preference.

3.3.2.1.4. struct mcm_nw_get_registration_status_resp_msg_v01

Response message; Gets the status associated with the connection of <id>.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	voice_-registration_- valid	Must be set to TRUE if voice_registration is being passed.
mcm_nw_-common_-registration_- t_v01	voice_-registration	Voice registration.
uint8_t	data_-registration_- valid	Must be set to TRUE if data_registration is being passed.

mcm_nw_-common_-registration_- t_v01	data_-registration	Data registration.
uint8_t	voice_-registration_-details_3gpp_-valid	Must be set to TRUE if voice_registration_details_3gpp is being passed.
mcm_nw-_3gpp_-registration_- t_v01	voice_-registration_-details_3gpp	Voice registration details for 3GPP.

Type	Parameter	Description
uint8_t	data_-registration_-details_3gpp_-valid	Must be set to TRUE if data_registration_details_3gpp is being passed.
mcm_nw-_3gpp_-registration_- t_v01	data_-registration_-details_3gpp	Data registration details for 3GPP.
uint8_t	voice_-registration_-details_3gpp2_-valid	Must be set to TRUE if voice_registration_details_3gpp2 is being passed.
mcm_nw-_3gpp2_-registration_- t_v01	voice_-registration_-details_3gpp2	Voice registration details for 3GPP2.
uint8_t	data_-registration_-details_3gpp2_-valid	Must be set to TRUE if data_registration_details_3gpp2 is being passed.
mcm_nw-_3gpp2_-registration_- t_v01	data_-registration_-details_3gpp2	Data registration details for 3GPP2.

3.3.2.1.5. struct mcm_nw_scan_resp_msg_v01

Response message; Gets the status associated with the connection of <id>.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	entry_valid	Must be set to TRUE if entry is being passed.
uint32_t	entry_len	Must be set to the number of elements in the entry.

mcm_nw_scan-_entry_t_v01	entry	Scan entry.

3.3.2.1.6. struct mcm_nw_get_operator_name_resp_msg_v01

Response message; Gets the operator name associated with the connection of <id>.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	operator_name-_valid	Must be set to TRUE if operator_name is being passed.
mcm_nw_-operator_name-_t_v01	operator_name	Operator name.

3.3.2.1.7. struct mcm_nw_screen_on_off_req_msg_v01

Request message; Turns the screen on/off to save the battery.

Data fields

Type	Parameter	Description
uint8_t	turn_off_screen	Turn the screen off.

3.3.2.1.8. struct mcm_nw_screen_on_off_resp_msg_v01

Response message; Turns the screen on/off to save the battery.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.3.2.1.9. struct mcm_nw_selection_req_msg_v01

Request message; Network selection (manual or automatic).

Data fields

Type	Parameter	Description
mcm_nw_-selection_t_v01	nw_selection_-info	Network selection information.

3.3.2.1.10. struct mcm_nw_selection_resp_msg_v01

Response message; Network selection (manual or automatic).

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.3.2.1.11. struct mcm_nw_get_signal_strength_resp_msg_v01

Response message; Gets signal strength information.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	gsm_sig_info_-valid	Must be set to TRUE if gsm_sig_info is being passed.
mcm_nw_gsm-_sig_info_t_- v01	gsm_sig_info	GSM signal information.
uint8_t	wcdma_sig_-info_valid	Must be set to TRUE if wcdma_sig_info is being passed.
mcm_nw_-wcdma_sig_-info_t_v01	wcdma_sig_-info	WCDMA signal information.
uint8_t	tdscdma_sig_-info_valid	Must be set to TRUE if tdscdma_sig_info is being passed.
mcm_nw_-tdscdma_sig_-info_t_v01	tdscdma_sig_-info	TDSCDMA signal information.
uint8_t	lte_sig_info_-valid	Must be set to TRUE if lte_sig_info is being passed.
mcm_nw_lte_-sig_info_t_v01	lte_sig_info	LTE signal information.
uint8_t	cdma_sig_info-_valid	Must be set to TRUE if cdma_sig_info is being passed.
mcm_nw_-cdma_sig_info-_t_v01	cdma_sig_info	CDMA signal information.
uint8_t	hdr_sig_info_-valid	Must be set to TRUE if hdr_sig_info is being passed.
mcm_nw_hdr_-sig_info_t_v01	hdr_sig_info	HDR signal information.

3.3.2.1.12. struct mcm_nw_get_cell_access_state_resp_msg_v01

Response message; Gets the cell access state.

Data fields

Type	Parameter	Description
mcm_response_t_v01	response	Result code.
uint8_t	nw_cell_access_state_valid	Must be set to TRUE if nw_cell_access_state is being passed.
mcm_nw_cell_access_state_t_v01	nw_cell_access_state	Network cell access state.

3.3.2.1.13. struct mcm_nw_get_nitz_time_info_resp_msg_v01

Response message; Get NITZ time information.

Data fields

Type	Parameter	Description
mcm_response_t_v01	response	Result code.
uint8_t	nw_nitz_time_valid	Must be set to TRUE if nw_nitz_time is being passed
mcm_nw_nitz_time_t_v01	nw_nitz_time	Network NITZ time.
uint8_t	abs_time_valid	Must be set to true if abs_time is being passed
uint64_t	abs_time	
uint8_t	leap_sec_valid	Must be set to true if leap_sec is being passed
int8_t	leap_sec	

3.3.2.1.14. struct mcm_nw_event_register_req_msg_v01

Request message; Registers for an indication of events.

Data fields

Type	Parameter	Description
uint8_t	register_voice_registration_event_valid	Must be set to TRUE if register_voice_registration_event is being passed.

Type	Parameter	Description
uint8_t	register_voice_-registration_- event	Register for a voice registration event.
uint8_t	register_data_-registration_-event_valid	Must be set to TRUE if register_data_registration_event is being passed.
uint8_t	register_data_-registration_- event	Register for a data registration event.
uint8_t	register_signal-_strength_-event_valid	Must be set to TRUE if register_signal_strength_event is being passed.
uint8_t	register_signal-_strength_event	Register for a signal strength event.
uint8_t	register_cell_-access_state_-change_event_-valid	Must be set to TRUE if register_cell_access_state_change_event is being passed.
uint8_t	register_cell_-access_state_-change_event	Register for a cell access state change event.
uint8_t	register_nitz_-time_update_-event_valid	Must be set to TRUE if register_nitz_time_update_event is being passed.
uint8_t	register_nitz_-time_update_-event	Register for a NITZ time update event.

3.3.2.1.15. struct mcm_nw_event_register_resp_msg_v01

Response message; Registers for an indication of events.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.3.2.1.16. struct mcm_nw_voice_registration_event_ind_msg_v01

Indication message; Indication for the corresponding registered event

MCM_NW_VOICE_REGISTRATION_EV.

Data fields

Type	Parameter	Description
uint8_t	voice_-registration_- valid	Must be set to TRUE if voice_registration is being passed.
mcm_nw_-common_-registration_- t_v01	voice_-registration	Voice registration.
uint8_t	voice_-registration_-details_3gpp_-valid	Must be set to TRUE if voice_registration_details_3gpp is being passed.
mcm_nw-_3gpp_-registration_- t_v01	voice_-registration_-details_3gpp	Voice registration details for 3GPP.
uint8_t	voice_-registration_-details_3gpp2_-valid	Must be set to TRUE if voice_registration_details_3gpp2 is being passed.
mcm_nw-_3gpp2_-registration_- t_v01	voice_-registration_-details_3gpp2	Voice registration details for 3GPP2.

3.3.2.1.17. struct mcm_nw_data_registration_event_ind_msg_v01

Indication message; Indication for the corresponding registered event

MCM_NW_DATA_REGISTRATION_EV.

Data fields

Type	Parameter	Description
uint8_t	data_-registration_- valid	Must be set to TRUE if data_registration is being passed.
mcm_nw_-common_-registration_- t_v01	data_-registration	Data registration.
uint8_t	data_-registration_-details_3gpp_-valid	Must be set to TRUE if data_registration_details_3gpp is being passed.
mcm_nw-_3gpp_-registration_- t_v01	data_-registration_-details_3gpp	Data registration details for 3GPP.

Type	Parameter	Description
uint8_t	data_-registration_-details_3gpp2_-valid	Must be set to TRUE if data_registration_details_3gpp2 is being passed.
mcm_nw-_3gpp2_-registration_- t_v01	data_-registration_-details_3gpp2	Data registration details for 3GPP2.

3.3.2.1.18. struct mcm_nw_signal_strength_event_ind_msg_v01

Indication message; Indication for the corresponding registered event

MCM_NW_SIGNAL_STRENGTH_EV.

Data Fields

Type	Parameter	Description
uint8_t	gsm_sig_info_-valid	Must be set to TRUE if gsm_sig_info is being passed.
mcm_nw_gsm-_sig_info_t_- v01	gsm_sig_info	GSM signal information.
uint8_t	wcdma_sig_-info_valid	Must be set to TRUE if wcdma_sig_info is being passed.
mcm_nw_-wcdma_sig_-info_t_v01	wcdma_sig_-info	WCDMA signal information.
uint8_t	tdscdma_sig_-info_valid	Must be set to TRUE if tdscdma_sig_info is being passed.
mcm_nw_-tdscdma_sig_-info_t_v01	tdscdma_sig_-info	TDSCDMA signal information.
uint8_t	lte_sig_info_-valid	Must be set to TRUE if lte_sig_info is being passed.
mcm_nw_lte_-sig_info_t_v01	lte_sig_info	LTE signal information.
uint8_t	cdma_sig_info-_valid	Must be set to TRUE if cdma_sig_info is being passed.
mcm_nw_-cdma_sig_info-_t_v01	cdma_sig_info	CDMA signal information.
uint8_t	hdr_sig_info_-valid	Must be set to TRUE if hdr_sig_info is being passed.
mcm_nw_hdr_-sig_info_t_v01	hdr_sig_info	HDR signal information.

3.3.2.1.19. struct mcm_nw_cell_access_state_change_event_ind_msg_v01

Indication message; Indication for a change in the cell access state, e.g., emergency only, CS call only.

Data fields

Type	Parameter	Description
mcm_nw_cell-_access_state_-t_v01	nw_cell_-access_state	Network cell access state.

3.3.2.1.20. struct mcm_nw_nitz_time_ind_msg_v01

Indication message; Indication to update NITZ time.

Data fields

Type	Parameter	Description
mcm_nw_nitz-_time_t_v01	nw_nitz_time	NITZ time upd
uint8_t	abs_time_valid	Must be set to true if abs_time is being passed
uint64_t	abs_time	
uint8_t	leap_sec_valid	Must be set to true if leap_sec is being passed
int8_t	leap_sec	

3.3.3. NW Constants

This section contains the MCM network registration constants.

The network selection modes are ORed together to specify the preferred network for searching and registrations. For example, MCM_NW_MODE_GSM j MCM_NW_MODE_WCDMA selects GSM/WCDMA networks with a preference for WCDMA.

MCM_NW_MODE_GSM j MCM_NW_MODE_WCDMA j MCM_NW_MODE_PRL is the same, but gives preference according to the roaming list.

3.3.3.1. Define Documentation

#define MCM_NW_SCAN_LIST_MAX_V01 40 Maximum items in the network scan list.

#define MCM_MODE_NONE_V01 ((mcm_nw_mode_type_v01)0x00ull) No network.

#define MCM_MODE_GSM_V01 ((mcm_nw_mode_type_v01)0x01ull) Include GSM networks.

#define MCM_MODE_WCDMA_V01 ((mcm_nw_mode_type_v01)0x02ull) Include WCDMA networks.

#define MCM_MODE_CDMA_V01 ((mcm_nw_mode_type_v01)0x04ull) Include CDMA networks.

#define MCM_MODE_EVDO_V01 ((mcm_nw_mode_type_v01)0x08ull) Include EVDO networks.

#define MCM_MODE_LTE_V01 ((mcm_nw_mode_type_v01)0x10ull) Include LTE networks.

#define MCM_MODE_TDSCDMA_V01 ((mcm_nw_mode_type_v01)0x20ull) Include NR5G networks.

#define MCM_MODE_NR5G_V01 ((mcm_nw_mode_type_v01) 0x40ull) Include TDSCDMA networks

#define MCM_MODE_PRL_V01 ((mcm_nw_mode_type_v01)0x10000ull) Give preference according to the preferred roaming list.

3.3.4. NW Enumerations

This section contains the MCM network registration enums.

3.3.4.1. Enumeration Type Documentation

3.3.4.1.1. enum mcm_nw_service_t_v01

Enumerator:

MCM_NW_SERVICE_NONE_V01 Not registered or no data.

MCM_NW_SERVICE_LIMITED_V01 Registered; emergency service only.

MCM_NW_SERVICE_FULL_V01 Registered, full service.

3.3.4.1.2. enum mcm_nw_selection_type_t_v01

Enumerator:

MCM_NW_SELECTION_AUTOMATIC_V01 Automatic network selection.

MCM_NW_SELECTION_MANUAL_V01 Manual network selection.

3.3.4.1.3. enum mcm_nw_network_status_t_v01

Enumerator:

MCM_NW_NETWORK_STATUS_NONE_V01 Network status not available.
MCM_NW_NETWORK_STATUS_CURRENT_SERVING_V01 Current serving network.
MCM_NW_NETWORK_STATUS_PREFERRED_V01 Preferred network.
MCM_NW_NETWORK_STATUS_NOT_PREFERRED_V01 Not the preferred network.
MCM_NW_NETWORK_STATUS_AVAILABLE_V01 Service available.
MCM_NW_NETWORK_STATUS_FORBIDDEN_V01 Forbidden service.

3.3.4.1.4. enum mcm_nw_radio_tech_t_v01

Enumerator:

MCM_NW_RADIO_TECH_GSM_V01 GSM; only supports voice.

MCM_NW_RADIO_TECH_HSPAP_V01 HSPA+.

MCM_NW_RADIO_TECH_LTE_V01 LTE.

MCM_NW_RADIO_TECH_EHRPD_V01 EHRPD.

MCM_NW_RADIO_TECH_EVDO_B_V01 EVDO B.

MCM_NW_RADIO_TECH_HSPA_V01 HSPA.

MCM_NW_RADIO_TECH_HSUPA_V01 HSUPA.

MCM_NW_RADIO_TECH_HSDPA_V01 HSDPA.

MCM_NW_RADIO_TECH_EVDO_A_V01 EVDO A.

MCM_NW_RADIO_TECH_EVDO_0_V01 EVDO 0.

MCM_NW_RADIO_TECH_1xRTT_V01 1xRTT.

MCM_NW_RADIO_TECH_IS95B_V01 IS95B.

MCM_NW_RADIO_TECH_IS95A_V01 IS95A.

MCM_NW_RADIO_TECH_UMTS_V01 UMTS.

MCM_NW_RADIO_TECH_EDGE_V01 EDGE.

MCM_NW_RADIO_TECH_GPRS_V01 GPRS.

MCM_NW_RADIO_TECH_NONE_V01 No technology selected.

3.3.4.1.5. enum mcm_nw_cell_access_state_t_v01

Enumerator:

MCM_NW_CELL_ACCESS_NONE_V01 Unknown cell access state.

MCM_NW_CELL_ACCESS_NORMAL_ONLY_V01 Cell access is allowed for normal calls only.

MCM_NW_CELL_ACCESS_EMERGENCY_ONLY_V01 Cell access is allowed for emergency calls only.

MCM_NW_CELL_ACCESS_NO_CALLS_V01 Cell access is not allowed for any call type.
MCM_NW_CELL_ACCESS_ALL_CALLS_V01 Cell access is allowed for all call types.

3.3.4.1.6. enum mcm_nw_roam_state_t_v01

Enumerator:

MCM_NW_ROAM_STATE_OFF_V01 None, or roaming indicator off.

MCM_NW_ROAM_STATE_ON_V01 Roaming indicator on.

3.3.4.1.7. enum mcm_nw_tech_domain_t_v01

Enumerator:

MCM_NW_TECH_DOMAIN_NONE_V01 None.

MCM_NW_TECH_DOMAIN_3GPP_V01 3GPP.

MCM_NW_TECH_DOMAIN_3GPP2_V01 3GPP2.

3.3.4.1.8. enum mcm_nw_deny_reason_t_v01

Enumerator:

MCM_NW_IMSI_UNKNOWN_HLR_DENY_REASON_V01 IMSI unknown in HLR.
MCM_NW_ILLEGAL_MS_DENY_REASON_V01 Illegal MS.
MCM_NW_IMSI_UNKNOWN_VLR_DENY_REASON_V01 IMSI unknown in VLR.
MCM_NW_IMEI_NOT_ACCEPTED_DENY_REASON_V01 IMEI not accepted.
MCM_NW_ILLEGAL_ME_DENY_REASON_V01 Illegal ME.
MCM_NW_PLMN_NOT_ALLOWED_DENY_REASON_V01 PLMN not allowed.
MCM_NW_LA_NOT_ALLOWED_DENY_REASON_V01 Location area not allowed.
MCM_NW_ROAMING_NOT_ALLOWED_LA_DENY_REASON_V01
Roaming is not allowed in this location area.

MCM_NW_NO_SUITABLE_CELLS_LA_DENY_REASON_V01
No suitable cells in the location area.

MCM_NW_NETWORK_FAILURE_DENY_REASON_V01 Network failure.
MCM_NW_MAC_FAILURE_DENY_REASON_V01 MAC failure.

MCM_NW_SYNCH_FAILURE_DENY_REASON_V01 Sync failure.
MCM_NW_CONGESTION_DENY_REASON_V01 Congestion.

MCM_NW_GSM_AUTHENTICATION_UNACCEPTABLE_DENY_REASON_V01 unacceptable.

MCM_NW_NOT_AUTHORIZED_CSG_DENY_REASON_V01 Not authorized in this CSG.
MCM_NW_SERVICE_OPTION_NOT_SUPPORTED_DENY_REASON_V01 Service option not supported.

MCM_NW_REQ_SERVICE_OPTION_NOT_SUBSCRIBED_DENY_REASON_V01 Requested service option not subscribed.

MCM_NW_CALL_CANNOT_BE_IDENTIFIED_DENY_REASON_V01 Call cannot be identified.

MCM_NW_SEMANTICALLY_INCORRECT_MSG_DENY_REASON_V01 Semantically incorrect

message.

MCM_NW_INVALID_MANDATORY_INFO_DENY_REASON_V01
Invalid mandatory information.
MCM_NW_MSG_TYPE_NON_EXISTENT_DENY_REASON_V01
Message type is non-existent or not implemented.

MCM_NW_INFO_ELEMENT_NON_EXISTENT_DENY_REASON_V01
Message type is not compatible with the protocol state.

MCM_NW_CONDITIONAL_IE_ERR_DENY_REASON_V01 Conditional IE error.

MCM_NW_MSG_INCOMPATIBLE_PROTOCOL_STATE_DENY_REASON_V01 Message not compatible with the protocol state.

MCM_NW_PROTOCOL_ERROR_DENY_REASON_V01 Unspecified protocol error.

3.3.5. NW Data Structures

This section contains the MCM network registration data structures.

3.3.5.1. Data Structure Documentation

3.3.5.1.1. struct mcm_nw_operator_name_t_v01

Data fields

Type	Parameter	Description
char	long_eons	Long EONS.
char	short_eons	Short EONS.
char	mcc	Mobile country code.
char	mnc	Mobile network code.

3.3.5.1.2. struct mcm_nw_scan_entry_t_v01

Data fields

Type	Parameter	Description
mcm_nw_-operator_name-_t_v01	operator_name	Operator name.
mcm_nw_-network_status-_t_v01	network_status	Network status.
mcm_nw_-radio_tech_t_- v01	rat	Radio technology.

3.3.5.1.3. struct mcm_nw_common_registration_t_v01

Data fields

Type	Parameter	Description
mcm_nw_tech-_domain_t_v01	tech_domain	Technology used to determine the structure type mcm_tech: 0 – None, 1 – 3GPP, 2 – 3GPP2.
mcm_nw_-radio_tech_t_- v01	radio_tech	Radio technology; see mcm_nw_radio_tech_t_v01.
char	mcc	Mobile country code.
char	mnc	Mobile network code.
mcm_nw_-roam_state_t_- v01	roaming	0 – Off, 1 – Roaming (3GPP2 has extended values).
uint8_t	forbidden	Forbidden: 0 – No, 1 – Yes.
uint32_t	cid	Cell ID for the registered 3GPP system.
uint16_t	lac	Location area code for the registered 3GPP system.
uint16_t	psc	Primary scrambling code (WCDMA only); 0 – None.
uint16_t	tac	Tracking area code information for LTE.

3.3.5.1.4. struct mcm_nw_3gpp_registration_t_v01

Data fields

Type	Parameter	Description
mcm_nw_tech-_domain_t_v01	tech_domain	Technology, used to determine structure type mcm_tech: 0 – None, 1 – 3GPP, 2 – 3GPP2.
mcm_nw_-radio_tech_t_- v01	radio_tech	Radio technology; see mcm_nw_radio_tech_t_v01.
char	mcc	Mobile country code.
char	mnc	Mobile network code.
mcm_nw_-roam_state_t_- v01	roaming	Roaming status; see mcm_nw_roam_state_t_v01.
uint8_t	forbidden	Forbidden: 0 – No, 1 – Yes.
uint8_t	inPRL	0 – Not in PRL, 1 – In PRL.
uint8_t	css	Concurrent services supported: 0 – No, 1 – Yes.
uint16_t	sid	CDMA system ID.
uint16_t	nid	CDMA network ID.
uint16_t	bsid	Base station ID.

3.3.5.1.5. struct mcm_nw_3gpp2_registration_t_v01

Data fields

Type	Parameter	Description
mcm_nw_tech-_domain_t_v01	tech_domain	Technology, used to determine structure type mcm_tech: 0 – None, 1 – 3GPP, 2 – 3GPP2.
mcm_nw_-radio_tech_t_- v01	radio_tech	Radio technology; see mcm_nw_radio_tech_t_v01.
char	mcc	Mobile country code.
char	mnc	Mobile network code.
mcm_nw_-roam_state_t_- v01	roaming	Roaming status; see mcm_nw_roam_state_t_v01.
uint8_t	forbidden	Forbidden: 0 – No, 1 – Yes.
uint8_t	inPRL	0 – Not in PRL, 1 – In PRL.
uint8_t	css	Concurrent services supported: 0 – No, 1 – Yes.
uint16_t	sid	CDMA system ID.
uint16_t	nid	CDMA network ID.
uint16_t	bsid	Base station ID.

3.3.5.1.6. struct mcm_nw_selection_t_v01

Data fields

Type	Parameter	Description
mcm_nw_-selection_type-_t_v01	nw_selection_-type	Network selection type.

Type	Parameter	Description
char	mcc	Mobile country code for a manual network selection.
char	mnc	Mobile network code for a manual network selection.
mcm_nw_-radio_tech_t_- v01	rat	Radio technology.

3.3.5.1.7. struct mcm_nw_gsm_sig_info_t_v01

Data fields

Type	Parameter	Description
int8_t	rssi	RSSI in dBm. Indicates received signal strength. A signed value; -125 or lower indicates no signal.

3.3.5.1.8. struct mcm_nw_wcdma_sig_info_t_v01

Data fields

Type	Parameter	Description
int8_t	rssi	RSSI in dBm. Indicates forward link pilot Ec. A signed value; -125 or lower indicates no signal.
int16_t	ecio	Ec/Io value representing negative 0.5 dB increments, e.g., 2 equals -1 dbm.

3.3.5.1.9. struct mcm_nw_tdscdma_sig_info_t_v01

Data fields

Type	Parameter	Description
int8_t	rssi	RSSI in dBm. Indicates forward link pilot Ec. a signed value; -125 or lower indicates no signal.
int8_t	rscp	RSCP in dBm.
int16_t	ecio	Ec/Io value representing negative 0.5 dB increments, e.g., 2 equals -1 dbm.
int8_t	sinr	Measured SINR in dB.

3.3.5.1.10. struct mcm_nw_lte_sig_info_t_v01

Data fields

Type	Parameter	Description
int8_t	rssi	RSSI in dBm. Indicates forward link pilot Ec. A signed value; -125 or lower indicates no signal.
int8_t	rsrq	RSRQ value in dB (signed integer value), as measured by L1. Range: -3 to -20 (-3 equals -3 dB, -20 equals -20 dB).
int8_t	rsrp	Current RSRP in dBm, as measured by L1. Range: -44 to -140 (-44 equals -44 dBm, -140 equals -140 dBm).
int8_t	snr	SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246.

3.3.5.1.11. struct mcm_nw_cdma_sig_info_t_v01

Data Fields

Type	Parameter	Description
int8_t	rssi	RSSI in dBm. Indicates forward link pilot Power (AGC) + Ec/Io. A signed value;of -125 or lower indicates no signal.
int16_t	ecio	Ec/Io value representing negative 0.5 dB increments, for example, 2 equals -1 dbm.

3.3.5.1.12. struct mcm_nw_hdr_sig_info_t_v01

Data Fields

Type	Parameter	Description
int8_t	rssi	RSSI in dBm. Indicates forward link pilot Power (AGC) + Ec/Io. A signed value; -125 or lower indicates no signal.
int16_t	ecio	Ec/Io value representing negative 0.5 dB increments, e.g., 2 equals -1 dbm.
int8_t	sinr	SINR level.
int32_t	io	Received IO in dBm.

3.3.5.1.13. struct mcm_nw_nr5g_sig_info_t_v01

Data Fields

Type	Parameter	Description
int16_t	rsrp	Current RSRP in dBm, as measured by L1. Range: -44 to -140 (-44 equals -44 dBm, -140 equals -140 dBm).
int16_t	snr	SNR level as a scaled integer in units of 0.1 dB; e.g., -16 dB has a value of -160 and 24.6 dB has a value of 246.

3.3.5.1.14. struct mcm_nw_nitz_time_t_v01

Data Fields

Type	Parameter	Description
Char	nitz_time	NITZ time.

3.4. Data Calls

This chapter describes the data calls processing, using MCM.

- Data Message Identifiers
- Data Message Structures
- Data Constants
- Data Enumerations

- Data Structures

3.4.1. Data Message Identifiers

This section contains the MCM data message identifiers.

```
#define MCM_DATA_START_DATA_CALL_REQ_V01 0x0100

#define MCM_DATA_START_DATA_CALL_RSP_V01 0x0100

#define MCM_DATA_STOP_DATA_CALL_REQ_V01 0x0101

#define MCM_DATA_STOP_DATA_CALL_RSP_V01 0x0101

#define MCM_DATA_GET_PKT_STATS_REQ_V01 0x0102

#define MCM_DATA_GET_PKT_STATS_RSP_V01 0x0102

#define MCM_DATA_RESET_PKT_STATS_REQ_V01 0x0103

#define MCM_DATA_RESET_PKT_STATS_RSP_V01 0x0103

#define MCM_DATA_GET_DEVICE_NAME_REQ_V01 0x0104

#define MCM_DATA_GET_DEVICE_NAME_RSP_V01 0x0104

#define MCM_DATA_GET_DEVICE_ADDR_COUNT_REQ_V01 0x0105

#define MCM_DATA_GET_DEVICE_ADDR_COUNT_RSP_V01 0x0105

#define MCM_DATA_GET_CALL_TECH_REQ_V01 0x0106

#define MCM_DATA_GET_CALL_TECH_RSP_V01 0x0106

#define MCM_DATA_GET_CALL_STATUS_REQ_V01 0x0107

#define MCM_DATA_GET_CALL_STATUS_RSP_V01 0x0107

#define MCM_DATA_GET_DEVICE_ADDR_REQ_V01 0x0108

#define MCM_DATA_GET_DEVICE_ADDR_RSP_V01 0x0108

#define MCM_DATA_GET_CHANNEL_RATE_REQ_MSG_V01 0x0109

#define MCM_DATA_GET_CHANNEL_RATE_RSP_MSG_V01 0x0109

#define MCM_DATA_EVENT_REGISTER_REQ_V01 0x010A

#define MCM_DATA_EVENT_REGISTER_RESP_V01 0x010A

#define MCM_DATA_GET_REG_STATUS_REQ_MSG_V01 0x010B

#define MCM_DATA_GET_REG_STATUS_RSP_MSG_V01 0x010B

#define MCM_DATA_UNSOL_EVENT_IND_V01 0x010C
```

3.4.2. Data Message Structures

This section contains the MCM data message structures.

3.4.2.1. Data Structure Documentation

3.4.2.1.1. struct mcm_data_start_data_call_req_msg_v01

Request message; Sends a request to start a data call for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
uint8_t	ip_family_valid	Must be set to TRUE if ip_family is being passed.
int8_t	ip_family	IP family requested: 4 – IPv4 6 – IPv6 10 – IPv4/v6
uint8_t	apn_name_-valid	Must be set to TRUE if apn_name is being passed.
char	apn_name	APN name requested. A character string that identifies a PDN, as defined by the operator.
uint8_t	user_name_-valid	Must be set to TRUE if user_name is being passed.
char	user_name	Username for the APN.
uint8_t	password_valid	Must be set to TRUE if password is being passed.
char	password	Password for the APN.
uint8_t	tech_pref_valid	Must be set to TRUE if tech_pref is being passed.
int8_t	tech_pref	Technology preference: 0 – CDMA 1 – UMTS
uint8_t	umts_profile_-valid	Must be set to TRUE if umts_profile is being passed.
int8_t	umts_profile	UMTS/LTE profile ID.
uint8_t	cdma_profile_-valid	Must be set to TRUE if a CDMA_profile is being passed.
int8_t	cdma_profile	CDMA profile ID.

3.4.2.1.2. struct mcm_data_start_data_call_rsp_msg_v01

Response message; Sends a request to start a data call for the connection associated with the specified call ID.

Data Fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

Type	Parameter	Description
mcm_data_call-_status_t_v01	call_status	MCM data call status.
uint8_t	call_id_valid	Must be set to TRUE if call_id is being passed.
int32_t	call_id	Call ID that gets generated for a successful call.
uint8_t	vce_reason_-valid	Must be set to TRUE if vce_reason is being passed.
mcm_data_-verbose_call_-end_reason_t_- v01	vce_reason	Call end reason in verbose.

3.4.2.1.3. struct mcm_data_stop_data_call_req_msg_v01

Request message; Sends a request to stop a data call for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the call to be stopped.

3.4.2.1.4. struct mcm_data_stop_data_call_rsp_msg_v01

Response message; Sends a request to stop a data call for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Response.

3.4.2.1.5. struct mcm_data_get_pkt_stats_req_msg_v01

Request message; Gets packet statistics for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to get the packet statistics.

3.4.2.1.6. struct mcm_data_get_pkt_stats_rsp_msg_v01

Response message; Gets packet statistics for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Response.
uint8_t	pkt_stats_valid	Must be set to TRUE if pkt_stats is being passed.
mcm_data_pkt-_stats_t_v01	pkt_stats	Packet statistics.

3.4.2.1.7. struct mcm_data_reset_pkt_stats_req_msg_v01

Request message; Resets packet statistics for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to reset the packet statistics.

3.4.2.1.8. struct mcm_data_get_device_name_req_msg_v01

Request message; Gets routing information for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to get the routing information.

3.4.2.1.9. struct mcm_data_get_device_name_rsp_msg_v01

Response message; Gets routing information for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Response.
uint8_t	device_name_-valid	Must be set to TRUE if device_name is being passed.
uint32_t	device_name_-len	Must be set to the number of elements in device_name.
char	device_name	Device name.

3.4.2.1.10. struct mcm_data_get_device_addr_count_req_msg_v01

Request message; Gets the number of IP addresses available for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to get the address count.

3.4.2.1.11. struct mcm_data_get_device_addr_count_rsp_msg_v01

Response message; Gets the number of IP addresses available for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to get the address count.

3.4.2.1.12. struct mcm_data_get_call_tech_req_msg_v01

Request message; Gets underlying technology available for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to get the call technology.

3.4.2.1.13. struct mcm_data_get_call_tech_rsp_msg_v01

Response message; Gets the underlying technology available for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Response.
uint8_t	call_tech_valid	Must be set to TRUE if call_tech is being passed.
mcm_data_-bearer_tech_-info_t_v01	call_tech	Data call bearer technology.

3.4.2.1.14. struct mcm_data_get_call_status_req_msg_v01

Request message; Gets the current status of the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to get the call status.

3.4.2.1.15. struct mcm_data_get_call_status_rsp_msg_v01

Response message; Gets the current status of the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
mcm_response_t_v01	resp	Response.
uint8_t	call_status_valid	Must be set to TRUE if call_status is being passed.
mcm_data_call_status_t_v01	call_status	Data call status.

3.4.2.1.16. struct mcm_data_get_reg_status_rsp_msg_v01

Response message; Gets the current data registration status.

Data Fields

Type	Parameter	Description
mcm_response_t_v01	resp	Response.
uint8_t	reg_status_valid	Must be set to TRUE if reg_status is being passed.
mcm_data_reg_status_t_v01	reg_status	Data modem registration status.

3.4.2.1.17. struct mcm_data_get_device_addr_req_msg_v01

Request message; Gets a list of IP address for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to get the IP address.

3.4.2.1.18. struct mcm_data_get_device_addr_rsp_msg_v01

Response message; Gets a list of IP address for the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Response.
uint8_t	addr_info_valid	Must be set to TRUE if addr_info is being passed.
uint32_t	addr_info_len	Must be set to the number of elements in addr_info.
mcm_data_-addr_t_info_- v01	addr_info	Data device address.

3.4.2.1.19. struct mcm_data_get_channel_rate_req_msg_v01

Request message; Gets the current and maximum channel rate of the connection associated with the specified call ID.

Data fields

Type	Parameter	Description
int32_t	call_id	Call ID of the connection for which to get the channel rate.

3.4.2.1.20. struct mcm_data_get_channel_rate_rsp_msg_v01

Response message; Gets the current and maximum channel rate of connection associated with the specified call ID.

Data Fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Response.
uint8_t	channel_rate_-valid	Must be set to TRUE if channel_rate is being passed
mcm_data_-channel_rate_t-_v01	channel_rate	Data channel rate.

3.4.2.1.21. struct mcm_data_event_register_req_msg_v01

Request message; Registers for an indication of events

Data fields

Type	Parameter	Description
uint8_t	register_net_-up_event_valid	Must be set to TRUE if register_net_up_event is being passed.
uint8_t	register_net_-up_event	Register for a network-up event.
uint8_t	register_net_-down_event_- valid	Must be set to TRUE if register_net_down_event is being passed.
uint8_t	register_net_-down_event	Register for a network down event.
uint8_t	register_net_-new_addr_-event_valid	Must be set to TRUE if register_net_new_addr_event is being passed.
uint8_t	register_net_-new_addr_- event	Register for a new address event.
uint8_t	register_net_-del_addr_event-_valid	Must be set to TRUE if register_net_del_addr_event is being passed.
uint8_t	register_net_-del_addr_event	Register for a delete address event.
uint8_t	register_reg_-srvc_status_-event_valid	Must be set to TRUE if register_reg_srvc_status_event is being passed.
uint8_t	register_reg_-srvc_status_- event	Register for a service status event.
uint8_t	register_bearer-_tech_status_-event_valid	Must be set to TRUE if register_bearer_tech_status_event is being passed.
uint8_t	register_bearer-_tech_status_-event	Register for a bearer technology status event.
uint8_t	register_-dormancy_-status_event_-valid	Must be set to TRUE if register_dormancy_status_event is being passed.
uint8_t	register_-dormancy_-status_event	Register for a dormancy status event.

3.4.2.1.22. struct mcm_data_event_register_resp_msg_v01

Response message; Registers for an indication of events.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Response.

3.4.2.1.23. struct mcm_data_unsol_event_ind_msg_v01

Indication message; Indication corresponding to an unsolicited event.

Data fields

Type	Parameter	Description
int32_t	event_id	Event ID: 0x00005001 – MCM_DATA_NET_UP_EV 0x00005002 – MCM_DATA_NET_DOWN_EV 0x00005003 – MCM_DATA_NET_NEW_ADDR_EV 0x00005004 – MCM_DATA_NET_DEL_ADDR_EV 0x00005005 – MCM_DATA_REG_SRVC_STATUS_EV 0x00005006 – MCM_DATA_BEARER_TECH_STATUS_EV 0x00005007 – MCM_DATA_DORMANCY_STATUS_EV
uint8_t	call_id_valid	Must be set to TRUE if call_id is being passed.
int32_t	call_id	Call ID that gets generated for a successful call.
uint8_t	call_status_-valid	Must be set to TRUE if call_status is being passed.
mcm_data_call-_status_t_v01	call_status	Data call status.
uint8_t	call_tech_valid	Must be set to TRUE if call_tech is being passed.
mcm_data_-bearer_tech_-info_t_v01	call_tech	Data bearer technology corresponding to the call ID.
uint8_t	reg_status_-valid	Must be set to TRUE if reg_status is being passed.
mcm_data_reg-_status_t_v01	reg_status	Data modem registration status.
uint8_t	dorm_status_-valid	Must be set to TRUE if dorm_status is being passed.
mcm_data_-dormancy_-state_t_v01	dorm_status	Data dormancy status.
uint8_t	addr_count_-valid	Must be set to TRUE if addr_count is being passed.
int8_t	addr_count	Data device address count.
uint8_t	addr_info_valid	Must be set to TRUE if addr_info is being passed.
uint32_t	addr_info_len	Must be set to the number of elements in addr_info.
mcm_data_-addr_t_info_- v01	addr_info	Data device address.
uint8_t	vce_reason_-valid	Must be set to TRUE if vce_reason is being passed.
mcm_data_-verbose_call_-end_reason_t_- v01	vce_reason	Data call end reason in verbose.

3.4.3. Data Constants

This section contains the MCM data constants.

3.4.3.1. Define Documentation

#define MCM_DATA_NET_UP_EV_V01 0x00005001 Event ID indicating that a data call is connected.

#define MCM_DATA_NET_DOWN_EV_V01 0x00005002 Event ID indicating that a data call is disconnected.

#define MCM_DATA_NET_NEW_ADDR_EV_V01 0x00005003 Event ID indicating that a new IP address is configured for the interface.

#define MCM_DATA_NET_DEL_ADDR_EV_V01 0x00005004 Event ID indicating that one of the IP addresses is lost.

#define MCM_DATA_REG_SRVC_STATUS_EV_V01 0x00005005 Event ID indicating the current service status of the modem.

#define MCM_DATA_BEARER_TECH_STATUS_EV_V01 0x00005006 Event ID indicating the current bearer technology of the call.

#define MCM_DATA_DORMANCY_STATUS_EV_V01 0x00005007 Event ID indicating the dormancy state of the call.

#define MCM_DATA_MAX_APN_LEN_V01 150 Maximum length of the APN.

#define MCM_DATA_MAX_USERNAME_LEN_V01 127 Maximum length of the profile user name.

#define MCM_DATA_MAX_PASSWORD_LEN_V01 127 Maximum length of the password.

#define MCM_DATA_MAX_ADDR_LEN_V01 128 Maximum address length.

#define MCM_DATA_MAX_DEVICE_NAME_LEN_V01 13 Maximum length of the device name.

#define MCM_DATA_MAX_ADDR_COUNT_V01 10 Maximum number of IP addresses.

3.4.4. Data Enumerations

This section contains the MCM data enums.

3.4.4.1. Enumeration Type Documentation

3.4.4.1.1. enum mcm_data_call_status_t_v01

Enumerator:

MCM_DATA_CALL_STATE_INVALID_V01 Call state is invalid.

MCM_DATA_CALL_STATE_CONNECTING_V01 Call is connecting.

MCM_DATA_CALL_STATE_CONNECTED_V01 Call is connected.

MCM_DATA_CALL_STATE_DISCONNECTING_V01 Call is disconnecting.

MCM_DATA_CALL_STATE_DISCONNECTED_V01 Call is disconnected.

3.4.4.1.2. enum mcm_data_srv_status_t_v01

Enumerator:

MCM_DATA_MODEM_STATE_OOS_V01 Modem is out of service.

MCM_DATA_MODEM_STATE_IN_SERVICE_V01 Modem is in service.

3.4.4.1.3. enum mcm_data_bearer_tech_info_t_v01

Enumerator:

MCM_DATA_BEARER_TECH_TYPE_UNKNOWN_V01 Unknown technology.

MCM_DATA_BEARER_TECH_TYPE_CDMA_1X_V01 1X technology.

MCM_DATA_BEARER_TECH_TYPE_EVDO_REV0_V01 CDMA Rev 0.

MCM_DATA_BEARER_TECH_TYPE_EVDO_REVA_V01 CDMA Rev A.

MCM_DATA_BEARER_TECH_TYPE_EVDO_REVB_V01 CDMA Rev B.

MCM_DATA_BEARER_TECH_TYPE_EHRPD_V01 EHRPD.

MCM_DATA_BEARER_TECH_TYPE_FMC_V01 Fixed mobile convergence.

MCM_DATA_BEARER_TECH_TYPE_HRPD_V01 HRPD.

MCM_DATA_BEARER_TECH_TYPE_3GPP2_WLAN_V01 IWLAN.

MCM_DATA_BEARER_TECH_TYPE_WCDMA_V01 WCDMA.

MCM_DATA_BEARER_TECH_TYPE_GPRS_V01 GPRS.

MCM_DATA_BEARER_TECH_TYPE_HSDPA_V01 HSDPA.

MCM_DATA_BEARER_TECH_TYPE_HSUPA_V01 HSUPA.

MCM_DATA_BEARER_TECH_TYPE_EDGE_V01 EDGE.

MCM_DATA_BEARER_TECH_TYPE_LTE_V01 LTE.

MCM_DATA_BEARER_TECH_TYPE_HSDPA_PLUS_V01 HSDPA+.

MCM_DATA_BEARER_TECH_TYPE_DC_HSDPA_PLUS_V01

MCM_DATA_BEARER_TECH_TYPE_HSPA_V01 HSPA.

MCM_DATA_BEARER_TECH_TYPE_64_QAM_V01 64 QAM.

MCM_DATA_BEARER_TECH_TYPE_TDSCDMA_V01 TD-SCDMA.

MCM_DATA_BEARER_TECH_TYPE_GSM_V01 GSM.

MCM_DATA_BEARER_TECH_TYPE_3GPP_WLAN_V01 IWLAN.

3.4.4.1.4. enum mcm_data_dormancy_state_t_v01

Enumerator:

MCM_DATA_DORMANCY_STATE_PHYSLINK_ACTIVE_V01 Call is not dormant.
MCM_DATA_DORMANCY_STATE_PHYSLINK_DORMANT_V01 Call is dormant.

3.4.4.1.5. enum mcm_data_call_end_reason_type_t_v01

Enumerator:

MCM_DATA_TYPE_UNSPECIFIED_V01 Unspecified.

MCM_DATA_TYPE_MOBILE_IP_V01

MCM_DATA_TYPE_INTERNAL_V01

MCM_DATA_TYPE_CALL_MANAGER_DEFINED_V01 Call manager-defined.

MCM_DATA_TYPE_3GPP_SPEC_DEFINED_V01 3GPP specification-defined.

MCM_DATA_TYPE_PPP_V01

MCM_DATA_TYPE_EHRPD_V01 EHRPD.

MCM_DATA_TYPE_IPV6_V01 IPV6.

3.4.4.1.6. enum mcm_data_call_end_reason_code_t_v01

Enumerator:

MCM_DATA_CE_INVALID_V01

MCM_DATA_CE_MIP_FA_ERR_REASON_UNSPECIFIED_V01
Mobile IP; unspecified error.

MCM_DATA_CE_MIP_FA_ERR_ADMINISTRATIVELY_PROHIBITED_V01
Mobile IP; administratively prohibited.

MCM_DATA_CE_MIP_FA_ERR_INSUFFICIENT_RESOURCES_V01
Mobile IP; insufficient resources.

MCM_DATA_CE_MIP_FA_ERR_MOBILE_NODE_AUTHENTICATION_FAILURE_V01
Mobile IP; mobile node authentication failure.

MCM_DATA_CE_MIP_FA_ERR_HA_AUTHENTICATION_FAILURE_V01
Mobile IP; HA authentication failure.

MCM_DATA_CE_MIP_FA_ERR_REQUESTED_LIFETIME_TOO_LONG_V01
Mobile IP; requested lifetime is too long.

MCM_DATA_CE_MIP_FA_ERR_MALFORMED_REQUEST_V01
Mobile IP; malformed request.

MCM_DATA_CE_MIP_FA_ERR_MALFORMED_REPLY_V01
Mobile IP; malformed reply.

MCM_DATA_CE_MIP_FA_ERR_ENCAPSULATION_UNAVAILABLE_V01
Mobile IP; encapsulation is unavailable.

MCM_DATA_CE_MIP_FA_ERR_VJHC_UNAVAILABLE_V01
Mobile IP; VJHC is unavailable.

MCM_DATA_CE_MIP_FA_ERR_REVERSE_TUNNEL_UNAVAILABLE_V01 Mobile IP; reverse tunnel is unavailable.

MCM_DATA_CE_MIP_FA_ERR_REVERSE_TUNNEL_IS_MANDATORY_AND_T_BIT_NOT_SET_V01
Mobile IP; the reverse tunnel is mandatory and the T-bit is not set.

MCM_DATA_CE_MIP_FA_ERR_DELIVERY_STYLE_NOT_SUPPORTED_V01
Mobile IP; delivery style is not supported.

MCM_DATA_CE_MIP_FA_ERR_MISSING_NAI_V01
Mobile IP; missing NAI.

MCM_DATA_CE_MIP_FA_ERR_MISSING_HA_V01
Mobile IP; missing HA.

MCM_DATA_CE_MIP_FA_ERR_MISSING_HOME_ADDR_V01
Mobile IP; missing home address.

MCM_DATA_CE_MIP_FA_ERR_UNKNOWN_CHALLENGE_V01
Mobile IP; unknown challenge.

CM_DATA_CE_MIP_FA_ERR_MISSING_CHALLENGE_V01
Mobile IP; missing challenge.

CM_DATA_CE_MIP_FA_ERR_STALE_CHALLENGE_V01
Mobile IP; stale challenge.

MCM_DATA_CE_MIP_HA_ERR_REASON_UNSPECIFIED_V01
Mobile IP; the reason is unspecified.

MCM_DATA_CE_MIP_HA_ERR_ADMINISTRATIVELY_PROHIBITED_V01
Mobile IP;administratively prohibited.

MCM_DATA_CE_MIP_HA_ERR_INSUFFICIENT_RESOURCES_V01
Mobile IP; insufficient resources.

CM_DATA_CE_MIP_HA_ERR_MOBILE_NODE_AUTHENTICATION_FAILURE_V01 Mobile IP; mobile node authentication failure.

MCM_DATA_CE_MIP_HA_ERR_FA_AUTHENTICATION_FAILURE_V01
Mobile IP; FA authentication failure.

MCM_DATA_CE_MIP_HA_ERR_REGISTRATION_ID_MISMATCH_V01
Mobile IP; registration ID mismatch.

MCM_DATA_CE_MIP_HA_ERR_MALFORMED_REQUEST_V01
Mobile IP; malformed request.

CM_DATA_CE_MIP_HA_ERR_UNKNOWN_HA_ADDR_V01
Mobile IP; unknown HA address.

CM_DATA_CE_MIP_HA_ERR_REVERSE_TUNNEL_UNAVAILABLE_V01
Mobile IP; reverse tunnel is unavailable.

MCM_DATA_CE_MIP_HA_ERR_REVERSE_TUNNEL_IS_MANDATORY_AND_T_BIT_NOT_SET_V01
Mobile IP; reverse tunnel is mandatory and the T-bit is not set.

MCM_DATA_CE_MIP_HA_ERR_ENCAPSULATION_UNAVAILABLE_V01
Mobile IP; encapsulation is unavailable.

MCM_DATA_CE_MIP_ERR_REASON_UNKNOWN_V01

MCM_DATA_CE_INTERNAL_ERROR_V01

MCM_DATA_CE_CALL_ENDED_V01

MCM_DATA_CE_INTERNAL_UNKNOWN_CAUSE_CODE_V01
Internal error; internal unknown cause code.

MCM_DATA_CE_UNKNOWN_CAUSE_CODE_V01
Internal error; unknown cause code.

MCM_DATA_CE_CLOSE_IN_PROGRESS_V01
Internal error; close in progress.

MCM_DATA_CE_NW_INITIATED_TERMINATION_V01
Internal error; NW-initiated termination.

MCM_DATA_CE_APP_PREEMPTED_V01
Internal error; the application was preempted.

MCM_DATA_CE_CDMA_LOCK_V01
CDMA; CDMA lock.

MCM_DATA_CE_INTERCEPT_V01
CDMA; intercept.

MCM_DATA_CE_REORDER_V01
CDMA; reorder.

MCM_DATA_CE_REL_SO_REJ_V01
CDMA; release SO was rejected.

MCM_DATA_CE_INCOM_CALL_V01
CDMA; incoming call.

MCM_DATA_CE_ALERT_STOP_V01
CDMA; alert stop.

MCM_DATA_CE_ACTIVATION_V01
CDMA; activation.

MCM_DATA_CE_MAX_ACCESS_PROBE_V01
CDMA; maximum access probe.

MCM_DATA_CE_CCS_NOT_SUPPORTED_BY_BS_V01
CDMA; CCS is not supported by the base station.

MCM_DATA_CE_NO_RESPONSE_FROM_BS_V01
CDMA; no response from the base station.

MCM_DATA_CE_REJECTED_BY_BS_V01
CDMA; rejected by the base station.

MCM_DATA_CE_INCOMPATIBLE_V01
CDMA; incompatible.

MCM_DATA_CE_ALREADY_IN_TC_V01
CDMA; already in TC.

MCM_DATA_CE_USER_CALL_ORIG_DURING_GPS_V01
CDMA; user call originated during GPS.

MCM_DATA_CE_USER_CALL_ORIG_DURING_SMS_V01
CDMA; user call originated during SMS.
MCM_DATA_CE_NO_CDMA_SRV_V01
CDMA; no CDMA service.

MCM_DATA_CE_CONF_FAILED_V01
CDMA; confirmation failed.

MCM_DATA_CE_INCOM_REJ_V01
CDMA; incoming call was rejected.

MCM_DATA_CE_NO_GW_SRV_V01
CDMA; no GW service.

MCM_DATA_CE_NO_GPRS_CONTEXT_V01
CDMA; no GPRS context.

MCM_DATA_CE_ILLEGAL_MS_V01
CDMA; illegal MS.

MCM_DATA_CE_ILLEGAL_ME_V01
CDMA; illegal ME.

MCM_DATA_CE_GPRS_SERVICES_AND_NON_GPRS_SERVICES_NOT_ALLOWED_V01
CDMA; GPRS services and non-GPRS services are not allowed.

MCM_DATA_CE_GPRS_SERVICES_NOT_ALLOWED_V01
CDMA; GPRS services are not allowed.

MCM_DATA_CE_MS_IDENTITY_CANNOT_BE_DERIVED_BY_THE_NETWORK_V01
CDMA; MS identity cannot be derived by the network.

MCM_DATA_CE_IMPLICITLY_DETACHED_V01

MCM_DATA_CE_PLMN_NOT_ALLOWED_V01

MCM_DATA_CE_LA_NOT_ALLOWED_V01

MCM_DATA_CE_GPRS_SERVICES_NOT_ALLOWED_IN_THIS_PLMN_V01
CDMA; GPRS services are not allowed in this PLMN.

MCM_DATA_CE_PDP_DUPLICATE_V01
CDMA; PDP duplicate.

MCM_DATA_CE_UE_RAT_CHANGE_V01
CDMA; UE RAT change.

MCM_DATA_CE_CONGESTION_V01
CDMA; congestion.

MCM_DATA_CE_NO_PDP_CONTEXT_ACTIVATED_V01
CDMA; no PDP context is activated.

MCM_DATA_CE_ACCESS_CLASS_DSAC_REJECTION_V01
CDMA; access class DSAC rejection.

MCM_DATA_CE_CD_GEN_OR_BUSY_V01
CDMA; CD is generating or busy.

MCM_DATA_CE_CD_BILL_OR_AUTH_V01
CDMA; CD bill or authorization.

MCM_DATA_CE_CHG_HDR_V01
CDMA; change HDR.
MCM_DATA_CE_EXIT_HDR_V01
CDMA; exit HDR.

MCM_DATA_CE_HDR_NO_SESSION_V01
CDMA; HDR no session.

MCM_DATA_CE_HDR_ORIG_DURING_GPS_FIX_V01
CDMA; HDR originated during a GPS fix.

MCM_DATA_CE_HDR_CS_TIMEOUT_V01
CDMA; HDR CS timeout.

MCM_DATA_CE_HDR_RELEASED_BY_CM_V01
CDMA; HDR released by the CM.

MCM_DATA_CE_CLIENT_END_V01
CDMA; client end.

MCM_DATA_CE_NO_SRV_V01
CDMA; no service.

MCM_DATA_CE_FADE_V01
CDMA; fade.

MCM_DATA_CE_REL_NORMAL_V01
CDMA; release is normal.

MCM_DATA_CE_ACC_IN_PROG_V01
CDMA; access is in progress.

MCM_DATA_CE_ACC_FAIL_V01
CDMA; access failure.

MCM_DATA_CE_REDIR_OR_HANDOFF_V01
CDMA; redirect or handoff.

MCM_DATA_CE_OPERATOR_DETERMINED_BARRING_V01
3GPP Spec defined; operator determined barring.

MCM_DATA_CE_LLC_SNDCP_FAILURE_V01
3GPP Spec defined; LLC SNDCP failure.

MCM_DATA_CE_INSUFFICIENT_RESOURCES_V01
3GPP Spec defined; insufficient resources.

MCM_DATA_CE_UNKNOWN_APN_V01
3GPP Spec defined; unknown APN.

MCM_DATA_CE_UNKNOWN_PDP_V01
3GPP Spec defined; unknown PDP.

MCM_DATA_CE_AUTH_FAILED_V01
3GPP Spec defined; authorization failed.

MCM_DATA_CE_GGSN_REJECT_V01
3GPP Spec defined; GGSN was rejected.

MCM_DATA_CE_ACTIVATION_REJECT_V01
3GPP Spec defined; activation was rejected.

MCM_DATA_CE_OPTION_NOT_SUPPORTED_V01
3GPP Spec defined; option is not supported.
MCM_DATA_CE_OPTION_UNSUBSCRIBED_V01
3GPP Spec defined; option is unsubscribed.

MCM_DATA_CE_OPTION_TEMP_OOO_V01
3GPP Spec defined; option is temporarily out of operation.

MCM_DATA_CE_NSAPI_ALREADY_USED_V01
3GPP Spec defined; NSAPI was already used.

MCM_DATA_CE_REGULAR_DEACTIVATION_V01
3GPP Spec defined; regular deactivation.

MCM_DATA_CE_QOS_NOT_ACCEPTED_V01
3GPP Spec defined; QoS was not accepted.

MCM_DATA_CE_NETWORK_FAILURE_V01
3GPP Spec defined; network failure.

MCM_DATA_CE_UMTS_REACTIVATION_REQ_V01
3GPP Spec defined; UMTS reactivation is required.

MCM_DATA_CE_FEATURE_NOT_SUPPORTED_V01
3GPP Spec defined; feature is not supported.

MCM_DATA_CE_TFT_SEMANTIC_ERROR_V01
3GPP Spec defined; TFT semantic error.

MCM_DATA_CE_TFT_SYNTAX_ERROR_V01
3GPP Spec defined; TFT syntax error.

MCM_DATA_CE_UNKNOWN_PDP_CONTEXT_V01
3GPP Spec defined; unknown PDP context.

MCM_DATA_CE_FILTER_SEMANTIC_ERROR_V01
3GPP Spec defined; filter semantic error.

MCM_DATA_CE_FILTER_SYNTAX_ERROR_V01
3GPP Spec defined; filter syntax error.

MCM_DATA_CE_PDP_WITHOUT_ACTIVE_TFT_V01
3GPP Spec defined; PDP is without an active TFT.

MCM_DATA_CE_IP_V4_ONLY_ALLOWED_V01
3GPP Spec defined; only IPv4 is allowed.

MCM_DATA_CE_IP_V6_ONLY_ALLOWED_V01
3GPP Spec defined; only IPv6 is allowed.

MCM_DATA_CE_SINGLE_ADDR_BEARER_ONLY_V01
3GPP Spec defined; single address bearer only.

MCM_DATA_CE_INVALID_TRANSACTION_ID_V01
3GPP Spec defined; invalid transaction ID.

MCM_DATA_CE_MESSAGE_INCORRECT_SEMANTIC_V01
3GPP Spec defined; message has incorrect semantic.

MCM_DATA_CE_INVALID_MANDATORY_INFO_V01
3GPP Spec defined; invalid mandatory information.

MCM_DATA_CE_MESSAGE_TYPE_UNSUPPORTED_V01
3GPP Spec defined; message type is unsupported.
MCM_DATA_CE_MSG_TYPE_NONCOMPATIBLE_STATE_V01
3GPP Spec defined; message type is in a non-compatible state.

MCM_DATA_CE_UNKNOWN_INFO_ELEMENT_V01
3GPP Spec defined; unknown information element.

MCM_DATA_CE_CONDITIONAL_IE_ERROR_V01
3GPP Spec defined; conditional IE error.

MCM_DATA_CE_MSG_AND_PROTOCOL_STATE_UNCOMPATIBLE_V01
3GPP Spec defined;message and protocol state are incompatible.

MCM_DATA_CE_PROTOCOL_ERROR_V01
3GPP Spec defined; protocol error.

MCM_DATA_CE_APN_TYPE_CONFLICT_V01
3GPP Spec defined; APN type conflict.

MCM_DATA_CE_PPP_TIMEOUT_V01
PPP; timeout.

MCM_DATA_CE_PPP_AUTH_FAILURE_V01
PPP; authorization failure.

MCM_DATA_CE_PPP_OPTION_MISMATCH_V01
PPP; option mismatch.

MCM_DATA_CE_PPP_PAP_FAILURE_V01
PPP; PAP failure.

MCM_DATA_CE_PPP_CHAP_FAILURE_V01
PPP; CHAP failure.

MCM_DATA_CE_PPP_UNKNOWN_V01
PPP; unknown.

MCM_DATA_CE_EHRPD_SUBS_LIMITED_TO_V4_V01
EHRPD; subscription is limited to v4.

MCM_DATA_CE_EHRPD_SUBS_LIMITED_TO_V6_V01
EHRPD; subscription is limited to v6.

MCM_DATA_CE_EHRPD_VSNCP_TIMEOUT_V01
EHRPD VSNCP; timeout.

MCM_DATA_CE_EHRPD_VSNCP_FAILURE_V01
EHRPD VSNCP; failure.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_GEN_ERROR_V01
EHRPD VSNCP 3GPP2I; generation error.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_UNAUTH_APN_V01
EHRPD VSNCP 3GPP2I; unauthorized APN.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_PDN_LIMIT_EXCEED_V01
EHRPD VSNCP 3GPP2I; PDM limit was exceeded.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_NO_PDN_GW_V01
EHRPD VSNCP 3GPP2I; no PDN GW.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_PDN_GW_UNREACH_V01
EHRPD VSNCP 3GPP2I; PDN GW is unreachable.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_PDN_GW_REJ_V01
EHRPD VSNCP 3GPP2I; PDN GW was rejected.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_INSUFF_PARAM_V01
EHRPD VSNCP 3GPP2I; insufficient parameters.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_RESOURCE_UNAVAIL_V01
EHRPD VSNCP 3GPP2I; resource is unavailable.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_ADMIN_PROHIBIT_V01
EHRPD VSNCP 3GPP2I; administratively prohibited.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_PDN_ID_IN_USE_V01
EHRPD VSNCP 3GPP2I; PDN ID is in use.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_SUBSCR_LIMITATION_V01
EHRPD VSNCP 3GPP2I; subscriber limitation.

MCM_DATA_CE_EHRPD_VSNCP_3GPP2I_PDN_EXISTS_FOR_THIS_APN_V01 EHRPD VSNCP 3GPP2I; PDN exists for this APN.

MCM_DATA_CE_PREFIX_UNAVAILABLE_V01
IPv6; prefix is unavailable.

MCM_DATA_CE_IPV6_ERR_HRPD_IPV6_DISABLED_V01
IPv6; HRPD IPv6 is disabled.

3.4.5. Data Structures

This section contains the MCM data structures.

3.4.5.1. Data Structure Documentation

3.4.5.1.1. struct mcm_data_reg_status_t_v01

Data fields

Type	Parameter	Description
mcm_data_srv-_status_t_v01	srv_status	Identifies the service state of the modem.
mcm_data_-bearer_tech_-info_t_v01	tech_info	Identifies the preferred technology type.

3.4.5.1.2. struct mcm_data_pkt_stats_t_v01

Data fields

Type	Parameter	Description
uint32_t	pkts_tx	Number of packets transmitted.
uint32_t	pkts_rx	Number of packets received.
uint64_t	bytes_tx	Number of bytes transmitted.
uint64_t	bytes_rx	Number of bytes received.
uint32_t	pkts_dropped_-tx	Number of transmit packets dropped.
uint32_t	pkts_dropped_-rx	Number of receive packets dropped.

3.4.5.1.3. struct mcm_data_channel_rate_t_v01

Data fields

Type	Parameter	Description
uint32_t	current_tx_rate	Current Tx data rate for the channel.
uint32_t	current_rx_rate	Current Rx data rate for the channel.
uint32_t	max_tx_rate	Maximum Tx data rate for the channel.
uint32_t	max_rx_rate	Maximum Rx data rate for the channel.

3.4.5.1.4. struct mcm_data_addr_t_v01

Data fields

Type	Parameter	Description
char	valid_addr	Indicates whether a valid address is available.
uint8_t	addr	Stores the IP address.

3.4.5.1.5. struct mcm_data_addr_t_info_v01

Data fields

Type	Parameter	Description
mcm_data_-addr_t_v01	iface_addr_s	Network interface address.
uint32_t	iface_mask	Subnet mask.
mcm_data_-addr_t_v01	gtwy_addr_s	Gateway server address.
uint32_t	gtwy_mask	Gateway mask.
mcm_data_-addr_t_v01	dnsp_addr_s	Primary DNS server address.
mcm_data_-addr_t_v01	dnss_addr_s	Secondary DNS server address.

3.4.5.1.6. struct mcm_data_verbose_call_end_reason_t_v01

Data fields

Type	Parameter	Description
mcm_data_call-_end_reason_-type_t_v01	call_end_-reason_type	Call end reason type. Values: 0 – Unspecified 1 – Mobile IP 2 – Internal 3 – Call manager-defined 6 – 3GPP specification-defined 7 – PPP 8 – EHRPD 9 – IPv6
mcm_data_call-_end_reason_-code_t_v01	call_end_-reason_code	Verbose data call end reason.

3.5. SMS

This section describes the functions and events for managing the mobile wireless Simple Messaging Service (SMS) for the device, using MCM.

- SMS Message Identifiers
- SMS Message Structures
- SMS Constants
- SMS Enumerations
- SMS Data Structures

3.5.1. SMS Message Identifiers

This section contains the MCM SMS message identifiers.

- #define MCM_SMS_SET_SERVICE_CENTER_CFG_TYPE_REQ_V01 0x0700
- #define MCM_SMS_SET_SERVICE_CENTER_CFG_TYPE_RESP_V01 0x0700
- #define MCM_SMS_GET_SERVICE_CENTER_CFG_TYPE_REQ_V01 0x0701
- #define MCM_SMS_GET_SERVICE_CENTER_CFG_TYPE_RESP_V01 0x0701
- #define MCM_SMS_SEND_MO_MSG_REQ_V01 0x0702
- #define MCM_SMS_SEND_MO_MSG_RESP_V01 0x0702
- #define MCM_SMS_SET_MSG_CONFIG_REQ_V01 0x0703
- #define MCM_SMS_SET_MSG_CONFIG_RESP_V01 0x0703
- #define MCM_SMS_GET_MSG_CONFIG_REQ_V01 0x0704
- #define MCM_SMS_GET_MSG_CONFIG_RESP_V01 0x0704
- #define MCM_SMS_SET_RECEPTION_MODE_REQ_V01 0x0705
- #define MCM_SMS_SET_RECEPTION_MODE_RESP_V01 0x0705
- #define MCM_SMS_EVENT_REGISTER_REQ_V01 0x0706
- #define MCM_SMS_EVENT_REGISTER_RESP_V01 0x0706
- #define MCM_SMS_PP_IND_V01 0x0707
- #define MCM_SMS_CB_IND_V01 0x0708
- #define MCM_SMS_CB_CMAS_IND_V01 0x0709

3.5.2. SMS Message Structures

This section contains the MCM SMS message structures.

3.5.2.1. Data Structure Documentation

3.5.2.1.1. struct mcm_sms_set_service_center_cfg_type_req_msg_v01

Request message; Sets the service center configuration type.

Data fields

Type	Parameter	Description
char	service_center-_addr	Address of the service center.
uint8_t	validity_time_-valid	Must be set to TRUE if validity_time is being passed.
int64_t	validity_time	Validity time.

3.5.2.1.2. struct mcm_sms_set_service_center_cfg_type_resp_msg_v01

Response message; Sets the service center configuration type.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

struct mcm_sms_get_service_center_cfg_type_resp_msg_v01

Response message; Gets the service center configuration type.

Data fields

Type	Parameter	Description
mcm_sms_-msg_format_t_-v01	message_-format	Message format.
char	message_-content	Message content.
char	destination	Destination.
uint8_t	size_validation-_valid	Must be set to TRUE if size_validation is being passed.
mcm_sms_-msg_size_-validation_-mode_t_v01	size_validation	Size validation.

3.5.2.1.3. struct mcm_sms_send_mo_msg_req_msg_v01

Request message; Sends an MO message.

Data fields

Type	Parameter	Description
mcm_sms_-msg_format_t_-v01	message_-format	Message format.
char	message_-content	Message content.
char	destination	Destination.
uint8_t	size_validation-_valid	Must be set to TRUE if size_validation is being passed.
mcm_sms_-msg_size_-validation_-mode_t_v01	size_validation	Size validation.

3.5.2.1.4. struct mcm_sms_send_mo_msg_resp_msg_v01

Response message; Sends an MO message.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.5.2.1.5. struct mcm_sms_set_msg_config_req_msg_v01

Request message; Sets the message configutation.

Data fields

Type	Parameter	Description
uint8_t	default_size_-validation_-mode_valid	Must be set to TRUE if default_size_validation_mode is being passed.
mcm_sms_-msg_size_-validation_-mode_t_v01	default_size_-validation_- mode	Default size validation mode.
uint8_t	enable_cb_-valid	Must be set to TRUE if enable_cb is being passed.
uint8_t	enable_cb	Enable callback.

3.5.2.1.6. struct mcm_sms_set_msg_config_resp_msg_v01

Response message; Sets the message configutation.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.5.2.1.7. struct mcm_sms_get_msg_config_resp_msg_v01

Response message; Gets the message configuration.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	default_size_-validation_-mode_valid	Must be set to TRUE if default_size_validation_mode is being passed.
mcm_sms_-msg_size_-validation_-mode_t_v01	default_size_-validation_- mode	Default size validation mode.
uint8_t	enable_cb_-valid	Must be set to TRUE if enable_cb is being passed.
uint8_t	enable_cb	Enable callback.

3.5.2.1.8. struct mcm_sms_set_reception_mode_req_msg_v01

Request message; Sets the reception mode.

Data fields

Type	Parameter	Description
mcm_sms_-reception_-mode_t_v01	reception_mode	Reception mode.
uint8_t	last_absorbed_-message_id_- valid	Must be set to TRUE if last_absorbed_message_id is being passed.
int64_t	last_absorbed_-message_id	Last absorbed message ID.

3.5.2.1.9. struct mcm_sms_set_reception_mode_resp_msg_v01

Response message; Sets the reception mode.

Data fields

Type	Parameter	Description
mcm_response_t_v01	response	Result code.

3.5.2.1.10. struct mcm_sms_event_register_req_msg_v01

Request message; Registers for an indication of events.

Data fields

Type	Parameter	Description
uint8_t	register_sms_pp_event_valid	Must be set to TRUE if register_sms_pp_event is being passed.
uint8_t	register_sms_pp_event	Receive a PP SMS event.

3.5.2.1.11. struct mcm_sms_event_register_resp_msg_v01

Response message; Registers for an indication of events.

Data fields

Type	Parameter	Description
mcm_response_t_v01	response	Result code.

3.5.2.1.12. struct mcm_sms_pp_ind_msg_v01

Indication message; Point-to-point message indication.

Data fields

Type	Parameter	Description
mcm_sms_msg_format_t_v01	message_format	Message format.
char	message_content	Message content.
char	source_address	Source address.
int64_t	message_id	Message ID.
uint8_t	message_class_valid	Must be set to TRUE if message_class is being passed.
mcm_sms_message_class_t_v01	message_class	Message class.
uint8_t	message_content_length_valid	Must be set to true if message_content_length is being passed
uint32_t	message_content_length	Message Content Length.

3.5.2.1.13. struct mcm_sms_cb_ind_msg_v01

Indication message; Cell broadcast message indication.

Data fields

Type	Parameter	Description
mcm_sms_msg_format_t_v01	message_format	Message format.
char	message_content	Message content.
uint8_t	message_content_length_valid	Must be set to true if message_content_length is being passed
uint32_t	message_content_length	Message Content Length.

3.5.2.1.14. struct mcm_sms_cb_cmas_ind_msg_v01

Indication message; Cell broadcast CMAS message indication.

Data fields

Type	Parameter	Description
uint8_t	type_0_record_valid	Must be set to TRUE if type_0_record is being passed.
mcm_cbs_cmae_record_type_0_t_v01	type_0_record	Type 0 record.
uint8_t	type_1_record_valid	Must be set to TRUE if type_1_record is being passed.
mcm_cbs_cmae_record_type_1_t_v01	type_1_record	Type 1 record.
uint8_t	type_2_record_valid	Must be set to TRUE if type_2_record is being passed.
mcm_cbs_cmae_record_type_2_t_v01	type_2_record	Type 2 record.

3.5.3. SMS Constants

This section contains the MCM DM constants.

3.5.3.1. Define Documentation

- #define MCM_SMS_MAX_MO_MSG_LENGTH_V01 1440
  Maximum length of an MO SMS (9 160).
- #define MCM_SMS_MAX_MT_MSG_LENGTH_V01 160 Maximum length of an SMS.
- #define MCM_SMS_MAX_ADDR_LENGTH_V01 252 Maximum string length.

3.5.4. SMS Enumerations

This section contains the MCC SMS enumerations.

3.5.4.1. Enumeration Type Documentation

3.5.4.1.1. enum mcm_sms_msg_format_t_v01

Enumerator:

MCM_SMS_MSG_FORMAT_TEXT_ASCII_V01 Message format ASCII text.
MCM_SMS_MSG_FORMAT_TEXT_UTF8_V01 Message format UTF8 text.
MCM_SMS_MSG_FORMAT_BINARY_STREAM_V01 Message format binary stream.

3.5.4.1.2. enum mcm_sms_msg_size_validation_mode_t_v01

Enumerator:

MCM_SMS_MSG_SIZE_VALIDATION_MODE_AUTO_BREAK_V01 Message size validation mode; Auto-break into 160-byte segments.

MCM_SMS_MSG_SIZE_VALIDATION_MODE_NO_AUTO_BREAK_V01 Message size validation mode; No auto-break.

3.5.4.1.3. enum mcm_sms_reception_mode_t_v01

Enumerator:

MCM_SMS_RECEPTION_MODE_NO_RECEPTION_V01 No reception.
MCM_SMS_RECEPTION_MODE_ON_AUTO_CONFIRM_TO_NW_V01 Reception on with auto confirm to network.

MCM_SMS_RECEPTION_MODE_ON_WITHOUT_AUTO_CONFIRM_TO_NW_V01 Reception on without auto confirm to network.

3.5.4.1.4. enum mcm_sms_message_class_t_v01

Enumerator:

MCM_SMS_MESSAGE_CLASS_0_V01 Class 0.

MCM_SMS_MESSAGE_CLASS_1_V01 Class 1.

MCM_SMS_MESSAGE_CLASS_2_V01 Class 2.

MCM_SMS_MESSAGE_CLASS_3_V01 Class 3.

MCM_SMS_MESSAGE_CLASS_NONE_V01 None.

3.5.4.1.5. enum mcm_cbs_cmae_category_type_t_v01

Enumerator:

MCM_CBS_CMAE_CATEGORY_GEO_V01
Geophysical, including landslide.

MCM_CBS_CMAE_CATEGORY_MET_V01
Meteorological, including flood.

MCM_CBS_CATEGORY_SAFETY_V01
Safety (general emergency and public safety).

MCM_CBS_CMAE_CATEGORY_SECURITY_V01
Security (law enforcement, military, homeland, and local/private security).

MCM_CBS_CMAE_CATEGORY_RESCUE_V01
Rescue (rescue and recovery).

MCM_CBS_CMAE_CATEGORY_FIRE_V01
Fire (fire suppression and rescue).

MCM_CBS_CMAE_CATEGORY_HEALTH_V01
Health (medical and public health).

MCM_CBS_CMAE_CATEGORY_ENV_V01
Environment (pollution and other environmental factors).

MCM_CBS_CMAE_CATEGORY_TRANSPORT_V01
Transport (public and private transportation).

MCM_CBS_CMAE_CATEGORY_INFRA_V01
Infrastructure (utility, telecommunication, and other nontransport infrastructure).

MCM_CBS_CMAE_CATEGORY_CBRNE_V01
CBRNE (chemical, biological, radiological, nuclear, or high-yield explosive thread or attack).

MCM_CBS_CMAE_CATEGORY_OTHER_V01
Other events.

3.5.4.1.6. enum mcm_cbs_cmae_response_type_t_v01

Enumerator:

MCM_CBS_CMAE_RESPONSE_TYPE_SHELTER_V01
Shelter (take shelter in place).

MCM_CBS_CMAE_RESPONSE_TYPE_EVACUATE_V01
Evacuate (relocate).

MCM_CBS_CMAE_RESPONSE_TYPE_PREPARE_V01
Prepare (make preparations).

MCM_CBS_CMAE_RESPONSE_TYPE_EXECUTE_V01
Execute (execute a preplanned activity).

MCM_CBS_CMAE_RESPONSE_TYPE_MONITOR_V01
Monitor (attend to information sources).

MCM_CBS_CMAE_RESPONSE_TYPE_AVOID_V01
Avoid (avoid hazards).

MCM_CBS_CMAE_RESPONSE_TYPE_ASSESS_V01
Assess (evaluate the information in this message).

MCM_CBS_CMAE_RESPONSE_TYPE_NONE_V01
None (no action recommended).

3.5.4.1.7. enum mcm_cbs_cmae_severity_type_t_v01

Enumerator:

MCM_CBS_CMAE_SEVERITY_EXTREME_V01
Extreme (extraodinary threat to life or property).

MCM_CBS_CMAE_SEVERITY_SEVERE_V01
Severe (significant threat to life or property).

3.5.4.1.8. enum mcm_cbs_cmae_urgency_type_t_v01

Enumerator:

MCM_CBS_CMAE_URGENCY_IMMEDIATE_V01
Immediate (responsive action should be taken immediately).

MCM_CBS_CMAE_URGENCY_EXPECTED_V01
Expected (reponsive action should be taken soon, i.e., within the next hour).

3.5.4.1.9. enum mcm_cbs_cmae_certainty_type_t_v01

Enumerator:

MCM_CBS_CMAE_CERTAINTY_OBSERVED_V01
Observed (determined to have occurred or to be ongoing).

MCM_CBS_CMAE_CERTAINTY_LIKELY_V01
Likely (probabiltiy > 50%).

3.5.5. SMS Data Structures

This section contains the MCM SMS data structures.

3.5.5.1. Data Structure Documentation

3.5.5.1.1. struct mcm_cbs_cmae_expire_t_v01

Data fields

Type	Parameter	Description
uint8_t	year	Year – Range 00 to 99 (UTC).
uint8_t	month	Month – Range 1 to 12 (UTC).
uint8_t	day	Day – Range 1 to 31 (UTC).
uint8_t	hours	Hour – Range 0 to 23 (UTC).
uint8_t	minutes	Minutes – Range 0 to 59 (UTC).
uint8_t	seconds	Seconds – Range 0 to 59 (UTC).

3.5.5.1.2. struct mcm_cbs_cmae_record_type_0_t_v01

Data fields

Type	Parameter	Description
uint32_t	message_-content_len	Must be set to the number of elements in message_content.
char	message_-content	Message content.

3.5.5.1.3. struct mcm_cbs_cmae_record_type_1_t_v01

Data fields

Type	Parameter	Description
mcm_cbs_-cmae_category-_type_t_v01	category	Category of the CMAS alert.

Type	Parameter	Description
mcm_cbs_-cmae_response-_type_t_v01	response	Response indicated for the CMAS alert.
mcm_cbs_-cmae_severity-_type_t_v01	severity	Severity of the CMAS alert.
mcm_cbs_-cmae_urgency-_type_t_v01	urgency	Urgency of the CMAS alert.
mcm_cbs_-cmae_certainty-_type_t_v01	certainty	Certainty of the CMAS alert.

3.5.5.1.4. struct mcm_cbs_cmae_record_type_2_t_v01

Data fields

Type	Parameter	Description
uint16_t	id	Identification of the message.
uint8_t	alert_handling	Indicates whether this alert message requires special handling.
mcm_cbs_-cmae_expire_t-_v01	expire	Expiration date and time of the CMAS alert.
uint8_t	language	Language used for the message content.

3.6. Mobile Access Point

This section contains APIs for enabling and disabling Mobile Access Point (Mobile AP) functionality, backhaul connectivity, and obtaining other Mobile AP-related configuration.

- Mobile AP Message Identifiers
- Mobile AP Message Structures
- Mobile AP Constants
- Mobile AP Enumerations
- Mobile AP Data Structures

3.6.1. Mobile AP Message Identifiers

This section contains the MCM mobile AP message identifiers.

- #define MCM_MOBILEAP_ENABLE_REQ_V01 0x0400
- #define MCM_MOBILEAP_ENABLE_RESP_V01 0x0400
- #define MCM_MOBILEAP_DISABLE_REQ_V01 0x0401
- #define MCM_MOBILEAP_DISABLE_RESP_V01 0x0401
- #define MCM_MOBILEAP_BRING_UP_WWAN_REQ_V01 0x0402
- #define MCM_MOBILEAP_BRING_UP_WWAN_RESP_V01 0x0402
- #define MCM_MOBILEAP_TEAR_DOWN_WWAN_REQ_V01 0x0403
- #define MCM_MOBILEAP_TEAR_DOWN_WWAN_RESP_V01 0x0403
- #define MCM_MOBILEAP_ADD_STATIC_NAT_ENTRY_REQ_V01 0x0404
- #define MCM_MOBILEAP_ADD_STATIC_NAT_ENTRY_RESP_V01 0x0404
- #define MCM_MOBILEAP_GET_STATIC_NAT_ENTRY_REQ_V01 0x0405
- #define MCM_MOBILEAP_GET_STATIC_NAT_ENTRY_RESP_V01 0x0405
- #define MCM_MOBILEAP_DELETE_STATIC_NAT_ENTRY_REQ_V01 0x0406
- #define MCM_MOBILEPA_DELETE_STATIC_NAT_ENTRY_RESP_V01 0x0406
- #define MCM_MOBILEAP_SET_NAT_TIMEOUT_REQ_V01 0x0407
- #define MCM_MOBILEAP_SET_NAT_TIMEOUT_RESP_V01 0x0407
- #define MCM_MOBILEAP_GET_NAT_TIMEOUT_REQ_V01 0x0408
- #define MCM_MOBILEAP_GET_NAT_TIMEOUT_RESP_V01 0x0408

- #define MCM_MOBILEAP_SET_NAT_TYPE_REQ_V01 0x0409
- #define MCM_MOBILEAP_SET_NAT_TYPE_RESP_V01 0x0409
- #define MCM_MOBILEAP_GET_NAT_TYPE_REQ_V01 0x040A
- #define MCM_MOBILEAP_GET_NAT_TYPE_RESP_V01 0x040A
- #define MCM_MOBILEAP_ADD_FIREWALL_ENTRY_REQ_V01 0x040B
- #define MCM_MOBILEAP_ADD_FIREWALL_ENTRY_RESP_V01 0x040B
- #define MCM_MOBILEAP_GET_FIREWALL_ENTRIES_HANDLE_LIST_REQ_V01 0x040C
- #define MCM_MOBILEAP_GET_FIREWALL_ENTRIES_HANDLE_LIST_RESP_V01 0x040C
- #define MCM_MOBILEAP_GET_FIREWALL_ENTRY_REQ_V01 0x040D
- #define MCM_MOBILEAP_GET_FIREWALL_ENTRY_RESP_V01 0x040D
- #define MCM_MOBILEAP_DELETE_FIREWALL_ENTRY_REQ_V01 0x040E
- #define MCM_MOBILEAP_DELETE_FIREWALL_ENTRY_RESP_V01 0x040E
- #define MCM_MOBILEAP_SET_FIREWALL_CONFIG_REQ_V01 0x040F
- #define MCM_MOBILEAP_SET_FIREWALL_CONFIG_RESP_V01 0x040F
- #define MCM_MOBILEAP_SET_DMZ_REQ_V01 0x0410
- #define MCM_MOBILEAP_SET_DMZ_RESP_V01 0x0410
- #define MCM_MOBILEAP_DELETE_DMZ_REQ_V01 0x0411
- #define MCM_MOBILEAP_DELETE_DMZ_RESP_V01 0x0411
- #define MCM_MOBILEAP_GET_DMZ_REQ_V01 0x0412
- #define MCM_MOBILEAP_GET_DMZ_RESP_V01 0x0412
- #define MCM_MOBILEAP_GET_IPV4_WWAN_CONFIG_REQ_V01 0x0413
- #define MCM_MOBILEAP_GET_IPV4_WWAN_CONFIG_RESP_V01 0x0413
- #define MCM_MOBILEAP_GET_WWAN_STATS_REQ_V01 0x0414
- #define MCM_MOBILEAP_GET_WWAN_STATS_RESP_V01 0x0414
- #define MCM_MOBILEAP_RESET_WWAN_STATS_REQ_V01 0x0415
- #define MCM_MOBILEAP_RESET_WWAN_STATS_RESP_V01 0x0415
- #define MCM_MOBILEAP_SET_DHCPD_CONFIG_REQ_V01 0x0416
- #define MCM_MOBILEAP_SET_DHCPD_CONFIG_RESP_V01 0x0416

- #define MCM_MOBILEAP_ENABLE_WLAN_REQ_V01 0x0417
- #define MCM_MOBILEAP_ENABLE_WLAN_RESP_V01 0x0417
- #define MCM_MOBILEAP_DISABLE_WLAN_REQ_V01 0x0418
- #define MCM_MOBILEAP_DISABLE_WLAN_RESP_V01 0x0418
- #define MCM_MOBILEAP_GET_IPSEC_VPN_PASS_THROUGH_REQ_V01 0x0419
- #define MCM_MOBILEAP_GET_IPSEC_VPN_PASS_THROUGH_RESP_V01 0x0419
- #define MCM_MOBILEAP_SET_IPSEC_VPN_PASS_THROUGH_REQ_V01 0x041A
- #define MCM_MOBILEAP_SET_IPSEC_VPN_PASS_THROUGH_RESP_V01 0x041A
- #define MCM_MOBILEAP_GET_PPTP_VPN_PASS_THROUGH_REQ_V01 0x041B
- #define MCM_MOBILEAP_GET_PPTP_VPN_PASS_THROUGH_RESP_V01 0x041B
- #define MCM_MOBILEAP_SET_PPTP_VPN_PASS_THROUGH_REQ_V01 0x041C
- #define MCM_MOBILEAP_SET_PPTP_VPN_PASS_THROUGH_RESP_V01 0x041C
- #define MCM_MOBILEAP_GET_L2TP_VPN_PASS_THROUGH_REQ_V01 0x041D
- #define MCM_MOBILEAP_GET_L2TP_VPN_PASS_THROUGH_RESP_V01 0x041D
- #define MCM_MOBILEAP_SET_L2TP_VPN_PASS_THROUGH_REQ_V01 0x041E
- #define MCM_MOBILEAP_SET_L2TP_VPN_PASS_THROUGH_RESP_V01 0x041E
- #define MCM_MOBILEAP_SET_AUTO_CONNECT_REQ_V01 0x041F
- #define MCM_MOBILEAP_SET_AUTO_CONNECT_RESP_V01 0x041F
- #define MCM_MOBILEAP_GET_AUTO_CONNECT_REQ_V01 0x0420
- #define MCM_MOBILEAP_GET_AUTO_CONNECT_RESP_V01 0x0420
- #define MCM_MOBILEAP_SET_ROAMING_PREF_REQ_V01 0x0421
- #define MCM_MOBILEAP_SET_ROAMING_PREF_RESP_V01 0x0421
- #define MCM_MOBILEAP_GET_ROAMING_PREF_REQ_V01 0x0422
- #define MCM_MOBILEAP_GET_ROAMING_PREF_RESP_V01 0x0422
- #define MCM_MOBILEAP_SET_DUALAP_CONFIG_REQ_V01 0x0423
- #define MCM_MOBILEAP_SET_DUALAP_CONFIG_RESP_V01 0x0423
- #define MCM_MOBILEAP_STATION_MODE_ENABLE_REQ_V01 0x0424
- #define MCM_MOBILEAP_STATION_MODE_ENABLE_RESP_V01 0x0424
- #define MCM_MOBILEAP_STATION_MODE_DISABLE_REQ_V01 0x0425

- #define MCM_MOBILEAP_STATION_MODE_DISABLE_RESP_V01 0x0425
- #define MCM_MOBILEAP_EVENT_REGISTER_REQ_V01 0x0426
- #define MCM_MOBILEAP_EVENT_REGISTER_RESP_V01 0x0426
- #define MCM_MOBILEAP_UNSOL_EVENT_IND_V01 0x0427

3.6.2. Mobile AP Message Structures

This section contains the MCM mobile access point message structures.

3.6.2.1. Data Structure Documentation

3.6.2.1.1. struct mcm_mobileap_enable_resp_msg_v01

Response message; Enables the mobile AP functionality via a single mobile AP instance on the ARM® Cortex™-A5 processor.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	mcm_-mobileap_-handle_valid	Must be set to TRUE if mcm_mobileap_handle is being passed.
uint32_t	mcm_-mobileap_- handle	Mobile AP handle.

3.6.2.1.2. struct mcm_mobileap_disable_req_msg_v01

Request message; Disables the mobile AP functionality for a mobile AP instance on the Cortex-A5 processor.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.3. struct mcm_mobileap_disable_resp_msg_v01

Response message; Disables the mobile AP functionality for a mobile AP instance on the Cortex-A5 processor.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.4. struct mcm_mobileap_bring_up_wwan_req_msg_v01

Request message; Invokes bringing up the WWAN from the mobile AP.

The call is established using the stored network policy that enabled the mobile AP via MCM_MOBILEAP_ENABLE_REQ. If the control point issues multiple requests in short intervals, an MCM_ERROR_NO_EFFECT error is returned indicating that the previous request is still in process.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	ip_version_-valid	Must be set to TRUE if ip_version is being passed.
mcm_-mobileap_ip-_version_t_v01	ip_version	IP version.

3.6.2.1.5. struct mcm_mobileap_bring_up_wwan_resp_msg_v01

Response message; Invokes bringing up the WWAN from the mobile AP.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.6. struct mcm_mobileap_tear_down_wwan_req_msg_v01

Request message; Brings down a WWAN call, if connected.

This command brings down the backhaul functionality. If the control point issues multiple requests in short intervals, an MCM_ERROR_NO_EFFECT error is returned indicating that the previous request is still in process.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	ip_version_-valid	Must be set to TRUE if ip_version is being passed.
mcm_-mobileap_ip-_version_t_v01	ip_version	IP version.

3.6.2.1.7. struct mcm_mobileap_tear_down_wwan_resp_msg_v01

Response message; Brings down a WWAN call, if connected.

Data fields

Type	Parameter	Description
uint8_t	resp_valid	Must be set to TRUE if resp is being passed.
mcm_response-_t_v01	resp	Result code.

3.6.2.1.8. struct mcm_mobileap_add_static_nat_entry_req_msg_v01

Request message; Adds a static network address translation (NAT) entry.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_-static_nat_-entry_conf_t_- v01	nat_entry_-config	Mobile AP static NAT entry configuration.

3.6.2.1.9. struct mcm_mobileap_add_static_nat_entry_resp_msg_v01

Response message; Adds a static NAT entry.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.10. struct mcm_mobileap_get_static_nat_entry_req_msg_v01

Request message; Queries all static NAT entries.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_MOBILEAP_ENABLE_REQ.
uint32_t	max_entries	Maximum number of SNAT entries requested by the client.

3.6.2.1.11. struct mcm_mobileap_get_static_nat_entry_resp_msg_v01

Response message; Queries all static NAT entries. The response message contains the number of entries followed by the value of these entries sequentially.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	snat_entries_-valid	Must be set to TRUE if snat_entries is being passed.
uint32_t	snat_entries_-len	Must be set to the number of elements in snat_entries.
mcm_-mobileap_-static_nat_-entry_conf_t_- v01	snat_entries	MCM mobile AP static NAT entry configuration.

3.6.2.1.12. struct mcm_mobileap_delete_static_nat_entry_req_msg_v01

Request message; Deletes a static NAT entry.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_-static_nat_-entry_conf_t_- v01	snat_entry	MCM mobile AP static NAT entry request message.

3.6.2.1.13. struct mcm_mobileap_delete_static_nat_entry_resp_msg_v01

Response message; Deletes a static NAT entry.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.14. struct mcm_mobileap_set_nat_timeout_req_msg_v01

Request message; Configures different types of NAT timeouts. The command handler overwrites any previously configured value with the current value.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_nat_-timeout_t_v01	timeout_type	NAT timeout type to be used. Values: MCM_MOBILEAP_NAT_TIMEOUT_GENERIC (0x01) – Generic NAT timeout. MCM_MOBILEAP_NAT_TIMEOUT_ICMP (0x02) – NAT timeout for ICMP. MCM_MOBILEAP_NAT_TIMEOUT_TCP_ESTABLISHED (0x03) – NAT timeout for the established TCP. MCM_MOBILEAP_NAT_TIMEOUT_UDP (0x04) – NAT timeout for UDP.
uint32_t	timeout_value	NAT timeout value to be used, in seconds.

3.6.2.1.15. struct mcm_mobileap_set_nat_timeout_resp_msg_v01

Response message; Configures different types of NAT timeouts.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.16. struct mcm_mobileap_get_nat_timeout_req_msg_v01

Request message; Gets the configured NAT timeout value.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_nat_-timeout_t_v01	timeout_type	NAT timeout type used. Values: MCM_MOBILEAP_NAT_TIMEOUT_GENERIC (0x01) – Generic NAT timeout. MCM_MOBILEAP_NAT_TIMEOUT_ICMP (0x02) – NAT timeout for ICMP. MCM_MOBILEAP_NAT_TIMEOUT_TCP_ESTABLISHED (0x03) – NAT timeout for the established TCP. MCM_MOBILEAP_NAT_TIMEOUT_UDP (0x04) – NAT timeout for UDP.

3.6.2.1.17. struct mcm_mobileap_get_nat_timeout_resp_msg_v01

Response message; Gets the configured NAT timeout value.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	
uint8_t	timeout_value-_valid	Must be set to TRUE if timeout_value is being passed.
uint32_t	timeout_value	NAT timeout value used, in seconds.

3.6.2.1.18. struct mcm_mobileap_set_nat_type_req_msg_v01

Request message; Configures the NAT type setting. The command handler overwrites any previously configured value with the current value.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_-nat_type_t_v01	nat_type	Type of NAT. Values: MCM_MOBILEAP_NAT_SYMMETRIC_NAT (0) – Symmetric NAT MCM_MOBILEAP_NAT_PORT_RESTRICTED_CONE_NAT – Port-restricted cone NAT MCM_MOBILEAP_NAT_FULL_CONE_NAT (2) – Full cone NAT MCM_MOBILEAP_NAT_ADDRESS_RESTRICTED_NAT (3) – Address-restricted NAT

3.6.2.1.19. struct mcm_mobileap_set_nat_type_resp_msg_v01

Response message; Configures the NAT type setting.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.20. struct mcm_mobileap_get_nat_type_req_msg_v01

Request message; Gets the NAT type setting.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.21. struct mcm_mobileap_get_nat_type_resp_msg_v01

Response message; Gets the NAT type setting.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	nat_type_valid	Must be set to TRUE if nat_type is being passed.
mcm_-mobileap_-nat_type_t_v01	nat_type	Type of NAT. Values: MCM_MOBILEAP_NAT_SYMMETRIC_NAT (0) – Symmetric NAT MCM_MOBILEAP_NAT_PORT_RESTRICTED_CONE_NAT – Port-restricted cone NAT MCM_MOBILEAP_NAT_FULL_CONE_NAT (2) – Full cone NAT MCM_MOBILEAP_NAT_ADDRESS_RESTRICTED_NAT (3) – Address-restricted NAT

3.6.2.1.22. struct mcm_mobileap_add_firewall_entry_req_msg_v01

Request message; Adds IP filter-based firewall rules.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_ip-_family_t_v01	ip_version	Firewall family version. Values: MCM_MOBILEAP_IP_FAMILY_V4 (0x04) – IP family v4 MCM_MOBILEAP_IP_FAMILY_V6 (0x06) – IP family v6 MCM_MOBILEAP_IP_FAMILY_V4V6 (0x0A) – IP family v4/v6
uint8_t	next_hdr_prot-_valid	Must be set to TRUE if next_hdr_prot is being passed.
uint8_t	next_hdr_prot	Next protocol header after the IP header.
uint8_t	tcp_udp_src_-valid	Must be set to TRUE if tcp_udp_src is being passed.
mcm_-mobileap_-tcp_udp_port_-range_t_v01	tcp_udp_src	TCP_UDP source port.
uint8_t	tcp_udp_dst_-valid	Must be set to TRUE if tcp_udp_dst is being passed.
mcm_-mobileap_-tcp_udp_port_-range_t_v01	tcp_udp_dst	TCP_UDP destination port.
uint8_t	icmp_type_-valid	Must be set to TRUE if icmp_type is being passed.
uint8_t	icmp_type	ICMP type, as specified in the ICMP protocol, RFC 792.
uint8_t	icmp_code_-valid	Must be set to TRUE if icmp_code is being passed.
uint8_t	icmp_code	ICMP code as specified in the ICMP protocol, RFC 792.
uint8_t	esp_spi_valid	Must be set to TRUE if esp_spi is being passed.
uint32_t	esp_spi	Security parameter index, as specified in the ESP protocol, RFC 4303.
uint8_t	ip4_src_addr_-valid	Must be set to TRUE if ip4_src_addr is being passed.
mcm_-mobileap_ip4-_addr_subnet_-mask_t_v01	ip4_src_addr	IPv4 source address and subnet mask.
uint8_t	ip4_dst_addr_-valid	Must be set to TRUE if ip4_dst_addr is being passed.
mcm_-mobileap_ip4-_addr_subnet_-mask_t_v01	ip4_dst_addr	IPv4 destination address and subnet mask.
uint8_t	ip4_tos_valid	Must be set to TRUE if ip4_tos is being passed.
mcm_-	ip4_tos	IPv6 TOS value and mask.

Type	Parameter	Description
mobileap_-ip4_tos_t_v01		
uint8_t	ip6_src_addr_-valid	Must be set to TRUE if ip6_src_addr is being passed.
mcm_-mobileap_ip6-_addr_prefix_-len_t_v01	ip6_src_addr	IPv6 source address and prefix length.
uint8_t	ip6_dst_addr_-valid	Must be set to TRUE if ip6_dst_addr is being passed.
mcm_-mobileap_ip6-_addr_prefix_-len_t_v01	ip6_dst_addr	IPv6 source address and prefix length.
uint8_t	ip6_trf_cls_-valid	Must be set to TRUE if ip6_trf_cls is being passed.
mcm_-mobileap_ip6_-traffic_class_t_-v01	ip6_trf_cls	IPv6 traffic class value and mask.

3.6.2.1.23. struct mcm_mobileap_add_firewall_entry_resp_msg_v01

Response message; Adds IP filter-based firewall rules.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	firewall_-handle_valid	Must be set to TRUE if firewall_handle is being passed.
uint32_t	firewall_handle	Identifies the handle for the added firewall rule.

3.6.2.1.24. struct mcm_mobileap_get_firewall_entries_handle_list_req_msg_v01

Request message; Gets the handles of all firewall rules.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. Value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

Type	Parameter	Description
mcm_-mobileap_ip-_family_t_v01	ip_version	Identifies the firewall family version. Values: MCM_MOBILEAP_IP_FAMILY_V4 (0x04) – IP family v4 MCM_MOBILEAP_IP_FAMILY_V6 (0x06) – IP family v6 MCM_MOBILEAP_IP_FAMILY_V4V6 (0x0A) – IP family v4/v6

3.6.2.1.25. struct mcm_mobileap_get_firewall_entries_handle_list_resp_msg_v01

Response message; Gets the handles of all firewall rules.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	firewall_-handle_list_- valid	Must be set to TRUE if firewall_handle_list is being passed.
uint32_t	firewall_-handle_list_len	Must be set to the number of elements in firewall_handle_list.
uint32_t	firewall_-handle_list	Handles identifying the firewall entry. The value must be the handle previously returned by MCM_MOBILEAP_ADD_FIREWALL_ENTRY_RESP.

3.6.2.1.26. struct mcm_mobileap_get_firewall_entry_req_msg_v01

Request message; Gets the firewall rules.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint32_t	firewall_handle	Handle identifying the firewall entry. The value must be the handle previously returned by MCM_MOBILEAP_ADD_FIREWALL_ENTRY_RESP or MCM_MOBILEAP_GET_FIREWALL_ENTRIES_HANDLE_LI- ST_RESP.

3.6.2.1.27. struct mcm_mobileap_get_firewall_entry_resp_msg_v01

Response message; Gets the firewall rules.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	ip_version_-valid	Must be set to TRUE if ip_version is being passed.
mcm_-mobileap_ip-_family_t_v01	ip_version	Identifies the firewall family version. Values: MCM_MOBILEAP_IP_FAMILY_V4 (0x04) – IP family v4 MCM_MOBILEAP_IP_FAMILY_V6 (0x06) – IP family v6 MCM_MOBILEAP_IP_FAMILY_V4V6 (0x0A) – IP family v4/v6
uint8_t	next_hdr_prot-_valid	Must be set to TRUE if next_hdr_prot is being passed.
uint8_t	next_hdr_prot	IPv4/IPv6 next header protocol.
uint8_t	tcp_udp_src_-valid	Must be set to TRUE if tcp_udp_src is being passed.
mcm_-mobileap_-tcp_udp_port_-range_t_v01	tcp_udp_src	TCP, UDP, and TCP_UDP source port.
uint8_t	tcp_udp_dst_-valid	Must be set to TRUE if tcp_udp_dst is being passed.
mcm_-mobileap_-tcp_udp_port_-range_t_v01	tcp_udp_dst	TCP, UDP, and TCP_UDP destination port.
uint8_t	icmp_type_-valid	Must be set to TRUE if icmp_type is being passed.
uint8_t	icmp_type	ICMP type, as specified in the ICMP protocol, RFC 792.
uint8_t	icmp_code_-valid	Must be set to TRUE if icmp_code is being passed.
uint8_t	icmp_code	ICMP code, as specified in the ICMP protocol, RFC 792.
uint8_t	esp_spi_valid	Must be set to TRUE if esp_spi is being passed.
uint32_t	esp_spi	Security parameter index, as specified in the ESP protocol, RFC 4303.
uint8_t	ip4_src_addr_-valid	Must be set to TRUE if ip4_src_addr is being passed.
mcm_-mobileap_ip4-_addr_subnet_-mask_t_v01	ip4_src_addr	IPv4 source address and subnet mask.
uint8_t	ip4_dst_addr_-valid	Must be set to TRUE if ip4_dst_addr is being passed.
mcm_-mobileap_ip4-	ip4_dst_addr	IPv4 destination address and subnet mask.

Type	Parameter	Description
_addr_subnet_-mask_t_v01		
uint8_t	ip4_tos_valid	Must be set to TRUE if ip4_tos is being passed.
mcm_-mobileap_-ip4_tos_t_v01	ip4_tos	IPv4 TOS value and mask.
uint8_t	ip6_src_addr_-valid	Must be set to TRUE if ip6_src_addr is being passed.
mcm_-mobileap_ip6-_addr_prefix_-len_t_v01	ip6_src_addr	IPv6 source address and prefix length.
uint8_t	ip6_dst_addr_-valid	Must be set to TRUE if ip6_dst_addr is being passed.
mcm_-mobileap_ip6-_addr_prefix_-len_t_v01	ip6_dst_addr	IPv6 destination address and prefix length.
uint8_t	ip6_trf_cls_-valid	Must be set to TRUE if ip6_trf_cls is being passed.
mcm_-mobileap_ip6_-traffic_class_t_-v01	ip6_trf_cls	IPv6 traffic class value and mask.

3.6.2.1.28. struct mcm_mobileap_delete_firewall_entry_req_msg_v01

Request message; Deletes a firewall rule identified by a handle.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. Value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint32_t	firewall_handle	Handle identifying the firewall entry.

3.6.2.1.29. struct mcm_mobileap_delete_firewall_entry_resp_msg_v01

Response message; Deletes a firewall rule identified by a handle.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.30. struct mcm_mobileap_set_firewall_config_req_msg_v01

Request message; Sets the firewall configuration.

This command enables or disables the firewall. If the firewall is enabled, it sets the firewall state to accept or drop the packets.

Data fields

Type	Parameter	Description
uint32_t	mcm_- mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	firewall_- enabled	Indicates whether the firewall is to be enabled or disabled; a Boolean value.
uint8_t	pkts_allowed_- valid	Must be set to TRUE if pkts_allowed is being passed.
uint8_t	pkts_allowed	Indicates whether packets are to be accepted or dropped; a Boolean value.

3.6.2.1.31. struct mcm_mobileap_set_firewall_config_resp_msg_v01

Response message; Sets the firewall configuration.

Data fields

Type	Parameter	Description
mcm_response- _t_v01	resp	Result code.

3.6.2.1.32. struct mcm_mobileap_add_dmz_req_msg_v01

Request message; Sets the DMZ (perimeter network) IP address for the mobile AP.

Data fields

Type	Parameter	Description
uint32_t	mcm_- mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint32_t	dmz_ip_addr	DMZ IP address.

3.6.2.1.33. struct mcm_mobileap_add_dmz_resp_msg_v01

Response message; Sets the DMZ (perimeter network) IP address for the mobile AP.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.34. struct mcm_mobileap_get_dmz_req_msg_v01

Request message; Queries the DMZ IP address on the mobile AP.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.35. struct mcm_mobileap_get_dmz_resp_msg_v01

Response message; Queries the DMZ IP address on the mobile AP. If no DMZ is set by the client, an IP address of 0.0.0.0 is returned.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	dmz_ip_addr_-valid	Must be set to TRUE if dmz_ip_addr is being passed.
uint32_t	dmz_ip_addr	DMZ IP address.

3.6.2.1.36. struct mcm_mobileap_delete_dmz_req_msg_v01

Request message; Deletes the DMZ entry or DMZ IP address.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint32_t	dmz_ip_addr	DMZ IP address.

3.6.2.1.37. struct mcm_mobileap_delete_dmz_resp_msg_v01

Response message; Deletes the DMZ entry or DMZ IP address.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.38. struct mcm_mobileap_get_ipv4_wwan_config_req_msg_v01

Request message; Queries the WWAN IP configuration. The command must be issued by the control point after MCM_MOBILEAP_WWAN_STATUS_IND has indicated success in bringing up a WWAN, otherwise an MCM_ERROR_INTERNAL error is returned.

Data fields

Type	Parameter	Description
uint32_t	mcm_- mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.39. struct mcm_mobileap_get_ipv4_wwan_config_resp_msg_v01

Response message; Queries the WWAN IP configuration.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	v4_addr_valid	Must be set to TRUE if v4_addr is being passed.
uint32_t	v4_addr	IPv4 address.
uint8_t	v4_prim_dns_-addr_valid	Must be set to TRUE if v4_prim_dns_addr is being passed.
uint32_t	v4_prim_dns_-addr	IPv4 primary DNS address.
uint8_t	v4_sec_dns_-addr_valid	Must be set to TRUE if v4_sec_dns_addr is being passed.
uint32_t	v4_sec_dns_-addr	IPv4 secondary DNS address.

3.6.2.1.40. struct mcm_mobileap_get_wwan_stats_req_msg_v01

Request message; Gets WWAN statistics.

Data fields

Type	Parameter	Description
uint32_t	mcm_- mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

Type	Parameter	Description
mcm_-mobileap_ip-_family_t_v01	ip_family	Identifies the IP version to be used. Values: MCM_MOBILEAP_IP_FAMILY_V4 (0x04) – IP family v4 MCM_MOBILEAP_IP_FAMILY_V6 (0x06) – IP family v6 MCM_MOBILEAP_IP_FAMILY_V4V6 (0x0A) – IP family v4/v6

3.6.2.1.41. struct mcm_mobileap_get_wwan_stats_resp_msg_v01

Response message; Gets WWAN statistics.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	wwan_stats_-valid	Must be set to TRUE if wwan_stats is being passed.
mcm_-mobileap-_wwan_-statistics_t_v01	wwan_stats	WWAN statistics.

3.6.2.1.42. struct mcm_mobileap_reset_wwan_stats_req_msg_v01

Request message; Resets WWAN statistics.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_ip-_family_t_v01	ip_family	Identifies the IP version to be used. Values: MCM_MOBILEAP_IP_FAMILY_V4 (0x04) – IP family v4 MCM_MOBILEAP_IP_FAMILY_V6 (0x06) – IP family v6 MCM_MOBILEAP_IP_FAMILY_V4V6 (0x0A) – IP family v4/v6

3.6.2.1.43. struct mcm_mobileap_reset_wwan_stats_resp_msg_v01

Response message; Resets WWAN statistics.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.44. struct mcm_mobileap_set_dhcpd_config_req_msg_v01

Request message; Sets the DHCPD configuration.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_-dhcpd_config_-t_v01	dhcpd_config	DHCPD configuration.

3.6.2.1.45. struct mcm_mobileap_set_dhcpd_config_resp_msg_v01

Response message; Sets the DHCPD configuration.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.46. struct mcm_mobileap_enable_wlan_req_msg_v01

Request message; Enables the WLAN.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_MOBILEAP_ENABLE_REQ.

3.6.2.1.47. struct mcm_mobileap_enable_wlan_resp_msg_v01

Response message; Enables the WLAN.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.48. struct mcm_mobileap_disable_wlan_req_msg_v01

Request message; Disables the WLAN.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	vpn_pass_-through_value	Indicates whether an IPSec VPN passthrough is allowed; a Boolean value.

3.6.2.1.49. struct mcm_mobileap_disable_wlan_resp_msg_v01

Response message; Disables the WLAN.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.50. struct mcm_mobileap_set_ipsec_vpn_pass_through_req_msg_v01

Request message; Configures the Internet Protocol Security (IPSec) Virtual Private Network (VPN) passthrough setting.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	vpn_pass_-through_value	Indicates whether an IPSec VPN passthrough is allowed; a Boolean value.

3.6.2.1.51. struct mcm_mobileap_set_ipsec_vpn_pass_through_resp_msg_v01

Response message; Configures the IPSec VPN passthrough setting. The command handler overwrites any previously configured value with the current value.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.52. struct mcm_mobileap_get_ipsec_vpn_pass_through_req_msg_v01

Request message; Queries the IPSec VPN passthrough setting.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.53. struct mcm_mobileap_get_ipsec_vpn_pass_through_resp_msg_v01

Response message; Queries the IPSec VPN passthrough setting.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	vpn_pass_-through_value-_valid	Must be set to TRUE if vpn_pass_through_value is being passed.
uint8_t	vpn_pass_-through_value	VPN passthrough value. Indicates whether a PPTP VPN passthrough is allowed; a Boolean value.

3.6.2.1.54. struct mcm_mobileap_set_pptp_vpn_pass_through_req_msg_v01

Request message; Configures the Point-to-Point Tunneling Protocol (PPTP) VPN passthrough setting. The command handler overwrites any previously configured value with the current value.

Data fields

Type	Parameter	Description
uint32_t	mcm_- mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	vpn_pass_- through_value	Indicates whether an L2TP VPN passthrough is allowed; a Boolean value.

3.6.2.1.55. struct mcm_mobileap_set_pptp_vpn_pass_through_resp_msg_v01

Response message; Configures the PPTP VPN passthrough setting.

Data fields

Type	Parameter	Description
mcm_response- _t_v01	resp	Result code.

3.6.2.1.56. struct mcm_mobileap_get_pptp_vpn_pass_through_req_msg_v01

Request message; Queries the PPTP VPN passthrough setting.

Data fields

Type	Parameter	Description
uint32_t	mcm_- mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.57. struct mcm_mobileap_get_pptp_vpn_pass_through_resp_msg_v01

Response message; Queries the PPTP VPN passthrough setting.

Data fields

Type	Parameter	Description
mcm_response- _t_v01	resp	Result code.
uint8_t	vpn_pass_- through_value- _valid	Must be set to TRUE if vpn_pass_through_value is being passed.
uint8_t	vpn_pass_- through_value	Indicates whether an L2TP VPN passthrough is allowed; a Boolean value.

3.6.2.1.58. struct mcm_mobileap_set_l2tp_vpn_pass_through_req_msg_v01

Request message; Configures the Layer 2 Tunneling Protocol (L2TP) VPN passthrough setting. The command handler overwrites any previously configured value with the current value.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	vpn_pass_-through_value	Indicates whether an L2TP VPN passthrough is allowed; a Boolean value.

3.6.2.1.59. struct mcm_mobileap_set_l2tp_vpn_pass_through_resp_msg_v01

Response message; Configures the Layer 2 Tunneling Protocol (L2TP) VPN passthrough setting.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.60. struct mcm_mobileap_get_l2tp_vpn_pass_through_req_msg_v01

Request message; Queries the L2TP VPN passthrough setting.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.61. struct mcm_mobileap_get_l2tp_vpn_pass_through_resp_msg_v01

Response message; Queries the L2TP VPN passthrough setting.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	vpn_pass_-through_value-_valid	Must be set to TRUE if vpn_pass_through_value is being passed.
uint8_t	vpn_pass_-through_value	Indicates whether an L2TP VPN passthrough is allowed; a Boolean value.

3.6.2.1.62. struct mcm_mobileap_set_auto_connect_req_msg_v01

Request message; Sets the autoconnect flag.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	enable	Enable/disable autoconnect. Values: TRUE – Enable FALSE – Disable

3.6.2.1.63. struct mcm_mobileap_set_auto_connect_resp_msg_v01

Response message; Sets the autoconnect flag.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.64. struct mcm_mobileap_get_auto_connect_req_msg_v01

Request message; Gets the autoconnect flag.

Data fields

Type	Parameter	Description
uint32_t	mcm_- mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.65. struct mcm_mobileap_get_auto_connect_resp_msg_v01

Response message; Gets the autoconnect flag.

Data fields

Type	Parameter	Description
mcm_response- _t_v01	resp	Result code.
uint8_t	auto_conn_- flag_valid	Must be set to TRUE if auto_conn_flag is being passed.
uint8_t	auto_conn_flag	Autoconnect status. Values: TRUE – Enabled FALSE – Disabled

3.6.2.1.66. struct mcm_mobileap_set_roaming_pref_req_msg_v01

Request message; Configures whether QCMAP_MSGR initiates WWAN data calls while roaming. The roaming mode determines the QCMAP_MSGR policy for establishing new data calls. By default, this is assumed to be FALSE. If modified through this interface, it is stored persistently.

Note: The roaming mode does not affect a currently established data connection. For example, if the roaming mode is set to FALSE, but a roaming data call is connected (e.g., by a different client or because the mode was TRUE when the call was established), QCMAP_MSGR uses the currently established WWAN data connection.

Data fields

Type	Parameter	Description
uint32_t	mcm_- mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
uint8_t	allow_wwan_- calls_while_- roaming	Roaming mode. Indicates whether QCMAP_MSGR connects a data call while roaming; a Boolean value.

3.6.2.1.67. struct mcm_mobileap_set_roaming_pref_resp_msg_v01

Response message; Configures whether QCMAP_MSGR initiates WWAN data calls while roaming.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.68. struct mcm_mobileap_get_roaming_pref_req_msg_v01

Request message; Gets the roaming flag.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.69. struct mcm_mobileap_get_roaming_pref_resp_msg_v01

Response message; Gets the roaming flag.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.
uint8_t	allow_wwan_-calls_while_-roaming_valid	Must be set to TRUE if allow_wwan_calls_while_roaming is being passed.
uint8_t	allow_wwan_-calls_while_-roaming	Determines whether the mobile AP connects a data call while roaming; a Boolean value.

3.6.2.1.70. struct mcm_mobileap_set_dualap_config_req_msg_v01

Request message; Configures whether the mobile AP initiates WWAN data calls while roaming.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_MSGR_MOBILE_AP_ENABLE_REQ.
mcm_-mobileap_-dualap_config-_t_v01	dualap_config	Mobile AP dual SSID configuration.

3.6.2.1.71. struct mcm_mobileap_set_dualap_config_resp_msg_v01

Response message; Configures whether the mobile AP initiates WWAN data calls while roaming.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.72. struct mcm_mobileap_station_mode_enable_req_msg_v01

Request message; Enables Station (STA) mode functionality for a mobile AP instance on the modem.

After this request is successfully processed, all packet connectivity to an outside network occurs through the WLAN station. The modem routing engine appropriately handles the packet routing into and out of the modem.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.
mcm_-mobileap_-sta_connection-_config_t_v01	cfg	Station mode configuration to indicate dynamic or static IP configuration.

3.6.2.1.73. struct mcm_mobileap_station_mode_enable_resp_msg_v01

Response message; Enables STA mode functionality for a mobile AP instance on the modem.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.74. struct mcm_mobileap_station_mode_disable_req_msg_v01

Request message; Disables STA mode functionality for a mobile AP instance on the modem. When this request has been successfully processed, the control point invokes bringing up the WWAN from the mobile AP if auto-connect is enabled.

Data fields

Type	Parameter	Description
uint32_t	mcm_-mobileap_- handle	Handle identifying the mobile AP call instance. The value must be the handle previously returned by MCM_MOBILEAP_ENABLE_REQ.

3.6.2.1.75. struct mcm_mobileap_station_mode_disable_resp_msg_v01

Response message; Disables STA mode functionality for a mobile AP instance on the modem.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Result code.

3.6.2.1.76. struct mcm_mobileap_event_register_req_msg_v01

Request message; Registers for an indication of events.

Data fields

Type	Parameter	Description
uint8_t	register_event_-enabled_valid	Must be set to TRUE if register_event_enabled is being passed.
uint8_t	register_event_-enabled	Event registration is enabled.
uint8_t	register_event_-lan_connecting-_valid	Must be set to TRUE if register_event_lan_connecting is being passed.
uint8_t	register_event_-lan_connecting	Register for a LAN connecting event.
uint8_t	register_event_-lan_connecting-_fail_valid	Must be set to TRUE if register_event_lan_connecting_fail is being passed.

Type	Parameter	Description
uint8_t	register_event_-lan_connecting-_fail	Register for a LAN connection failure event.
uint8_t	register_event_-lan_ipv6_-connecting_-fail_valid	Must be set to TRUE if register_event_lan_ipv6_connecting_fail is being passed.
uint8_t	register_event_-lan_ipv6_-connecting_fail	Register for a LAN IPv6 connection failure event.
uint8_t	register_event_-lan_connected-_valid	Must be set to TRUE if register_event_lan_connected is being passed.
uint8_t	register_event_-lan_connected	Register for a LAN connected event.
uint8_t	register_event_-sta_connected-_valid	Must be set to TRUE if register_event_sta_connected is being passed.
uint8_t	register_event_-sta_connected	Register for a STA connected event.
uint8_t	register_event_-lan_ipv6_-connected_- valid	Must be set to TRUE if register_event_lan_ipv6_connected is being passed.
uint8_t	register_event_-lan_ipv6_-connected	Register for a LAN IPv6 connected event.
uint8_t	register_event_-wan_- connecting_-valid	Must be set to TRUE if register_event_wan_connecting is being passed.
uint8_t	register_event_-wan_- connecting	Register for a WAN connecting event.
uint8_t	register_event_-wan_- connecting_-fail_valid	Must be set to TRUE if register_event_wan_connecting_fail is being passed.

Type	Parameter	Description
uint8_t	register_event_-wan_-connecting_fail	Register for a WAN connection failure event.
uint8_t	register_event_-wan_ipv6_-connecting_-fail_valid	Must be set to TRUE if register_event_wan_ipv6_connecting_fail is being passed.
uint8_t	register_event_-wan_ipv6_-connecting_fail	Register for a WAN IPv6 connection failure event.
uint8_t	register_event_-wan_- connected_-valid	Must be set to TRUE if register_event_wan_connected is being passed.
uint8_t	register_event_-wan_connected	Register for a WAN connected event.
uint8_t	register_event_-wan_ipv6_-connected_- valid	Must be set to TRUE if register_event_wan_ipv6_connected is being passed.
uint8_t	register_event_-wan_ipv6_-connected	Register for a WAN IPv6 connected event.
uint8_t	register_event_-wan_-disconnected_-valid	Must be set to TRUE if register_event_wan_disconnected is being passed.
uint8_t	register_event_-wan_-disconnected	Register for a WAN disconnected event.
uint8_t	register_event_-wan_ipv6_-disconnected_-valid	Must be set to TRUE if register_event_wan_ipv6_disconnected is being passed.
uint8_t	register_event_-wan_ipv6_-disconnected	Register for a WAN IPv6 disconnected event.
uint8_t	register_event_-lan_-disconnected_-valid	Must be set to TRUE if register_event_lan_disconnected is being passed.

Type	Parameter	Description
uint8_t	register_event_-lan_- disconnected	Register for a LAN disconnected event.
uint8_t	register_event_-lan_ipv6_-disconnected_-valid	Must be set to TRUE if register_event_lan_ipv6_disconnected is being passed.
uint8_t	register_event_-lan_ipv6_-disconnected	Register for a LAN IPv6 disconnected event.
uint8_t	register_event_-disabled_valid	Must be set to TRUE if register_event_disabled is being passed.
uint8_t	register_event_-disabled	Event registration is disabled.

3.6.2.1.77. struct mcm_mobileap_event_register_resp_msg_v01

Response message; Registers for an indication of events.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.6.2.1.78. struct mcm_mobileap_unsol_event_ind_msg_v01

Request message; Indication corresponding to a registered unsolicited event.

Data fields

Type	Parameter	Description
int32_t	event_id	Event ID thats gets populated. Values: 0x8000L – MCM_MOBILEAP_ENABLED_EV 0x8001L – MCM_MOBILEAP_LAN_CONNECTING_EV 0x8002L – MCM_MOBILEAP_LAN_CONNECTING_FAIL_EV 0x8003L – MCM_MOBILEAP_LAN_IPv6_CONNECTING_FAIL_EV 0x8004L – MCM_MOBILEAP_LAN_CONNECTED_EV 0x8005L – MCM_MOBILEAP_STA_CONNECTED_EV 0x8006L – MCM_MOBILEAP_LAN_IPv6_CONNECTED_EV 0x8007L – MCM_MOBILEAP_WAN_CONNECTING_EV 0x8008L – MCM_MOBILEAP_WAN_CONNECTING_FAIL_EV 0x8009L – MCM_MOBILEAP_WAN_IPv6_CONNECTING_FAIL_EV 0x800AL – MCM_MOBILEAP_WAN_CONNECTED_EV 0x800BL – MCM_MOBILEAP_WAN_IPv6_CONNECTED_EV 0x800CL – MCM_MOBILEAP_WAN_DISCONNECTED_EV 0x800DL – MCM_MOBILEAP_WAN_IPv6_DISCONNECTED_EV 0x800EL – MCM_MOBILEAP_LAN_DISCONNECTED_EV 0x800FL – MCM_MOBILEAP_LAN_IPv6_DISCONNECTED_EV 0x8010L – MCM_MOBILEAP_DISABLED_EV

3.6.3. Mobile AP Constants

This section contains the MCM mobile access point constants.

3.6.3.1. Define Documentation

- #define MCM_MOBILEAP_ENABLED_EV_V01 0x8000 Enabled event.
- #define MCM_MOBILEAP_LAN_CONNECTING_EV_V01 0x8001 LAN connecting.
- #define MCM_MOBILEAP_LAN_CONNECTING_FAIL_EV_V01 0x8002 LAN connection failed.
- #define MCM_MOBILEAP_LAN_IPv6_CONNECTING_FAIL_EV_V01 0x8003 LAN IPv6 connection failed.
- #define MCM_MOBILEAP_LAN_CONNECTED_EV_V01 0x8004 LAN connected.
- #define MCM_MOBILEAP_STA_CONNECTED_EV_V01 0x8005 Station connected.
- #define MCM_MOBILEAP_LAN_IPv6_CONNECTED_EV_V01 0x8006 LAN IPv6 connected.
- #define MCM_MOBILEAP_WAN_CONNECTING_EV_V01 0x8007 WAN connecting.
- #define MCM_MOBILEAP_WAN_CONNECTING_FAIL_EV_V01 0x8008 WAN connection failed.
- #define MCM_MOBILEAP_WAN_IPv6_CONNECTING_FAIL_EV_V01 0x8009 WAN IPv6 connection failed.
- #define MCM_MOBILEAP_WAN_CONNECTED_EV_V01 0x800A WAN connected.

- #define MCM_MOBILEAP_WAN_IPv6_CONNECTED_EV_V01 0x800B WAN IPv6 connected.
- #define MCM_MOBILEAP_WAN_DISCONNECTED_EV_V01 0x800C WAN disconnected.
- #define MCM_MOBILEAP_WAN_IPv6_DISCONNECTED_EV_V01 0x800D WAN IPv6 disconnected.
- #define MCM_MOBILEAP_LAN_DISCONNECTED_EV_V01 0x800E LAN disconnected.
- #define MCM_MOBILEAP_LAN_IPv6_DISCONNECTED_EV_V01 0x800F LAN IPv6 disconnected.
- #define MCM_MOBILEAP_DISABLED_EV_V01 0x8010 Disabled event.
- #define MCM_MOBILEAP_MAX_FIREWALL_ENTRIES_V01 50 Maximum firewall entries.
- #define MCM_MOBILEAP_MAX_STATIC_NAT_ENTRIES_V01 50 Maximum static NAT entries.
- #define MCM_MOBILEAP_IPV6_ADDR_LEN_V01 16 IPv6 address length.
- #define MCM_MOBILEAP_MAC_ADDR_LEN_V01 6 MAC address length.
- #define MCM_MOBILEAP_DEVICE_NAME_MAX_V01 100 Maximum length of the device name.
- #define MCM_MOBILEAP_LEASE_TIME_LEN_V01 100 Maximum lease time length.
- #define MCM_MOBILEAP_MSG_TIMEOUT_VALUE_V01 90000 Maximum timeout for SYNC msgs (in milliseconds).

3.6.4. Mobile Ap Enumerations

This section contains the MCM mobile access point enums.

3.6.4.1. Enumeration Type Documentation

3.6.4.1.1. enum mcm_mobileap_nat_type_t_v01

Enumerator:

MCM_MOBILEAP_NAT_SYMMETRIC_NAT_V01 Symmetric NAT.

MCM_MOBILEAP_NAT_PORT_RESTRICTED_CONE_NAT_V01
Port-restricted cone NAT.

MCM_MOBILEAP_NAT_FULL_CONE_NAT_V01 Full cone NAT (currently not supported).

MCM_MOBILEAP_NAT_ADDRESS_RESTRICTED_NAT_V01 Address-restricted NAT (currently not supported).

3.6.4.1.2. enum mcm_mobileap_ip_family_t_v01

Enumerator:

MCM_MOBILEAP_IP_FAMILY_V4_V01 IP family v4.

MCM_MOBILEAP_IP_FAMILY_V6_V01 IP family v6 (currently not supported).

MCM_MOBILEAP_IP_FAMILY_V4V6_V01 IP family v4/v6.

3.6.4.1.3. enum mcm_mobileap_nat_timeout_t_v01

Enumerator:

MCM_MOBILEAP_NAT_TIMEOUT_GENERIC_V01
Generic NAT timeout.

MCM_MOBILEAP_NAT_TIMEOUT_ICMP_V01
NAT timeout for ICMP (currently not supported).

MCM_MOBILEAP_NAT_TIMEOUT_TCP_ESTABLISHED_V01
NAT timeout for the established TCP (currently not supported).

MCM_MOBILEAP_NAT_TIMEOUT_UDP_V01
NAT timeout for UDP (currently not supported).

3.6.4.1.4. enum mcm_mobileap_ip_version_t_v01

Enumerator:

MCM_MOBILEAP_IP_V4_V01 IPv4.

MCM_MOBILEAP_IP_V6_V01 IPv6.

3.6.4.1.5. enum mcm_mobileap_wwan_status_t_v01

Enumerator:

MCM_MOBILEAP_WWAN_STATUS_CONNECTING_V01
IPv4 WWAN is in the Connecting state.

MCM_MOBILEAP_WWAN_STATUS_CONNECTING_FAIL_V01
IPv4 connection to the WWAN failed.

MCM_MOBILEAP_WWAN_STATUS_CONNECTED_V01
IPv4 WWAN is in the Connected state.

MCM_MOBILEAP_WWAN_STATUS_DISCONNECTING_V01
IPv4 WWAN is disconnecting.

MCM_MOBILEAP_WWAN_STATUS_DISCONNECTING_FAIL_V01
IPv4 WWAN failed to disconnect.

MCM_MOBILEAP_WWAN_STATUS_DISCONNECTED_V01
IPv4 WWAN is disconnected.

MCM_MOBILEAP_WWAN_STATUS_IPV6_CONNECTING_V01
IPv6 WWAN is in the Connecting state.

MCM_MOBILEAP_WWAN_STATUS_IPV6_CONNECTING_FAIL_V01
IPv6 connection to the WWAN failed.

MCM_MOBILEAP_WWAN_STATUS_IPV6_CONNECTED_V01
IPv6 WWAN is in the Connected state.

MCM_MOBILEAP_WWAN_STATUS_IPV6_DISCONNECTING_V01
IPv6 WWAN is disconnecting.

MCM_MOBILEAP_WWAN_STATUS_IPV6_DISCONNECTING_FAIL_V01
IPv6 WWAN failed to disconnect.

MCM_MOBILEAP_WWAN_STATUS_IPV6_DISCONNECTED_V01
IPv6 WWAN is disconnected.

3.6.4.1.6. enum mcm_mobileap_wwan_call_end_type_t_v01

Enumerator:

MCM_MOBILEAP_WWAN_CALL_END_TYPE_INVALID_V01
Unknown.

MCM_MOBILEAP_WWAN_CALL_END_TYPE_MOBILE_IP_V01

MCM_MOBILEAP_WWAN_CALL_END_TYPE_INTERNAL_V01

MCM_MOBILEAP_WWAN_CALL_END_TYPE_CALL_MANAGER_DEFINED_V01
Call manager-defined.

MCM_MOBILEAP_WWAN_CALL_END_TYPE_3GPP_SPEC_DEFINED_V01
3GPP specification-defined.

MCM_MOBILEAP_WWAN_CALL_END_TYPE_PPP_V01
PPP.

MCM_MOBILEAP_WWAN_CALL_END_TYPE_EHRPD_V01
E-HRPD.

MCM_MOBILEAP_WWAN_CALL_END_TYPE_IPV6_V01
IPv6.

3.6.4.1.7. enum mcm_mobileap_sta_connection_t_v01

Enumerator:

> MCM_MOBILEAP_STA_CONNECTION_DYNAMIC_V01 Dynamic station connection. MCM_MOBILEAP_STA_CONNECTION_STATIC_V01 Static station connection.

3.6.5. Mobile AP Data Structures

This section contains the MCM mobile access point data structures.

3.6.5.1. Data Structure Documentation

3.6.5.1.1. struct mcm_mobileap_wwan_call_end_reason_t_v01

Data fields

Type	Parameter	Description
mcm_-mobileap_-wwan_call_-end_type_t_v01	wwan_call_-end_reason_- type	WWAN call end type.
int32_t	wwan_call_-end_reason_- code	WWAN call end reason code.

3.6.5.1.2. struct mcm_mobileap_ip4_addr_subnet_mask_t_v01

Data fields

Type	Parameter	Description
uint32_t	addr	IPv4 address as specified in the IPv4 protocol specification, RFC 791.
uint32_t	subnet_mask	IPv4 subnet mask as specified in the IPv4 protocol specification, RFC 791.

3.6.5.1.3. struct mcm_mobileap_ip6_addr_prefix_len_t_v01

Data fields

Type	Parameter	Description
uint8_t	addr	IPv6 address as specified in the IPv6 protocol specification, RFC 2460.
uint8_t	prefix_len	IPv6 prefix length as specified in the IPv6 protocol specification, RFC 3513.

3.6.5.1.4. struct mcm_mobileap_tcp_udp_port_range_t_v01

Data fields

Type	Parameter	Description
uint16_t	port	TCP/UDP port as specified in the TCP and UDP protocols, RFC 793 and RFC 768.
uint16_t	range	TCP/UDP port range as specified in the TCP and UDP protocols, RFC 793 and RFC 768.

3.6.5.1.5. struct mcm_mobileap_ip4_tos_t_v01

Data fields

Type	Parameter	Description
uint8_t	value	TOS value as specified in the IPv4 protocol, RFC 791.
uint8_t	mask	IPv4 TOS mask

3.6.5.1.6. struct mcm_mobileap_ip6_traffic_class_t_v01

Data fields

Type	Parameter	Description
uint8_t	value	IPv6 traffic class value as specified in the IPv6 protocol, RFC 2460.
uint8_t	mask	IPv6 traffic class mask

3.6.5.1.7. struct mcm_mobileap_static_nat_entry_conf_t_v01

Data fields

Type	Parameter	Description
uint32_t	port_fwding_-private_ip	Port forwarding private IP.
uint16_t	port_fwding_-private_port	Port forwarding private port.
uint16_t	port_fwding_-global_port	Port forwarding global IP.
uint8_t	port_fwding_-protocol	Port forwarding protocol.

3.6.5.1.8. struct mcm_mobileap_wwan_statistics_t_v01

Data fields

Type	Parameter	Description
uint64_t	bytes_rx	Number of Rx bytes.
uint64_t	bytes_tx	Number of Tx bytes.
uint32_t	pkts_rx	Number of Rx packets.
uint32_t	pkts_tx	Number of Tx packets.
uint32_t	pkts_dropped_-rx	Number of dropped Rx packets.
uint32_t	pkts_dropped_-tx	Number of dropped Tx packets.

3.6.5.1.9. struct mcm_mobileap_dhcpd_config_t_v01

Data fields

Type	Parameter	Description
uint32_t	intf	Interface.
uint32_t	start	Start.
uint32_t	end	End.
Char	leasetime	Lease time length.

3.6.5.1.10. struct mcm_mobileap_dualap_config_t_v01

Data fields

Type	Parameter	Description
uint8_t	enable	Enable dual AP.
uint32_t	a5_ip_address	A5 IP address.
uint32_t	sub_net_mask	Subnet mask.

3.6.5.1.11. struct mcm_mobileap_sta_static_ip_config_t_v01

Data fields

Type	Parameter	Description
uint32_t	ip_addr	IP address.
uint32_t	gw_ip	GW IP.
uint32_t	netmask	Net mask.
uint32_t	dns_addr	DNS address.

3.6.5.1.12. struct mcm_mobileap_sta_static_ip_config_t_v01

Data fields

Type	Parameter	Description
mcm_-mobileap_-sta_connection-_t_v01	conn_type	Connection type.
mcm_-mobileap_-sta_static_ip_-config_t_v01	static_ip_config	Static IP configuration.

3.7. Voice

This section provides the messages, constants, enumerations, and data structures for voice call processing, using MCM.

- Voice Message Identifiers
- Voice Message Structures
- Voice Constants
- Voice Enumerations
- Voice Data Structures

3.7.1. Voice Message Identifiers

This section contains the MCM voice message identifiers.

- #define MCM_VOICE_GET_CALLS_REQ_V01 0x1000
- #define MCM_VOICE_GET_CALLS_RESP_V01 0x1000
- #define MCM_VOICE_DIAL_REQ_V01 0x1001
- #define MCM_VOICE_DIAL_RESP_V01 0x1001
- #define MCM_VOICE_GET_CALL_STATUS_REQ_V01 0x1002
- #define MCM_VOICE_GET_CALL_STATUS_RESP_V01 0x1002
- #define MCM_VOICE_DTMF_REQ_V01 0x1003
- #define MCM_VOICE_DTMF_RESP_V01 0x1003
- #define MCM_VOICE_START_DTMF_REQ_V01 0x1004
- #define MCM_VOICE_START_DTMF_RESP_V01 0x1004
- #define MCM_VOICE_STOP_DTMF_REQ_V01 0x1005
- #define MCM_VOICE_STOP_DTMF_RESP_V01 0x1005
- #define MCM_VOICE_MUTE_REQ_V01 0x1006
- #define MCM_VOICE_MUTE_RESP_V01 0x1006
- #define MCM_VOICE_FLASH_REQ_V01 0x1007
- #define MCM_VOICE_FLASH_RESP_V01 0x1007
- #define MCM_VOICE_HANGUP_REQ_V01 0x1008
- #define MCM_VOICE_HANGUP_RESP_V01 0x1008
- #define MCM_VOICE_COMMAND_REQ_V01 0x1009

- #define MCM_VOICE_COMMAND_RESP_V01 0x1009
- #define MCM_VOICE_AUTO_ANSWER_REQ_V01 0x100A
- #define MCM_VOICE_AUTO_ANSWER_RESP_V01 0x100A
- #define MCM_VOICE_EVENT_REGISTER_REQ_V01 0x100B
- #define MCM_VOICE_EVENT_REGISTER_RESP_V01 0x100B
- #define MCM_VOICE_GET_CALL_FORWARDING_STATUS_REQ_V01 0x100C
- #define MCM_VOICE_GET_CALL_FORWARDING_STATUS_RESP_V01 0x100C
- #define MCM_VOICE_SET_CALL_FORWARDING_REQ_V01 0x100D
- #define MCM_VOICE_SET_CALL_FORWARDING_RESP_V01 0x100D
- #define MCM_VOICE_GET_CALL_WAITING_STATUS_REQ_V01 0x100E
- #define MCM_VOICE_GET_CALL_WAITING_STATUS_RESP_V01 0x100E
- #define MCM_VOICE_SET_CALL_WAITING_REQ_V01 0x100F
- #define MCM_VOICE_SET_CALL_WAITING_RESP_V01 0x100F
- #define MCM_VOICE_GET_CLIR_REQ_V01 0x1010
- #define MCM_VOICE_GET_CLIR_RESP_V01 0x1010
- #define MCM_VOICE_SET_CLIR_REQ_V01 0x1011
- #define MCM_VOICE_SET_CLIR_RESP_V01 0x1011
- #define MCM_VOICE_SET_FACILITY_LOCK_REQ_V01 0x1012
- #define MCM_VOICE_SET_FACILITY_LOCK_RESP_V01 0x1012
- #define MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REQ_V01 0x1013
- #define MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_RESP_V01 0x1013
- #define MCM_VOICE_SEND_USSD_REQ_V01 0x1014
- #define MCM_VOICE_SEND_USSD_RESP_V01 0x1014
- #define MCM_VOICE_CANCEL_USSD_REQ_V01 0x1015
- #define MCM_VOICE_CANCEL_USSD_RESP_V01 0x1015
- #define MCM_VOICE_COMMON_DIAL_REQ_V01 0x1016
- #define MCM_VOICE_COMMON_DIAL_RESP_V01 0x1016
- #define MCM_VOICE_CALL_IND_V01 0x1017
- #define MCM_VOICE_MUTE_IND_V01 0x1018

- #define MCM_VOICE_DTMF_IND_V01 0x1019
- #define MCM_VOICE_RECEIVE_USSD_IND_V01 0x101A
- #define MCM_VOICE_UPDATE_ECALL_MSD_REQ_V01 0x101B
- #define MCM_VOICE_UPDATE_ECALL_MSD_RESP_V01 0x101B
- #define MCM_VOICE_E911_STATE_IND_V01 0x101C
- #define MCM_VOICE_GET_E911_STATE_REQ_V01 0x101D
- #define MCM_VOICE_GET_E911_STATE_RESP_V01 0x101D

3.7.2. Voice Message Structures

This section contains the MCM voice message structures.

3.7.2.1. Data Structure Documentation

3.7.2.1.1. struct mcm_voice_get_calls_resp_msg_v01

Response message; Gets the list of current calls.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	calls_valid	Must be set to TRUE if calls is being passed.
uint32_t	calls_len	Must be set to the number of elements in calls.
mcm_voice_-call_record_t_- v01	calls	Calls.

3.7.2.1.2. struct mcm_voice_dial_req_msg_v01

Request message; Dials a call to a specified address and returns a connection ID.

Data fields

Type	Parameter	Description
uint8_t	address_valid	Must be set to TRUE if address is being passed.
char	address	End point address of the connection to make.
uint8_t	call_type_valid	Must be set to TRUE if call_type is being passed.
mcm_voice_-call_type_t_v01	call_type	Connection (call) details, or NULL.
uint8_t	uusdata_valid	Must be set to TRUE if uusdata is being passed.
mcm_voice_-uusdata_t_v01	uusdata	Token ID used to track this command; NULL is OK.
uint8_t	emergency_cat-_valid	Must be set to TRUE if emergency_cat is being passed.

Type	Parameter	Description
mcm_voice_-emergency_cat-_t_v01	emergency_cat	Emergency call category.
uint8_t	ecall_msd_-valid	Must be set to TRUE if ecall_msd is being passed.
uint32_t	ecall_msd_len	Must be set to the number of elements in ecall_msd.
uint8_t	ecall_msd	Minimum set of data. Only honored when call_type is MCM_VOICE_CALL_TYPE_ECALL_AUTO or MCM_VOICE_CALL_TYPE_ECALL_MANUAL. Ignored otherwise.

3.7.2.1.3. struct mcm_voice_dial_resp_msg_v01

Response message; Dials a call to a specified address and returns a connection ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	call_id_valid	Must be set to TRUE if call_id is being passed.
uint32_t	call_id	Call ID.

3.7.2.1.4. struct mcm_voice_get_call_status_req_msg_v01

Request message; Gets the status associated with the connection ID.

Data fields

Type	Parameter	Description
uint32_t	call_id	Call ID of the connection to query.

3.7.2.1.5. struct mcm_voice_get_call_status_resp_msg_v01

Response message; Gets the status associated with the connection ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	status_valid	Must be set to TRUE if status is being passed.
mcm_voice_-call_record_t_- v01	status	Call status.

3.7.2.1.6. struct mcm_voice_dtmf_req_msg_v01

Request message; Sends a DTMF character over the connection ID.

Data fields

Type	Parameter	Description
char	dtmf	DTMF character to be sent. Valid DTMF characters are 0-9, A-D, ’*’, ’#’.

3.7.2.1.7. struct mcm_voice_dtmf_resp_msg_v01

Response message; Sends a DTMF character over the connection ID.

Data fields

Type	Parameter	Description
mcm_response_t_v01	response	Result code.

3.7.2.1.8. struct mcm_voice_start_dtmf_req_msg_v01

Request message; Starts sending a DTMF character over the call ID.

Data fields

Type	Parameter	Description
uint32_t	call_id	Call ID.
char	digit	DTMF character to be sent. Valid DTMF characters are 0-9, A-D, ’*’, ’#’.

3.7.2.1.9. struct mcm_voice_start_dtmf_resp_msg_v01

Response message; Starts sending a DTMF character over the call ID.

Data fields

Type	Parameter	Description
mcm_response_t_v01	response	Result code.
uint8_t	call_id_valid	Must be set to TRUE if call_id is being passed.
uint32_t	call_id	Call ID.

3.7.2.1.10. struct mcm_voice_stop_dtmf_req_msg_v01

Request message; Stops sending a DTMF character over the call ID.

Data fields

Type	Parameter	Description
uint32_t	call_id	Call ID.

3.7.2.1.11. struct mcm_voice_stop_dtmf_resp_msg_v01

Response message; Stops sending a DTMF character over the call ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	call_id_valid	Must be set to TRUE if call_id is being passed
uint32_t	call_id	Call ID.

3.7.2.1.12. struct mcm_voice_mute_req_msg_v01

Request message; Mutes/unmutes a voice call.

Data fields

Type	Parameter	Description
uint32_t	call_id	Call ID of the connection to mute/unmute.
mcm_voice_-mute_type_t_- v01	mute_type	Mute or unmute the voice call.

3.7.2.1.13. struct mcm_voice_mute_resp_msg_v01

Response message; Mutes/unmutes a voice call.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.14. struct mcm_voice_flash_req_msg_v01

Request message; Sends a flash sequence character over the connection call ID.

Data fields

Type	Parameter	Description
char	sflash_string	A NULL-terminated flash string to be sent; Maximum 82 characters.

3.7.2.1.15. struct mcm_voice_flash_resp_msg_v01

Response message; Sends a flash sequence character over the connection call ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.16. struct mcm_voice_hangup_req_msg_v01

Request message; Hangs up or disconnects a voice call connection with the specified call ID.

Data fields

Type	Parameter	Description
uint32_t	call_id	Call ID associated with the connection.

3.7.2.1.17. struct mcm_voice_hangup_resp_msg_v01

Response message; Hangs up or disconnects a voice call connection with the specified call ID.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.18. struct mcm_voice_command_req_msg_v01

Request message; Provides various operations for a voice call.

Data fields

Type	Parameter	Description
mcm_voice_-call_operation-_t_v01	call_operation	Call operation.
uint8_t	call_id_valid	Must be set to TRUE if call_id is being passed.
uint32_t	call_id	Call ID.
uint8_t	cause_valid	Must be set to TRUE if cause is being passed.
uint32_t	cause	Cause.

3.7.2.1.19. struct mcm_voice_command_resp_msg_v01

Response message; Provides various operations for a voice call.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.20. struct mcm_voice_auto_answer_req_msg_v01

Request message; Enables/disables an incoming voice call.

Data fields

Type	Parameter	Description
mcm_voice_-auto_answer_-type_t_v01	auto_answer_-type	Auto-answer type.
uint8_t	anto_answer_-timer_valid	Must be set to TRUE if anto_answer_timer is being passed
uint32_t	anto_answer_-timer	Auto-answer timer.

3.7.2.1.21. struct mcm_voice_auto_answer_resp_msg_v01

Response message; Enables/disables an incoming voice call.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.22. struct mcm_voice_event_register_req_msg_v01

Request message; Registers for an indication of events.

Data fields

Type	Parameter	Description
uint8_t	register_voice_-call_event_- valid	Must be set to TRUE if register_voice_call_event is being passed.
uint8_t	register_voice_-call_event	Register for a voice call event indication.
uint8_t	register_mute_-event_valid	Must be set to TRUE if register_mute_event is being passed.
uint8_t	register_mute_-event	Register for a mute event indication.
uint8_t	register_dtmf_-event_valid	Must be set to TRUE if register_dtmf_event is being passed
uint8_t	register_dtmf_-event	Register for a DTMF event indication.
uint8_t	register_e911_-state_event_- valid	Must be set to TRUE if register_e911_state_event is being passed.
uint8_t	register_e911_-state_event	MCM_VOICE_E911_STATE_INDICATION.

3.7.2.1.23. struct mcm_voice_event_register_resp_msg_v01

Response message; Registers for an indication of events.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.24. struct mcm_voice_call_ind_msg_v01

Indication message; Indication for MCM_VOICE_CONNECTION_EV.

Data fields

Type	Parameter	Description
uint32_t	calls_len	Must be set to the number of elements in calls.
mcm_voice_-call_record_t_- v01	calls	Calls.

3.7.2.1.25. struct mcm_voice_mute_ind_msg_v01

Indication message; Indication for MCM_VOICE_MUTE_EV.

Data fields

Type	Parameter	Description
uint8_t	is_mute	Indicates whether a call is muted.

3.7.2.1.26. struct mcm_voice_dtmf_ind_msg_v01

Indication message; Indication for DTMF.

Data fields

Type	Parameter	Description
mcm_voice_-dtmf_info_t_- v01	dtmf_info	DTMF information.

3.7.2.1.27. struct mcm_voice_get_call_forwarding_status_req_msg_v01

Request message; Call forwarding status query.

Data fields

Type	Parameter	Description
mcm_-voice_call_-forwarding_-reason_t_v01	reason	Call forwarding reason.

3.7.2.1.28. struct mcm_voice_get_call_forwarding_status_resp_msg_v01

Response message; Call forwarding status query.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
mcm_-voice_call_-forwarding_-status_t_v01	status	Call forwarding status.
uint8_t	info_valid	Must be set to TRUE if info is being passed.
uint32_t	info_len	Must be set to the number of elements in info.
mcm_-voice_call_-forwarding_-info_t_v01	info	Call forwarding information.

3.7.2.1.29. struct mcm_voice_set_call_forwarding_req_msg_v01

Request message; Sets call forwarding.

Data fields

Type	Parameter	Description
mcm_voice_-call_service_t_-v01	fwdservice	Call forwarding service.
mcm_-voice_call_-forwarding_-reason_t_v01	reason	Call forwarding reason.
uint8_t	forwarding_-number_valid	Must be set to TRUE if forwarding_number is being passed.
char	forwarding_-number	Call forwarding number.

3.7.2.1.30. struct mcm_voice_set_call_forwarding_resp_msg_v01

Response message; Sets call forwarding.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.31. struct mcm_voice_get_call_waiting_status_resp_msg_v01

Response message; Call waiting status query.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
mcm_voice_-call_waiting_-service_t_v01	status	Call waiting status.

3.7.2.1.32. struct mcm_voice_set_call_waiting_req_msg_v01

Request message; Sets call waiting.

Data fields

Type	Parameter	Description
mcm_voice_-call_waiting_-service_t_v01	cwservice	Call waiting service.

3.7.2.1.33. struct mcm_voice_set_call_waiting_resp_msg_v01

Response message; Sets call waiting.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.34. struct mcm_voice_get_clir_resp_msg_v01

Response message; CLIR status query.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

mcm_voice_-clir_action_t_- v01	action	CLIR action.
mcm_-voice_clir_-presentation_t-_v01	presentation	CLIR presentation.

3.7.2.1.35. struct mcm_voice_set_clir_req_msg_v01

Request message; Set CLIR.

Data fields

Type	Parameter	Description
mcm_voice_-clir_action_t_- v01	clir_action	CLIR action.

3.7.2.1.36. struct mcm_voice_set_clir_resp_msg_v01

Response message; Set CLIR.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Response.

3.7.2.1.37. struct mcm_voice_set_facility_lock_req_msg_v01

Request message; Sets a facility lock.

Data fields

Type	Parameter	Description
mcm_voice_-facility_code_t-_v01	code	Facility code. Refer to 3GPP TS 27.997, Section 7.4.
mcm_voice_-facility_lock_-status_t_v01	status	Facility lock status.
char	password	Facility lock password.

3.7.2.1.38. struct mcm_voice_set_facility_lock_resp_msg_v01

Response message; Sets a facility lock.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.39. struct mcm_voice_change_call_barring_password_req_msg_v01

Request message; Changes the call barring password.

Data fields

Type	Parameter	Description
mcm_voice-_change_-call_barring-_password_-reason_t_v01	reason	Reason for the password change. Refer to 3GPP TS 27.997, Section 7.4.
char	old_password	Old password.
char	new_password	New password.

3.7.2.1.40. struct mcm_voice_change_call_barring_password_resp_msg_v01

Response message; Changes the call barring password.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.41. struct mcm_voice_send_ussd_req_msg_v01

Request message; Sends Unstructured Supplementary Service Data (USSD).

Data fields

Type	Parameter	Description
mcm_voice-_ussd_msg_-type_t_v01	type	USSD message type.
mcm_voice_-ussd_encoding-_t_v01	encoding	USSD encoding.
char	ussd_string	USSD string.

3.7.2.1.42. struct mcm_voice_send_ussd_resp_msg_v01

Response message; Sends Unstructured Supplementary Service Data (USSD).

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.43. struct mcm_voice_cancel_ussd_resp_msg_v01

Response message; Cancels USSD.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.44. struct mcm_voice_receive_ussd_ind_msg_v01

Indication message; Receives a USSD indication.

Data fields

Type	Parameter	Description
mcm_voice-_ussd_ind_-notification_t_- v01	notification	USSD indication notification.
char	ussd	USSD indication message.

3.7.2.1.45. struct mcm_voice_common_dial_req_msg_v01

Request message; Voice/SS/USSD common dial API.

Data fields

Type	Parameter	Description
char	request	Request.
uint8_t	call_type_valid	Must be set to TRUE if call_type is being passed.
mcm_voice_-call_type_t_v01	call_type	Connection (call) details, or NULL,
uint8_t	uusdata_valid	Must be set to TRUE if uusdata is being passed.
mcm_voice_-uusdata_t_v01	uusdata	Token ID used to track this command; NULL is OK.
uint8_t	emergency_cat-_valid	Must be set to TRUE if emergency_cat is being passed.
mcm_voice_-emergency_cat-_t_v01	emergency_cat	Emergency call category.

Type	Parameter	Description
uint8_t	ecall_msd_-valid	Must be set to TRUE if ecall_msd is being passed.
uint32_t	ecall_msd_len	Must be set to the number of elements in ecall_msd.
uint8_t	ecall_msd	Minimum set of data. Only honored when call_type is MCM_VOICE_CALL_TYPE_ECALL_AUTO or MCM_VOICE_CALL_TYPE_ECALL_MANUAL. Ignored otherwise.

3.7.2.1.46. struct mcm_voice_common_dial_resp_msg_v01

Response message; Voice/SS/USSD common dial API.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	req_changed_-to_type_valid	Must be set to TRUE if req_changed_to_type is being passed.
mcm_voice_-common_dial_-type_t_v01	req_changed_-to_type	If SS, check optional SS fields. If not present, the voice call went through.
uint8_t	call_id_valid	Must be set to TRUE if call_id is being passed.
uint32_t	call_id	Call ID.
uint8_t	ss_get_cf_-status_valid	Must be set to TRUE if ss_get_cf_status is being passed.
mcm_-voice_call_-forwarding_-status_t_v01	ss_get_cf_-status	Get the call forwarding status.
uint8_t	ss_get_cf_info-_valid	Must be set to TRUE if ss_get_cf_info is being passed.
uint32_t	ss_get_cf_info-_len	Must be set to the number of elements in ss_get_cf_info.
mcm_-voice_call_-forwarding_-info_t_v01	ss_get_cf_info	Call forwarding information.
uint8_t	ss_get_cw_-status_valid	Must be set to TRUE if ss_get_cw_status is being passed.
mcm_voice_-call_waiting_-service_t_v01	ss_get_cw_-status	Call waiting status.
uint8_t	ss_get_clir_-action_valid	Must be set to TRUE if ss_get_clir_action is being passed.
mcm_voice_-clir_action_t_- v01	ss_get_clir_-action	CLIR action.
uint8_t	ss_get_clir_-presentation_-valid	Must be set to TRUE if ss_get_clir_presentation is being passed.

Type	Parameter	Description
mcm_-voice_clir_-presentation_t-_v01	ss_get_clir_-presentation	CLIR presentation.

3.7.2.1.47. struct mcm_voice_update_msd_req_msg_v01

Request message; Update the minimum set of data (MSD) for an ongoing or subsequent eCall call.

Data fields

Type	Parameter	Description
uint8_t	ecall_msd_-valid	Must be set to TRUE if ecall_msd is being passed.
uint32_t	ecall_msd_len	Must be set to the number of elements in ecall_msd.
uint8_t	ecall_msd	Minimum set of data, in ASN.1 PER unaligned format. Only honored when enable_msd is set to TRUE.

3.7.2.1.48. struct mcm_voice_update_msd_resp_msg_v01

Response message; Update the MSD for an ongoing or subsequent eCall.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.

3.7.2.1.49. struct mcm_voice_e911_state_ind_msg_v01

Indication message; Indication for MCM_VOICE_E911_STATE_IND.

Data fields

Type	Parameter	Description
mcm_voice_-e911_state_t_- v01	e911_state	E911 state.

3.7.2.1.50. struct mcm_voice_get_e911_state_resp_msg_v01

Response message; Indication for MCM_VOICE_GET_E911_STATE.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	response	Result code.
uint8_t	e911_state_-valid	Must be set to TRUE if e911_state is being passed.
mcm_voice_-e911_state_t_- v01	e911_state	E911 state.

3.7.3. Voice Constants

This section contains the MCM voice constants.

3.7.3.1. Define Documentation

- #define MCM_MAX_VOICE_CALLS_V01 8
  GSM provides up to 8 calls; 3GPP2 provides 2.
- #define MCM_MAX_PHONE_NUMBER_V01 82
  Maximum length for a phone number or SIP URI (81 + NULL).
- #define MCM_MAX_UUS_DATA_V01 20
  Maximum user-to-user data.
- #define MCM_MAX_DTMF_LENGTH_V01 20
  Maximum DTMF length.
- #define MCM_MAX_USSD_LENGTH_V01 128
  Maximum USSD length.
- #define MCM_MAX_PASSWORD_LENGTH_V01 4
  Maximum password length.
- #define MCM_MAX_CALL_FORWARDING_INFO_V01 13
  Maximum call forwarding information.
- #define MCM_MAX_ECALL_MSD_V01 140
  Maximum size of the MSD sent to the network with an eCall.

3.7.4. Voice Enumerations

This section contains the MCM voice enums.

3.7.4.1. Enumeration Type Documentation

3.7.4.1.1. enum mcm_voice_call_state_t_v01

Enumerator:

MCM_VOICE_CALL_STATE_INCOMING_V01 MT incoming; CC setup.

MCM_VOICE_CALL_STATE_DIALING_V01 Dialing state.

MCM_VOICE_CALL_STATE_ALERTING_V01 MT call waiting; MO alterting.

MCM_VOICE_CALL_STATE_ACTIVE_V01 Call is active.

MCM_VOICE_CALL_STATE_HOLDING_V01 Call is on hold.

MCM_VOICE_CALL_STATE_END_V01 Call is disconnected.

MCM_VOICE_CALL_STATE_WAITING_V01 Call is waiting.

3.7.4.1.2. enum mcm_voice_call_type_t_v01

Enumerator:

MCM_VOICE_CALL_TYPE_NOT_SPECIFIED_V01 Placeholder for a zero value.
MCM_VOICE_CALL_TYPE_VOICE_V01 Voice call.
MCM_VOICE_CALL_TYPE_EMERGENCY_V01 Emergency call.
MCM_VOICE_CALL_TYPE_ECALL_AUTO_V01 Automatically triggered eCall.
MCM_VOICE_CALL_TYPE_ECALL_MANUAL_V01 Manually triggered eCall.

3.7.4.1.3. enum mcm_voice_call_direction_type_t_v01

Enumerator:

MCM_VOICE_CALL_MOBILE_ORIGINATED_V01 Mobile-originated.

MCM_VOICE_CALL_MOBILE_TERMINATED_V01 Mobile-terminated.

3.7.4.1.4. enum mcm_voice_call_number_presentation_type_t_v01

Enumerator:

MCM_VOICE_CALL_NUMBER_ALLOWED_V01 Number allowed.

MCM_VOICE_CALL_NUMBER_RESTRICTED_V01 Number restricted.

MCM_VOICE_CALL_NUMBER_PAYPHONE_V01 Payhone number.

3.7.4.1.5. enum mcm_voice_reason_t_v01

Enumerator:

MCM_VOICE_REASON_NONE_V01
Placeholder for a zero value.

MCM_VOICE_REASON_NORMAL_V01
Call ended normally.

MCM_VOICE_REASON_BUSY_V01
Call was rejected (busy).

MCM_VOICE_REASON_CONGESTION_V01
Network congestion.

MCM_VOICE_REASON_CALL_BARRED_V01
Incoming calls barred.

MCM_VOICE_REASON_FDN_BLOCKED_V01
Blocked by fixed dialing.

MCM_VOICE_REASON_DIAL_MODIFIED_TO_USSD_V01
Converted to a USSD message.

MCM_VOICE_REASON_DIAL_MODIFIED_TO_SS_V01
Converted to a SUP.

MCM_VOICE_REASON_DIAL_MODIFIED_TO_DIAL_V01
Converted to another call type.

MCM_VOICE_REASON_ACM_LIMIT_EXCEEDED_V01 No funds.

3.7.4.1.6. enum mcm_voice_call_operation_t_v01

Enumerator:

MCM_VOICE_CALL_ANSWER_V01 Answer the call.

MCM_VOICE_CALL_END_V01 End the call.

MCM_VOICE_CALL_HOLD_V01 Hold the call.

MCM_VOICE_CALL_UNHOLD_V01 Release the call from hold.

MCM_VOICE_CALL_CONFERENCE_V01 Conference call.

MCM_VOICE_CALL_GO_PRIVATE_V01 Private call.

MCM_VOICE_CALL_END_ALL_V01 End all calls.

3.7.4.1.7. enum mcm_voice_uus_type_t_v01

Enumerator:

MCM_VOICE_UUS_TYPE1_IMPLICIT_V01 Type 1 implicit.

MCM_VOICE_UUS_TYPE1_REQUIRED_V01 Type 1 required.

MCM_VOICE_UUS_TYPE1_NOT_REQUIRED_V01 Type 1 not required.

MCM_VOICE_UUS_TYPE2_REQUIRED_V01 Type 2 required.

MCM_VOICE_UUS_TYPE2_NOT_REQUIRED_V01 Type 2 not required.

MCM_VOICE_UUS_TYPE3_REQUIRED_V01 Type 3 required.

MCM_VOICE_UUS_TYPE3_NOT_REQUIRED_V01 Type 3 not required.

MCM_VOICE_UUS_TYPE_DATA_V01 Data.

3.7.4.1.8. enum mcm_voice_uus_dcs_type_t_v01

Enumerator:

MCM_VOICE_UUS_DCS_IA5_V01 IA5.

MCM_VOICE_UUS_DCS_OHLP_V01 OHLP.

MCM_VOICE_UUS_DCS_USP_V01 USP.

MCM_VOICE_UUS_DCS_X244_V01 x244.

3.7.4.1.9. enum mcm_voice_tech_t_v01

Enumerator:

MCM_VOICE_TECH_NONE_V01 None.

MCM_VOICE_TECH_3GPP_V01 3GPP.

MCM_VOICE_TECH_3GPP2_V01 3GPP2.

3.7.4.1.10. enum mcm_voice_call_forwarding_status_t_v01

Enumerator:

MCM_VOICE_CALL_FORWARDING_DISABLED_V01 Disabled.

MCM_VOICE_CALL_FORWARDING_ENABLED_V01 Enabled.

3.7.4.1.11. enum mcm_voice_call_forwarding_type_t_v01

Enumerator:

MCM_VOICE_CALL_FORWARDING_TYPE_VOICE_V01 Voice.

MCM_VOICE_CALL_FORWARDING_TYPE_DATA_V01 Data.

MCM_VOICE_CALL_FORWARDING_TYPE_VOICE_DATA_V01 Voice and data.

3.7.4.1.12. enum mcm_voice_call_waiting_service_t_v01

Enumerator:

MCM_VOICE_CALL_WAITING_VOICE_ENABLED_V01
Voice call waiting enabled.

MCM_VOICE_CALL_WAITING_DATA_ENABLED_V01
Data call waiting enabled.

MCM_VOICE_CALL_WAITING_VOICE_DATA_ENABLED_V01
Voice and data call waiting enabled.

MCM_VOICE_CALL_WAITING_DISABLED_V01
Voice call waiting disabled.

3.7.4.1.13. enum mcm_voice_call_service_t_v01

Enumerator:

MCM_VOICE_SERVICE_ACTIVATE_V01 Activate.

MCM_VOICE_SERVICE_DEACTIVATE_V01 Deactivate.

MCM_VOICE_SERVICE_REGISTER_V01 Register.

MCM_VOICE_SERVICE_ERASE_V01 Erase.

3.7.4.1.14. enum mcm_voice_call_forwarding_reason_t_v01

Enumerator:

MCM_VOICE_CALL_FORWARD_UNCONDITIONALLY_V01
Unconditional call forwarding.

MCM_VOICE_CALL_FORWARD_MOBILEBUSY_V01
Forward when the mobile device is busy.

MCM_VOICE_CALL_FORWARD_NOREPLY_V01
Forward when there is no reply.

MCM_VOICE_CALL_FORWARD_UNREACHABLE_V01
Forward when the call is unreachable.

MCM_VOICE_CALL_FORWARD_ALLFORWARDING_V01
All forwarding.

MCM_VOICE_CALL_FORWARD_ALLCONDITIONAL_V01
All conditional forwarding.

3.7.4.1.15. enum mcm_voice_clir_action_t_v01

Enumerator:

MCM_VOICE_CLIR_INVOCATION_V01 Invocation.

MCM_VOICE_CLIR_SUPPRESSION_V01 Suppression.

3.7.4.1.16. enum mcm_voice_clir_presentation_t_v01

Enumerator:

MCM_VOICE_CLIR_NOT_PROVISIONED_V01 Not provisioned.

MCM_VOICE_CLIR_PROVISIONED_PERMANENT_MODE_V01 Permanently provisioned.

MCM_VOICE_CLIR_PRESENTATION_RESTRICTED_V01 Restricted presentation.

MCM_VOICE_CLIR_PRESENTATION_ALLOWED_V01 Allowed presentation.

3.7.4.1.17. enum mcm_voice_facility_lock_status_t_v01

Enumerator:

MCM_VOICE_FACILITY_LOCK_ENABLE_V01 Enable.

MCM_VOICE_FACILITY_LOCK_DISABLE_V01 Disable.

3.7.4.1.18. enum mcm_voice_facility_code_t_v01

Enumerator:

MCM_VOICE_FACILITY_CODE_AO_V01
BAOC (Bar All Outgoing Calls) (refer to 3GPP TS 22.088, clause 1).

MCM_VOICE_FACILITY_CODE_OI_V01
BOIC (Bar Outgoing International Calls) (refer to 3GPP TS 22.088, clause 1).

MCM_VOICE_FACILITY_CODE_OX_V01
BOIC-exHC (Bar Outgoing International Calls except to Home Country) (refer to 3GPP TS 22.088, clause 1).

MCM_VOICE_FACILITY_CODE_AI_V01
BAIC (Bar All Incoming Calls) (refer 3GPP TS 22.088, clause 2).

MCM_VOICE_FACILITY_CODE_IR_V01
BIC-Roam (Bar Incoming Calls when Roaming outside the home country) (refer to 3GPP TS 22.088, clause 2).

MCM_VOICE_FACILITY_CODE_AB_V01
All barring services (refer to 3GPP TS 22.030) (applicable only for mode=0).

MCM_VOICE_FACILITY_CODE_AG_V01
All outgoing barring services (refer to 3GPP TS 22.030) (applicable only for mode=0).

MCM_VOICE_FACILITY_CODE_AC_V01
All incoming barring services (refer to 3GPP TS 22.030) (applicable only for mode=0).

3.7.4.1.19. enum mcm_voice_change_call_barring_password_reason_t_v01

Enumerator:

MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REASON_
ALLOUTGOING_V01
All outgoing.

MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REASON_
OUTGOINGINT_V01
Outgoing internal.

MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REASON_
OUTGOINGINTEXTOHOME_V01
Outgoing external to home.

MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REASON_
ALLINCOMING_V01
All incoming.

MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REASON_
INCOMINGROAMING_V01
Roaming incoming.

MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REASON_
ALLBARRING_V01
All calls are barred.

MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REASON_ALLOUTGOINGBARRING_V01
All outgoing calls are barred.

MCM_VOICE_CHANGE_CALL_BARRING_PASSWORD_REASON_ALLINCOMINGBARRING_V01
All incoming calls are barred.

3.7.4.1.20. enum mcm_voice_ussd_encoding_t_v01

Enumerator:

MCM_VOICE_USSD_ENCODING_ASCII_V01 ASCII coding scheme.

MCM_VOICE_USSD_ENCODING_8BIT_V01 8-bit coding scheme.

MCM_VOICE_USSD_ENCODING_UCS2_V01 UCS2.

3.7.4.1.21. enum mcm_voice_common_dial_type_t_v01

Enumerator:

MCM_VOICE_COMMON_DIAL_VOICE_V01 Voice.

MCM_VOICE_COMMON_DIAL_SS_V01 Supplementary service.

MCM_VOICE_COMMON_DIAL_USSD_V01 Unstructured supplementary service.

3.7.4.1.22. enum mcm_voice_mute_type_t_v01

Enumerator:

MCM_VOICE_MUTE_V01 Mute.

MCM_VOICE_UNMUTE_V01 Unmute.

3.7.4.1.23. enum mcm_voice_auto_answer_type_t_v01

Enumerator:

MCM_VOICE_AUTO_ANSWER_ENABLE_V01 Enable auto-answer.

MCM_VOICE_AUTO_ANSWER_DISABLE_V01 Disable auto-answer.

3.7.4.1.24. enum mcm_voice_dtmf_event_type_t_v01

Enumerator:

MCM_VOICE_DTMF_EVENT_BURST_V01 Burst DTMF.

MCM_VOICE_DTMF_EVENT_START_CONT_V01 Continuous DTMF start.

MCM_VOICE_DTMF_EVENT_STOP_CONT_V01 Continuous DTMF stop.

3.7.4.1.25. enum mcm_voice_ussd_msg_type_t_v01

Enumerator:

MCM_VOICE_USSD_MSG_TYPE_NEW_MESSAGE_V01
Initiate a new USSD sesion with the network.

MCM_VOICE_USSD_MSG_TYPE_REPLY_TO_IND_V01
Reply to a USSD indication from the network.

3.7.4.1.26. enum mcm_voice_ussd_ind_notification_t_v01

Enumerator:

MCM_VOICE_USSD_INDICATION_FURTHER_ACTION_NOT_REQUIRED_V01
USSD indication requires a USSD reply.

MCM_VOICE_USSD_INDICATION_FURTHER_ACTION_REQUIRED_V01
USSD indication does not require a reply.

3.7.4.1.27. enum mcm_voice_emergency_cat_t_v01

Enumerator:

MCM_VOICE_EMER_CAT_POLICE_V01 Police.

MCM_VOICE_EMER_CAT_AMBULANCE_V01 Ambulance.

MCM_VOICE_EMER_CAT_FIRE_BRIGADE_V01 Fire bridge

MCM_VOICE_EMER_CAT_MARINE_GUARD_V01 Marine guard

MCM_VOICE_EMER_CAT_MOUNTAIN_RESCUE_V01 Mountain rescue.

3.7.4.1.28. enum mcm_voice_e911_state_t_v01

Enumerator:

MCM_VOICE_E911_INACTIVE_V01 E911 INACTIVE.

MCM_VOICE_E911_ACTIVE_V01 E911 ACTIVE.

3.7.5. Voice Data Structures

This section contains the MCM voice data structures.

3.7.5.1. Data Structure Documentation

3.7.5.1.1. struct mcm_voice_uusdata_t_v01

Data fields

Type	Parameter	Description
mcm_voice_-uus_type_t_v01	type	UUS type; range – 0 to 6.
mcm_voice_-uus_dcs_type_-t_v01	dcs	UUS data coding scheme; range – 0 to 4.
uint32_t	uus_data_len	Must be set to the number of elements in uus_data.
uint8_t	uus_data	Voice call UUS data.

3.7.5.1.2. struct mcm_voice_call_record_t_v01

Data fields

Type	Parameter	Description
uint32_t	call_id	Call ID associated with this call.
mcm_voice_-call_state_t_- v01	state	Current call state (mcm_voice_call_state).
mcm_voice_-tech_t_v01	tech	Technology (mcm_tech).
char	number	Phone number.
mcm_voice_-call_number_-presentation_-type_t_v01	number_-presentation	Number presentation.
mcm_voice_-call_direction_-type_t_v01	direction	Voice call direction.
uint8_t	uusdata_valid	Indicates whether UUS data is valid.
mcm_voice_-uusdata_t_v01	uusdata	User-to-user signaling data.

3.7.5.1.3. struct mcm_voice_dtmf_info_t_v01

Data fields

Type	Parameter	Description
uint32_t	call_id	Call ID associated with this DTMF event.
mcm_voice_-dtmf_event_-type_t_v01	dtmf_event	DTMF event type.
uint32_t	digit_len	Must be set to the number of elements in digit.
char	digit	DTMF character.

3.7.5.1.4. struct mcm_voice_call_forwarding_info_t_v01

Data fields

Type	Parameter	Description
mcm_-voice_call_-forwarding_-type_t_v01	type	Call forwarding type.
char	number	Call forwarding number.

3.8. Subscriber Identity Module

This chapter provides the messages, constants, enumerations, and data structures for managing the Subscriber Identity Module (SIM), using MCM

- SIM Message Identifiers
- SIM Message Structures
- SIM Constants
- SIM Enumerations
- SIM Data Structures

3.8.1. SIM Message Identifiers

This section contains the MCM SIM message identifiers.

- #define MCM_SIM_GET_SUBSCRIBER_ID_REQ_V01 0x0B00
- #define MCM_SIM_GET_SUBSCRIBER_ID_RESP_V01 0x0B00
- #define MCM_SIM_GET_CARD_ID_REQ_V01 0x0B01
- #define MCM_SIM_GET_CARD_ID_RESP_V01 0x0B01
- #define MCM_SIM_GET_DEVICE_PHONE_NUMBER_REQ_V01 0x0B02
- #define MCM_SIM_GET_DEVICE_PHONE_NUMBER_RESP_V01 0x0B02
- #define MCM_SIM_GET_PREFERRED_OPERATOR_LIST_REQ_V01 0x0B03
- #define MCM_SIM_GET_PREFERRED_OPERATOR_LIST_RESP_V01 0x0B03
- #define MCM_SIM_READ_FILE_REQ_V01 0x0B04
- #define MCM_SIM_READ_FILE_RESP_V01 0x0B04
- #define MCM_SIM_WRITE_FILE_REQ_V01 0x0B05

- #define MCM_SIM_WRITE_FILE_RESP_V01 0x0B05
- #define MCM_SIM_GET_FILE_SIZE_REQ_V01 0x0B06
- #define MCM_SIM_GET_FILE_SIZE_RESP_V01 0x0B06
- #define MCM_SIM_VERIFY_PIN_REQ_V01 0x0B07
- #define MCM_SIM_VERIFY_PIN_RESP_V01 0x0B07
- #define MCM_SIM_CHANGE_PIN_REQ_V01 0x0B08
- #define MCM_SIM_CHANGE_PIN_RESP_V01 0x0B08
- #define MCM_SIM_UNBLOCK_PIN_REQ_V01 0x0B09
- #define MCM_SIM_UNBLOCK_PIN_RESP_V01 0x0B09
- #define MCM_SIM_ENABLE_PIN_REQ_V01 0x0B0A
- #define MCM_SIM_ENABLE_PIN_RESP_V01 0x0B0A
- #define MCM_SIM_DISABLE_PIN_REQ_V01 0x0B0B
- #define MCM_SIM_DISABLE_PIN_RESP_V01 0x0B0B
- #define MCM_SIM_GET_CARD_STATUS_REQ_V01 0x0B0C
- #define MCM_SIM_GET_CARD_STATUS_RESP_V01 0x0B0C
- #define MCM_SIM_DEPERSONALIZATION_REQ_V01 0x0B0D
- #define MCM_SIM_DEPERSONALIZATION_RESP_V01 0x0B0D
- #define MCM_SIM_PERSONALIZATION_REQ_V01 0x0B0E
- #define MCM_SIM_PERSONALIZATION_RESP_V01 0x0B0E
- #define MCM_SIM_EVENT_REGISTER_REQ_V01 0x0B0F
- #define MCM_SIM_EVENT_REGISTER_RESP_V01 0x0B0F
- #define MCM_SIM_CARD_STATUS_EVENT_IND_V01 0x0B10
- #define MCM_SIM_REFRESH_EVENT_IND_V01 0x0B11

3.8.2. SIM Message Structures

This section contains the MCM SIM message structures.

3.8.2.1. Data Structure Documentation

3.8.2.1.1. struct mcm_sim_get_subscriber_id_req_msg_v01

Request message; Retrieves the International Mobile Subscriber Identity (IMSI) value stored in the specified application.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.

3.8.2.1.2. struct mcm_sim_get_subscriber_id_resp_msg_v01

Response message; Retrieves the International Mobile Subscriber Identity (IMSI) value stored in the specified application.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	imsi_valid	Must be set to TRUE if imsi is being passed.
uint32_t	imsi_len	Must be set to the number of elements in imsi.
char	imsi	IMSI data in ASCII characters.

3.8.2.1.3. struct mcm_sim_get_card_id_req_msg_v01

Request message; Retrieves the Integrated Circuit Card ID (ICCID) stored on the card.

Data fields

Type	Parameter	Description
mcm_sim_slot-_id_t_v01	slot_id	Indicates the slot to be used. Valid values: 1 – Slot 1 2 – Slot 2

3.8.2.1.4. struct mcm_sim_get_card_id_resp_msg_v01

Response message; Retrieves the Integrated Circuit Card ID (ICCID) stored on the card.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	iccid_valid	Must be set to TRUE if iccid is being passed.
uint32_t	iccid_len	Must be set to the number of elements in iccid.
char	iccid	ICCID data in ASCII characters.

3.8.2.1.5. struct mcm_sim_get_device_phone_number_req_msg_v01

Request message; Retrieves the device phone number stored on the card.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.

3.8.2.1.6. struct mcm_sim_get_device_phone_number_resp_msg_v01

Response message; Retrieves the device phone number stored on the card.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	phone_number-_valid	Must be set to TRUE if phone_number is being passed.
uint32_t	phone_number-_len	Must be set to the number of elements in phone_number.
char	phone_number	Parsed phone number in ASCII characters.

3.8.2.1.7. struct mcm_sim_get_preferred_operator_list_req_msg_v01

Request message; Retrieves the preferred operator list stored on the card.

Note: This command is only supported by 3GPP applications.

Data fields

Type	Parameter	Description
mcm_sim_slot-_id_t_v01	slot_id	Indicates the slot to be used. Valid values: 1 – Slot 1 2 – Slot 2

3.8.2.1.8. struct mcm_sim_get_preferred_operator_list_resp_msg_v01

Response message; Retrieves the preferred operator list stored on the card.

Note: This command is only supported by 3GPP applications.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	preferred_-operator_list_-valid	Must be set to TRUE if preferred_operator_list is being passed.
uint32_t	preferred_-operator_list_- len	Must be set to the number of elements in preferred_operator_list.
mcm_sim_-plmn_t_v01	preferred_-operator_list	Preferred operator list.

3.8.2.1.9. struct mcm_sim_read_file_req_msg_v01

Request message; Reads data to a specific file on a specified application on the card. The type of file is determined by the record number field, which indicates a transparent file when zero and a record-based file otherwise.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
mcm_sim_file-_access_t_v01	file_access	File access information.

3.8.2.1.10. struct mcm_sim_read_file_resp_msg_v01

Response message; Reads data to a specific file on a specified application on the card. The response contains the status code received from the card (SW1 and SW2) when the card responded to the read request, as well as the data that was read from the file.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	card_result_-valid	Must be set to TRUE if card_result is being passed.
mcm_sim_-card_result_t_- v01	card_result	Card result.
uint8_t	data_valid	Must be set to TRUE if data is being passed.
uint32_t	data_len	Must be set to the number of elements in data.
uint8_t	data	Data retrieved from the card.

3.8.2.1.11. struct mcm_sim_write_file_req_msg_v01

Request message; Writes data to a specific file on a specified application on the card. The type of file is determined by the record number field, which indicates a transparent file when zero and a record-based file otherwise.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
mcm_sim_file-_access_t_v01	file_access	File access information.
uint32_t	data_len	Must be set to the number of elements in data.
uint8_t	data	Data to be updated on the card.

3.8.2.1.12. struct mcm_sim_write_file_resp_msg_v01

Response message; Writes data to a specific file on a specified application on the card. The response contains the status code received from the card (SW1 and SW2) when the card responded to the write request.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	card_result_-valid	Must be set to TRUE if card_result is being passed.
mcm_sim_-card_result_t_- v01	card_result	Card result.

3.8.2.1.13. struct mcm_sim_get_file_size_req_msg_v01

Request message; Retrieves the size of a specific file on a specified application on the card.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
uint32_t	path_len	Must be set to the number of elements in path.
char	path	File path in ASCII characters.

3.8.2.1.14. struct mcm_sim_get_file_size_resp_msg_v01

Response message; Retrieves the size of a specific file on a specified application on the card. The response contains the status code received from the card (SW1 and SW2) when the card responded to the get file size request. The response also contains the type of file associated with the size.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	card_result_-valid	Must be set to TRUE if card_result is being passed.
mcm_sim_-card_result_t_- v01	card_result	Card result.
uint8_t	file_info_valid	Must be set to TRUE if file_info is being passed.
mcm_sim_file-_info_t_v01	file_info	File information.

3.8.2.1.15. struct mcm_sim_verify_pin_req_msg_v01

Request message; Verifies the PIN value of an application. The same PIN can be used by multiple sessions (i.e., the PIN is shared between GSM and RUIM in an ICC card). The PIN is automatically verified for all the sessions when the command is executed.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
mcm_sim_pin-_id_t_v01	pin_id	PIN ID.
uint32_t	pin_value_len	Must be set to the number of elements in pin_value.
char	pin_value	PIN value.

3.8.2.1.16. struct mcm_sim_verify_pin_resp_msg_v01

Response message; Verifies the PIN value of an application.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.

Type	Parameter	Description
uint8_t	retries_left_-valid	Must be set to TRUE if retries_left is being passed.
uint8_t	retries_left	Retries remaining.

3.8.2.1.17. struct mcm_sim_change_pin_req_msg_v01

Request message; Changes the PIN value of an application. The application must pass both the new and the old values of the PIN to complete the operation.

The same PIN can be used by multiple sessions (i.e., the PIN is shared between GSM and RUIM in an ICC card). The PIN is automatically verified for all the sessions when the command is executed.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
mcm_sim_pin-_id_t_v01	pin_id	PIN ID.
uint32_t	old_pin_value-_len	Must be set to the number of elements in old_pin_value.
char	old_pin_value	Value of the old PIN as a sequence of ASCII characters.
uint32_t	new_pin_value-_len	Must be set to the number of elements in new_pin_value.
char	new_pin_value	Value of the new PIN as a sequence of ASCII characters.

3.8.2.1.18. struct mcm_sim_change_pin_resp_msg_v01

Response message; Changes the PIN value of an application.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	retries_left_-valid	Must be set to TRUE if retries_left is being passed.
uint8_t	retries_left	Retries remaining.

3.8.2.1.19. struct mcm_ sim_unblock_pin_resp_msg_v01

Request message; Unblocks a blocked PIN using the PUK code. The client must pass PUK1 to unblock PIN1 or PUK2 to unblock PIN2.

The same PIN can be used by multiple sessions (i.e., the PIN is shared between GSM and RUIM in an ICC card). The PIN is automatically verified for all the sessions when the command is executed.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
mcm_sim_pin-_id_t_v01	pin_id	PIN ID.
uint32_t	puk_value_len	Must be set to the number of elements in puk_value.
char	puk_value	Value of the PUK as a sequence of ASCII characters.
uint32_t	new_pin_value-_len	Must be set to the number of elements in new_pin_value.
char	new_pin_value	Value of the new PIN as a sequence of ASCII characters.

3.8.2.1.20. struct mcm_sim_unblock_pin_resp_msg_v01

Response message; Unblocks a blocked PIN using the PUK code.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	retries_left_-valid	Must be set to TRUE if retries_left is being passed.
uint8_t	retries_left	Retries remaining.

3.8.2.1.21. struct mcm_sim_enable_pin_req_msg_v01

Request message; Enables the PIN on an application.

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
mcm_sim_pin-_id_t_v01	pin_id	PIN ID.
uint32_t	pin_value_len	Must be set to the number of elements in pin_value.
char	pin_value	Value of the PIN as a sequence of ASCII characters.

3.8.2.1.22. struct mcm_sim_enable_pin_resp_msg_v01

Response message; Enables the PIN on an application.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	retries_left_-valid	Must be set to TRUE if retries_left is being passed.
uint8_t	retries_left	Retries remaining.

3.8.2.1.23. struct mcm_sim_disable_pin_req_msg_v01

Request message; Enables or disables the PIN of an application,

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
mcm_sim_pin-_id_t_v01	pin_id	PIN ID.
uint32_t	pin_value_len	Must be set to the number of elements in pin_value
char	pin_value	Value of the PIN as a sequence of ASCII characters.

3.8.2.1.24. struct mcm_sim_disable_pin_resp_msg_v01

Response message; Enables or disables the PIN of an application,

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	retries_left_-valid	Must be set to TRUE if retries_left is being passed.
uint8_t	retries_left	Retries remaining.

3.8.2.1.25. struct mcm_sim_depersonalization_req_msg_v01

Request message; Deactivates or unblocks the personalization on the phone. Each feature can be deactivated/unblocked independently of the other features.

Data fields

Type	Parameter	Description
mcm_sim_-depersonalization-_t_v01	depersonaliza-tion	Depersonalization.

3.8.2.1.26. struct mcm_sim_depersonalization_resp_msg_v01

Response message; Deactivates or unblocks the personalization on the phone.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	retries_left_-valid	Must be set to TRUE if retries_left is being passed.
mcm_sim_-perso_retries_-left_t_v01	retries_left	Retries remaining.

3.8.2.1.27. struct mcm_sim_personalization_req_msg_v01

Request message; Activates and sets the personalization data on the phone. Each feature can be activated independently of one another; however, network data configurations must be consistent across activated personalization modes in order to prevent contradicting featurization, and only one feature can be activated per message.

If personalization is already activated, it must first be deactivated before being reactivated with new data.

Data fields

Type	Parameter	Description
uint32_t	ck_value_len	Must be set to the number of elements in ck_value.
char	ck_value	Control key value. This value is a sequence of ASCII characters.
uint8_t	feature_gw_-network_perso-_valid	Must be set to TRUE if feature_gw_network_perso is being passed.
uint32_t	feature_gw_-network_perso-_len	Must be set to the number of elements in feature_gw_network_perso.
mcm_sim_-network_perso-_t_v01	feature_gw_-network_perso	GW network personalization.
uint8_t	feature_gw_-network_-subset_perso_-valid	Must be set to TRUE if feature_gw_network_subset_perso is being passed.
uint32_t	feature_gw_-network_-subset_perso_- len	Must be set to the number of elements in feature_gw_network_subset_perso.
mcm_sim_-gw_network_-subset_perso_t-_v01	feature_gw_-network_-subset_perso	GW network subset personalization.
uint8_t	feature_gw_sp-_perso_valid	Must be set to TRUE if feature_gw_sp_perso is being passed.
uint32_t	feature_gw_sp-_perso_len	Must be set to the number of elements in feature_gw_sp_perso.
mcm_sim_gw-_sp_perso_t_- v01	feature_gw_sp-_perso	GW service provider personalization.
uint8_t	feature_gw_-corporate_-perso_valid	Must be set to TRUE if feature_gw_corporate_perso is being passed.
uint32_t	feature_gw_-corporate_-perso_len	Must be set to the number of elements in feature_gw_corporate_perso.
mcm_sim_gw-_corporate_-perso_t_v01	feature_gw_-corporate_perso	GW corporate personalization.
uint8_t	feature_gw_-sim_perso_- valid	Must be set to TRUE if feature_gw_sim_perso is being passed.
uint32_t	feature_gw_-sim_perso_len	Must be set to the number of elements in feature_gw_sim_perso.
mcm_sim_sim-_perso_t_v01	feature_gw_-sim_perso	GW SIM personalization.

Type	Parameter	Description
uint8_t	feature_1x_-network1_-perso_valid	Must be set to TRUE if feature_1x_network1_perso is being passed.
uint32_t	feature_1x_-network1_-perso_len	Must be set to the number of elements in feature_1x_network1_perso.
mcm_sim_-network_perso-_t_v01	feature_1x_-network1_perso	1X network type 1 personalization.
uint8_t	feature_1x_-network2_-perso_valid	Must be set to TRUE if feature_1x_network2_perso is being passed.
uint32_t	feature_1x_-network2_-perso_len	Must be set to the number of elements in feature_1x_network2_perso.
mcm_sim_1x_-network_type2-_perso_t_v01	feature_1x_-network2_perso	1X network type 3 personalization.
uint8_t	feature_1x_-ruim_perso_- valid	Must be set to TRUE if feature_1x_ruim_perso is being passed.
uint32_t	feature_1x_-ruim_perso_len	Must be set to the number of elements in feature_1x_ruim_perso.
mcm_sim_sim-_perso_t_v01	feature_1x_-ruim_perso	1X RUIM personalization.

3.8.2.1.28. struct mcm_sim_personalization_resp_msg_v01

Response message; Activates and sets the personalization data on the phone.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: qmi_result_type – QMI_RESULT_SUCCESS or QMI_RESULT_FAILURE qmi_error_type – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	retries_left_-valid	Must be set to TRUE if retries_left is being passed.
mcm_sim_-perso_retries_-left_t_v01	retries_left	This value is returned only when activation and setting personalization data fails.

3.8.2.1.29. struct mcm_sim_get_card_status_req_msg_v01

Request message; Retrieves the card status stored on a card.

Data fields

Type	Parameter	Description
mcm_sim_slot-_id_t_v01	slot_id	Slot ID.

3.8.2.1.30. struct mcm_sim_get_card_status_resp_msg_v01

Response message; Retrieves the card status stored on a card. The result of this function can be used by the client to determine the number of slots supported by the specific target.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.
uint8_t	card_info_valid	Must be set to TRUE if card_info is being passed.
mcm_sim_-card_info_t_- v01	card_info	Card information.

3.8.2.1.31. struct mcm_sim_event_register_req_msg_v01

Request message; Registers/deregisters for unsolicited SIM event indications.

Data fields

Type	Parameter	Description
uint8_t	register_card_-status_event_-valid	Must be set to TRUE if register_card_status_event is being passed.
uint8_t	register_card_-status_event	Register for card status events.
uint8_t	register_-refresh_event_-valid	Must be set to TRUE if register_refresh_event is being passed.
uint8_t	register_-refresh_event	Register for refresh events.

3.8.2.1.32. struct mcm_sim_event_register_resp_msg_v01

Response message; Registers/deregisters for unsolicited SIM event indications. The client is notified only when any file that belongs to the requested session type is modified by the Refresh procedure. The client does not need to specify a list of files.

The client can deregister from card status and/or refresh events by indicating a FALSE Boolean value.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Standard response type. Contains the following data members: mcm_result_t – MCM_RESULT_SUCCESS or MCM_RESULT_FAILURE mcm_error_t – Error code. Possible error code values are described in the error codes section of each message definition.

3.8.2.1.33. struct mcm_sim_card_status_event_ind_msg_v01

Indication message; Indication corresponding to MCM_SIM_CARD_STATUS_EVENT_IND.

Data fields

Type	Parameter	Description
uint8_t	slot_id_valid	Must be set to TRUE if slot_id is being passed.
mcm_sim_slot-_id_t_v01	slot_id	Identifies to which card the indication corresponds.
uint8_t	card_info_valid	Must be set to TRUE if card_info is being passed.
mcm_sim_-card_info_t_- v01	card_info	Card information indication.

3.8.2.1.34. struct mcm_sim_refresh_event_ind_msg_v01

Indication message; Indication corresponding to MCM_SIM_REFRESH_EVENT_IND.

Data fields

Type	Parameter	Description
uint8_t	refresh_event_-valid	Must be set to TRUE if refresh_event is being passed.
mcm_sim_-refresh_event_-t_v01	refresh_event	Refresh indication information.

3.8.3. SIM Constants

This section contains the MCM SIM constants.

3.8.3.1. Define Documentation

- #define MCM_SIM_IMSI_LEN_V01 16
  Maximum length of IMSI data.
- #define MCM_SIM_ICCID_LEN_V01 20
- Maximum length of ICCID data.
- #define MCM_SIM_NUM_PLMN_MAX_V01 24
  Maximum number of PLMN data sets.
- #define MCM_SIM_CHAR_PATH_MAX_V01 20
  Maximum length of a full file path in ASCII format.
- #define MCM_SIM_DATA_MAX_V01 4096
  Maximum size of data to be read/written.
- #define MCM_SIM_PIN_MAX_V01 8
  Maximum length of PIN data.
- #define MCM_SIM_MAX_NUM_CARDS_V01 2
  Maximum number of cards.
- #define MCM_SIM_MAX_REFRESH_FILES_V01 35
  Maximum number of refresh files.
- #define MCM_SIM_MAX_BINARY_PATHS_V01 10
  Maximum length of a full file path in binary format.
- #define MCM_SIM_CK_MAX_V01 16
  Maximum length of personalization control key data
- #define MCM_SIM_MCC_LEN_V01 3
  Length of the MCC.
- #define MCM_SIM_MNC_MAX_V01 3
  Maximum length of the MNC.
- #define MCM_SIM_PHONE_NUMBER_MAX_V01 82
  Maximum phone number length.
- #define MCM_SIM_IRM_CODE_LEN_V01 4
  Length of the IRM code.

- #define MCM_SIM_MSIN_MAX_V01 10
  Maximum length of the MSIN.
- #define MCM_SIM_PERSO_NUM_NW_MAX_V01 85
  Maximum number of network personalization data sets.
- #define MCM_SIM_PERSO_NUM_NS_MAX_V01 64
  Maximum number of network subset personalization data sets.
- #define MCM_SIM_PERSO_NUM_GW_SP_MAX_V01 64
  Maximum number of service provider personalization data sets.
- #define MCM_SIM_PERSO_NUM_GW_CP_MAX_V01 51
  Maximum number of corporate personalization data sets.
- #define MCM_SIM_PERSO_NUM_SIM_MAX_V01 32
  Maximum number of SIM personalization data sets.
- #define MCM_SIM_PERSO_NUM_1X_NW2_MAX_V01 128
  Maximum number of network type 2 personalization data sets.

3.8.4. SIM Enumerations

This section contains the MCM SIM enums.

3.8.4.1. Enumeration Type Documentation

3.8.4.1.1. enum mcm_sim_slot_id_t_v01

Enumerator:

MCM_SIM_SLOT_ID_1_V01	Identify card in slot 1.
MCM_SIM_SLOT_ID_2_V01	Identify card in slot 2.

3.8.4.1.2. enum mcm_sim_app_type_t_v01

Enumerator:

MCM_SIM_APP_TYPE_UNKNOWN_V01 Unknown application type.
MCM_SIM_APP_TYPE_3GPP_V01 Identify the SIM/USIM application on the card.
MCM_SIM_APP_TYPE_3GPP2_V01 Identify the RUIM/CSIM application on the card.
MCM_SIM_APP_TYPE_ISIM_V01 Identify the ISIM application on the card.

3.8.4.1.3. enum mcm_sim_file_type_t_v01

Enumerator:

MCM_SIM_FILE_TYPE_UNKNOWN_V01 Unknown file type.

MCM_SIM_FILE_TYPE_TRANSPARENT_V01 File structure consisting of a sequence of bytes.

MCM_SIM_FILE_TYPE_CYCLIC_V01 File structure consisting of a sequence of records, each containing the same fixed size in chronological order. Once all the records have been used, the oldest data is overwritten.

MCM_SIM_FILE_TYPE_LINEAR_FIXED_V01 File structure consisting of a sequence of records, each containing the same fixed size.

3.8.4.1.4. enum mcm_sim_pin_id_t_v01

Enumerator:

MCM_SIM_PIN_ID_1_V01 Level 1 user verification.

MCM_SIM_PIN_ID_2_V01 Level 2 user verification.

3.8.4.1.5. enum mcm_sim_perso_feature_t_v01

Enumerator:

MCM_SIM_PERSO_FEATURE_STATUS_UNKNOWN_V01
Unknown personalization feature.

MCM_SIM_PERSO_FEATURE_STATUS_3GPP_NETWORK_V01
Featurization based on 3GPP MCC and MNC.

MCM_SIM_PERSO_FEATURE_STATUS_3GPP_NETWORK_SUBSET_V01 Featurization based on 3GPP MCC, MNC, and IMSI digits 6 and 7.

MCM_SIM_PERSO_FEATURE_STATUS_3GPP_SERVICE_PROVIDER_V01 Featurization based on 3GPP MCC, MNC, and GID1.

MCM_SIM_PERSO_FEATURE_STATUS_3GPP_CORPORATE_V01
Featurization based on 3GPP MCC, MNC, GID1, and GID2.

MCM_SIM_PERSO_FEATURE_STATUS_3GPP_SIM_V01
Featurization based on the 3GPP IMSI.

MCM_SIM_PERSO_FEATURE_STATUS_3GPP2_NETWORK_TYPE_1_V01 Featurization based on 3GPP2 MCC and MNC.

MCM_SIM_PERSO_FEATURE_STATUS_3GPP2_NETWORK_TYPE_2_V01 Featurization based on 3GPP2 IRM code.

MCM_SIM_PERSO_FEATURE_STATUS_3GPP2_RUIM_V01
Featurization based on 3GPP2 IMSI_M.

3.8.4.1.6. enum mcm_sim_perso_operation_t_v01

Enumerator:

MCM_SIM_PERSO_OPERATION_DEACTIVATE_V01
Disable an active personalization feature.

MCM_SIM_PERSO_OPERATION_UNBLOCK_V01
Unblock a personalization feature that has been blocked.

3.8.4.1.7. enum mcm_sim_card_t_v01

Enumerator:

MCM_SIM_CARD_TYPE_UNKNOWN_V01 Unidentified card type.

MCM_SIM_CARD_TYPE_ICC_V01 Card of SIM or RUIM type.

MCM_SIM_CARD_TYPE_UICC_V01 Card of USIM or CSIM type.

3.8.4.1.8. enum mcm_sim_subscription_t_v01

Enumerator:

MCM_SIM_PROV_STATE_NONE_V01 Nonprovisioning.

MCM_SIM_PROV_STATE_PRI_V01 Primary provisioning subscription.

MCM_SIM_PROV_STATE_SEC_V01 Secondary provisioning subscription.

3.8.4.1.9. enum mcm_sim_app_state_t_v01

Enumerator:

MCM_SIM_APP_STATE_UNKNOWN_V01 Application state unknown.
MCM_SIM_APP_STATE_DETECTED_V01 Detected state.
MCM_SIM_APP_STATE_PIN1_REQ_V01 PIN1 required.
MCM_SIM_APP_STATE_PUK1_REQ_V01 PUK1 required.
MCM_SIM_APP_STATE_INITALIZATING_V01 Initializing.
MCM_SIM_APP_STATE_PERSO_CK_REQ_V01 Personalization control key required.
MCM_SIM_APP_STATE_PERSO_PUK_REQ_V01
Personalization unblock key required.

MCM_SIM_APP_STATE_PERSO_PERMANENTLY_BLOCKED_V01
Personalization is permanently blocked.
MCM_SIM_APP_STATE_PIN1_PERM_BLOCKED_V01 PIN1 is permanently blocked.
MCM_SIM_APP_STATE_ILLEGAL_V01 Illegal application state.
MCM_SIM_APP_STATE_READY_V01 Application ready state.

3.8.4.1.10. enum mcm_sim_pin_state_t_v01

Enumerator:

MCM_SIM_PIN_STATE_UNKNOWN_V01 Unknown PIN state.

MCM_SIM_PIN_STATE_ENABLED_NOT_VERIFIED_V01 PIN required, but has not been verified.

MCM_SIM_PIN_STATE_ENABLED_VERIFIED_V01 PIN required and has been verified.

MCM_SIM_PIN_STATE_DISABLED_V01 PIN not required.

MCM_SIM_PIN_STATE_BLOCKED_V01 PIN verification has failed too many times and is blocked. Recoverable through PUK verification.

MCM_SIM_PIN_STATE_PERMANENTLY_BLOCKED_V01
PUK verification has failed too many times and is not recoverable.

3.8.4.1.11. enum mcm_sim_card_state_t_v01

Enumerator:

MCM_SIM_CARD_STATE_UNKNOWN_V01 Card state unknown.
MCM_SIM_CARD_STATE_ABSENT_V01 Card is absent.
MCM_SIM_CARD_STATE_PRESENT_V01 Card is present.
MCM_SIM_CARD_STATE_ERROR_UNKNOWN_V01 Unknown error state.
MCM_SIM_CARD_STATE_ERROR_POWER_DOWN_V01 Power down.
MCM_SIM_CARD_STATE_ERROR_POLL_ERROR_V01 Poll error.
MCM_SIM_CARD_STATE_ERROR_NO_ATR_RECEIVED_V01
Failed to receive an answer to reset.
MCM_SIM_CARD_STATE_ERROR_VOLT_MISMATCH_V01 Voltage mismatch.
MCM_SIM_CARD_STATE_ERROR_PARITY_ERROR_V01 Parity error.
MCM_SIM_CARD_STATE_ERROR_SIM_TECHNICAL_PROBLEMS_V01 problems.

3.8.4.1.12. enum mcm_sim_refresh_mode_t_v01

Enumerator:

MCM_SIM_REFRESH_RESET_V01 Refresh reset.

MCM_SIM_REFRESH_NAA_INIT_V01 Refresh NAA initialization.

MCM_SIM_REFRESH_NAA_FCN_V01 Refresh NAA file change notification.

MCM_SIM_REFRESH_NAA_INIT_FCN_V01
Refresh NAA initalization and file change notification.

MCM_SIM_REFRESH_NAA_INIT_FULL_FCN_V01
Refresh NAA initalization and full file change notification.

MCM_SIM_REFRESH_NAA_APP_RESET_V01 Refresh NAA application reset.

MCM_SIM_REFRESH_3G_SESSION_RESET_V01 Refresh 3G session reset.

3.8.5. SIM Data Structures

This section contains the MCM SIM data structures.

3.8.5.1. Data Structure Documentation

3.8.5.1.1. struct mcm_sim_application_identification_info_t_v01

Data fields

Type	Parameter	Description
mcm_sim_slot-_id_t_v01	slot_id	Indicates the slot to be used. Valid values: 1 – Slot 1 2 – Slot 2
mcm_sim_app-_type_t_v01	app_t	Indicates the type of the application. Valid values: 0 – Unknown 1 – 3GPP application 2 – 3GPP2 application 3 – ISIM application Other values are reserved for the future and are to be handled as Unknown.

3.8.5.1.2. struct mcm_sim_plmn_t_v01

Data fields

Type	Parameter	Description
char	mcc	MCC value in ASCII characters.
uint32_t	mnc_len	Must be set to the number of elements in the MNC.
char	mnc	MNC value in ASCII characters.

3.8.5.1.3. struct mcm_sim_file_access_t_v01

Data fields

Type	Parameter	Description
uint16_t	offset	Offset is only required for write file access where data length is indicated.
uint8_t	record_num	Number of records involved in file access. A record number of 0 indicates transparent file access.
uint32_t	path_len	Must be set to the number of elements in the path.
char	path	File path in ASCII characters.

3.8.5.1.4. struct mcm_sim_card_result_t_v01

Data fields

Type	Parameter	Description
uint8_t	sw1	SW1 received from the card.
uint8_t	sw2	SW2 received from the card.

3.8.5.1.5. struct mcm_sim_file_info_t_v01

Data fields

Type	Parameter	Description
mcm_sim_file-_type_t_v01	file_t	File type: 0xB00 – Unknown 0xB01 – Transparent 0xB02 – Cyclic 0xB03 – Linear fixed
uint16_t	file_size	Size of transparent files.
uint16_t	record_size	Size of each cyclic or linear fixed file record.
uint16_t	record_count	Number of cyclic or linear fixed file records.

3.8.5.1.6. struct mcm_sim_depersonalization_t_v01

Data fields

Type	Parameter	Description
mcm_sim_-perso_feature_-t_v01	feature	Indicates the personalization feature to deactivate or unblock. Valid values: 0 – GW network personalization 1 – GW network subset personalization 2 – GW service provider personalization 3 – GW corporate personalization 4 – GW UIM personalization 5 – 1X network type 1 personalization 6 – 1X network type 2 personalization 7 – 1X HRPD personalization 8 – 1X service provider personalization 9 – 1X corporate personalization 10 – 1X RUIM personalization
mcm_sim-_perso_-operation_t_- v01	operation	Indicates the operation to perform. Valid values: 0 – Deactivate personalization 1 – Unblock personalization
uint32_t	ck_value_len	Must be set to the number of elements in ck_value.
char	ck_value	Control key value. This value is a sequence of ASCII characters.

3.8.5.1.7. struct mcm_sim_gw_network_subset_perso_t_v01

Data fields

Type	Parameter	Description
char	mcc	MCC value in ASCII characters.
uint32_t	mnc_len	Must be set to the number of elements in the MNC.
char	mnc	MNC value in ASCII characters.

3.8.5.1.8. struct mcm_sim_gw_sp_perso_t_v01

Data fields

Type	Parameter	Description
mcm_sim_-network_perso-_t_v01	network	MCC and MNC network information.
uint8_t	gid1	Service provider code found in GID1.

3.8.5.1.9. struct mcm_sim_gw_corporate_perso_t_v01

Data fields

Type	Parameter	Description
mcm_sim_-network_perso-_t_v01	network	MCC and MNC network information.
uint8_t	gid1	Service provider code found in GID1.
uint8_t	gid2	Corporate customer code found in GID2.

3.8.5.1.10. struct mcm_sim_sim_perso_t_v01

Data fields

Type	Parameter	Description
mcm_sim_-network_perso-_t_v01	network	MCC and MNC network information.
uint32_t	msin_len	Must be set to the number of elements in MSIN.
char	msin	MSIN value stored on the card in ASCII characters.

3.8.5.1.11. struct mcm_sim_1x_network_type2_perso_t_v01

Data fields

Type	Parameter	Description
char	irm_code	First 4 digits of the IRM-based MIN of IMSI_M in ASCII characters.

3.8.5.1.12. struct mcm_sim_perso_retries_left_t_v01

Data fields

Type	Parameter	Description
uint8_t	verify_left	Number of the remaining attempts to verify the personalization.
uint8_t	unblock_left	Number of the remaining attempts to unblock the personalization.

3.8.5.1.13. struct mcm_sim_app_info_t_v01

Data fields

Type	Parameter	Description
mcm_sim_subscription_t_v01	subscription	Type of subscription (i.e., primary, secondary, etc.).
mcm_sim_app_state_t_v01	app_state	Current state of the application.
mcm_sim_perso_feature_t_v01	perso_feature	Current personalization state and feature enabled.
uint8_t	perso_retries	Number of personalization retries.
uint8_t	perso_unblock_retries	Number of personalization unblock retries.
mcm_sim_pin_state_t_v01	pin1_state	Current PIN 1 state.
uint8_t	pin1_num_retries	Number of PIN 1 retries.
uint8_t	puk1_num_retries	Number of PUK 1 retries.
mcm_sim_pin_state_t_v01	pin2_state	Current PIN 2 state.
uint8_t	pin2_num_retries	Number of PIN 2 retries.
uint8_t	puk2_num_retries	Number of PUK 2 retries.

3.8.5.1.14. struct mcm_sim_card_app_info_t_v01

Data fields

Type	Parameter	Description
mcm_sim_app_info_t_v01	app_3gpp	Stores 3GPP application information.
mcm_sim_app_info_t_v01	app_3gpp2	Stores 3GPP2 application information.
mcm_sim_app_info_t_v01	app_isim	Stores ISIM application information.

3.8.5.1.15. struct mcm_sim_card_info_t_v01

Data fields

Type	Parameter	Description
mcm_sim_-card_state_t_- v01	card_state	Current card and card error state.
mcm_sim_-card_t_v01	card_t	Card type.
mcm_sim_-card_app_info-_t_v01	card_app_info	Stores all relevant application information.

3.8.5.1.16. struct mcm_sim_refresh_file_list_t_v01

Data fields

Type	Parameter	Description
uint32_t	path_value_len	Must be set to the number of elements in path_value.
char	path_value	Path value.

3.8.5.1.17. struct mcm_sim_refresh_event_t_v01

Data fields

Type	Parameter	Description
mcm_sim_-application_-identification_-info_t_v01	app_info	Application identification information.
mcm_sim_-refresh_mode_-t_v01	refresh_mode	Refresh mode.
uint32_t	refresh_files_-len	Must be set to the number of elements in refresh_files.
mcm_sim_-refresh_file_-list_t_v01	refresh_files	Refresh file data.

3.9. Access Terminal Command Processor

This chapter provides the messages and constants for managing the Access Terminal Command Processor (ATCoP), using MCM.

- ATCoP Message Identifiers

  ATCoP Message Structures
- ATCoP Constants

3.9.1. ATCoP Message Identifiers

This section contains the MCM ATCoP message identifiers.

- #define MCM_ATCOP_REQ_V01 0x0600
- #define MCM_ATCOP_RESP_V01 0x0600

3.9.2. ATCoP Message Structures

This section contains the MCM ATCoP message structures

3.9.2.1. Data Structure Documentation

3.9.2.1.1. struct mcm_atcop_req_msg_v01

Request message; Communicates an ATCoP message to the server.

Data fields

Type	Parameter	Description
char	cmd_req	Request message.
uint32_t	cmd_len	Request command length.

3.9.2.1.2. struct mcm_atcop_resp_msg_v01

Response message; Communicates an ATCoP message to the server.

Data fields

Type	Parameter	Description
mcm_response-_t_v01	resp	Response code.
uint8_t	cmd_resp_valid	Must be set to TRUE if cmd_resp is being passed.
char	cmd_resp	Command response.
uint8_t	resp_len_valid	Must be set to TRUE if resp_len is being passed.
uint32_t	resp_len	Response message length.

3.9.3. ATCoP Constants

This section contains the MCM ATCoP constants.

3.9.3.1. Define Documentation

- #define MCM_ATCOP_MAX_REQ_MSG_SIZE_V01 513
  Maximum request message size.
- #define MCM_ATCOP_MAX_RESP_MSG_SIZE_V01 4097
  Maximum response message size.

4. USING MCM API'S

The MCM API is a callback-oriented API for accessing and manipulating communications for the device. The main method of accessing any functionality provided by the MCM framework is to create a request message structure, fill it with relevant parameters, and then pass it to the MCM framework via a synchronous or asynchronous call, which will then return a response message corresponding to the request. In addition, indication events can be received corresponding to system messages or changes.

The following sections provide the steps for development using the IoE MCM framework.

4.1. Initialize the MCM client

The MCM client must be initialized with the following code before any other calls are sent:

```
mcm_client_handle_type     hndl; mcm_client_init (&hndl, ind_cb,
async_cb);
```

Where:

- ind_cb = Indication callback
- async_cb = Asynchronous callback

The function returns 0 if successful.

4.2. Create a Request Object with Parameters

To create a request object, use the code below and fill it with relevant parameter. For example, to create a voice call request object:

```
mcm_voice_dial_req_msg_v01req;
```

The following parameters are optional:

```
req.address_valid=1;
strlcpy(req.address,phone_number, MCM_MAX_PHONE_NUMBER_V01 + 1);
req.call_type_valid = 1;
req.call_type = MCM_VOICE_CALL_TYPE_VOICE_V01; req.uusdata_valid = 0;
```

4.3. Create a Response Object and Allocate Memory

To create a response object, use the code below and dynamically allocate memory to the object. For example, to create a voice call request object:

```
mcm_voice_dial_req_msg_v01req;
rsp = malloc(sizeof(mcm_voice_dial_resp_msg_v01)); memset(rsp, 0,
sizeof(mcm_voice_dial_resp_msg_v01));
```

Note: The release of memory allocated with malloc function at this stage is the user responsibility.

4.4. Make a Call

This API supports both synchronous and asynchronous calls.

- To dial an asynchronous voice call:

```
MCM_CLIENT_EXECUTE_COMMAND_ASYNC(hndl, MCM_VOICE_DIAL_REQ_V01,
&req,
rsp, async_cb, &token_id);
```

- To dial a synchronous call:

```
MCM_CLIENT_EXECUTE_COMMAND_SYNC(hndl, MCM_VOICE_DIAL_REQ_V01,
&req,
rsp);
```

Where:

- hndl = MCM client handle
- MCM_VOICE_DIAL_REQ_V01 = Message ID for the request to identify the different requests req = Request object
- rsp = Response object async_cb = Asynchronous callback function
- token_id = Token ID returned from the request; used to verify whether a future callback is for the same async request

4.5. Define an Asynchronous Callback Function

This function is used to receive a response from the async call (see section **Error! Reference source not found. Error! Reference source not found.**). For example, to define a callback function for dialing a voice call:

```
void async_cb(mcm_client_handle_type hndl, uint32_t msg_id,
void    *resp_c_struct, uint32_t resp_len,
void    *token_id)

{
switch(msg_id)
{
case MCM_VOICE_DIAL_RESP_V01:
rsp = (mcm_voice_dial_resp_msg_v01*)resp_c_struct; if(!rsp-
>call_id_valid)
{
printf("Invalid Valid Call ID");
}
// Can add more error checks here depending on the structure of the
// response
```

Where:

- msg_id = Message ID for the response to identify different response types resp_c_struct = Response object returned by the framework.
- token_id = Toden ID returned from the callback; this is the same value as the value that was returned from the prior async request.

4.6. Define an Indication Callback Function (Optional)

This type of callback generally provides information concerning a change of state in the system:

```
void ind_cb(mcm_client_handle_type hndl,uint32_t msg_id, void
*ind_c_struct,uint32_t ind_len);
```

To register for these types of callbacks, an event register call must be used. For example:

```
MCM_CLIENT_EXECUTE_COMMAND_SYNC(hndl, MCM_VOICE_EVENT_REGISTER_REQ_V01,
&ind_req, &ind_rsp);
```

4.7. Release a Client Handle

To release the client handle, use the following code:

```
mcm_client_release(hndl);
```

The function returns 0 if successful.

4.8. Compile the Code

To compile the code, use the steps below:

1. Obtain the header files in the API folder and the mcm_client_stubs.c in the stubs folder.
2. Create a shared library (libmcm.so) using the mcm_client_stubs.c file. For example:

```
arm-none-linux-gnueabi-gcc -I ../api -shared -Wl,-
soname,libmcm.so.0 -o libmcm.so -fPIC mcm_client_stubs.c
```

Where:

- `arm-none-linux-gnueabi-gcc =` A cross compiler
- api = A folder containing all the MCM header files

3. Link to the above shared library while generating the executable program for the C code. For example:

```
arm-none-linux-gnueabi-gcc sample_code.c -I ../api -L. -lmcm -o
sample_code
```

Where:

- sample_code.c = C code with MCM-related functions
- lmcm = Shared library libmcm.so
- sample_code = Name of the executable that was generated

5. PRODUCT AND SAFETY INFORMATION

5.1. Copyrights and Other Notices

SPECIFICATIONS ARE SUBJECT TO CHANGE WITHOUT NOTICE

Although reasonable efforts have been made to ensure the accuracy of this document, Telit assumes no liability resulting from any inaccuracies or omissions in this document, or from the use of the information contained herein. The information contained in this document has been carefully checked and is believed to be reliable. Telit reserves the right to make changes to any of the products described herein, to revise it and to make changes from time to time without any obligation to notify anyone of such revisions or changes. Telit does not assume any liability arising from the application or use of any product, software, or circuit described herein; neither does it convey license under its patent rights or the rights of others.

This document may contain references or information about Telit's products (machines and programs), or services that are not announced in your country. Such references or information do not necessarily mean that Telit intends to announce such Telit products, programming, or services in your country.

5.1.1. Copyrights

This instruction manual and the Telit products described herein may include or describe Telit copyrighted material, such as computer programs stored in semiconductor memories or other media. The laws in Italy and in other countries reserve to Telit and its licensors certain exclusive rights for copyrighted material, including the exclusive right to copy, reproduce in any form, distribute and make derivative works of the copyrighted material. Accordingly, any of Telit's or its licensors' copyrighted material contained herein or described in this instruction manual, shall not be copied, reproduced, distributed, merged or modified in any way without the express written permission of the owner. Furthermore, the purchase of Telit products shall not be deemed to grant in any way, neither directly nor by implication, or estoppel, any license.

5.1.2. Computer Software Copyrights

Telit and the Third-Party supplied Software (SW) products, described in this instruction manual may include Telit's and other Third-Party's copyrighted computer programs stored in semiconductor memories or other media. The laws in Italy and in other countries reserve to Telit and other Third-Party, SW exclusive rights for copyrighted computer programs, including – but not limited to - the exclusive right to copy or reproduce in any form the copyrighted products. Accordingly, any copyrighted computer programs contained in Telit's products described in this instruction manual shall not be copied (reverse engineered) or reproduced in any manner without the express written permission of the copyright owner, being Telit or the Third-Party software supplier. Furthermore, the purchase of Telit products shall not be deemed to grant either directly or by implication, estoppel, or in any other way, any license under the copyrights, patents or patent applications of Telit or other Third-Party supplied SW, except for the normal non-exclusive, royalty free license to use arising by operation of law in the sale of a product.

5.2. Usage and Disclosure Restrictions

5.2.1. License Agreements

The software described in this document is owned by Telit and its licensors. It is furnished by express license agreement only and shall be used exclusively in accordance with the terms of such agreement.

5.2.2. Copyrighted Materials

The Software and the documentation are copyrighted materials. Making unauthorized copies is prohibited by the law. The software or the documentation shall not be reproduced, transmitted, transcribed, even partially, nor stored in a retrieval system, nor translated into any language or computer language, in any form or by any means, without prior written permission of Telit.

5.2.3. High-Risk Materials

Components, units, or third-party goods used in the making of the product described herein are NOT fault-tolerant and are NOT designed, manufactured, or intended for use as on-line control equipment in the following hazardous environments requiring fail-safe controls: operations of Nuclear Facilities, Aircraft Navigation or Aircraft Communication Systems, Air Traffic Control, Life Support, or Weapons Systems ("High-Risk Activities"). Telit and its supplier(s) specifically disclaim any expressed or implied warranty of fitness eligibility for such High-Risk Activities.

5.2.4. Trademarks

TELIT and the Stylized T-Logo are registered in the Trademark Office. All other product or service names are property of their respective owners.

5.2.5. Third-Party Rights

The software may include Third-Party's software Rights. In this case the user agrees to comply with all terms and conditions imposed in respect of such separate software rights. In addition to Third-Party Terms, the disclaimer of warranty and limitation of liability provisions in this License, shall apply to the Third-Party Rights software as well.

TELIT HEREBY DISCLAIMS ANY AND ALL WARRANTIES EXPRESSED OR IMPLIED FROM ANY THIRD-PARTY REGARDING ANY SEPARATE FILES, ANY THIRD-PARTY MATERIALS INCLUDED IN THE SOFTWARE, ANY THIRD-PARTY MATERIALS FROM WHICH THE SOFTWARE IS DERIVED (COLLECTIVELY "OTHER CODES"), AND THE USE OF ANY OR ALL OTHER CODES IN CONNECTION WITH THE SOFTWARE, INCLUDING (WITHOUT LIMITATION) ANY WARRANTIES OF SATISFACTORY QUALITY OR FITNESS FOR A PARTICULAR PURPOSE.

NO THIRD-PARTY LICENSORS OF OTHER CODES MUST BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING WITHOUT LIMITATION LOST OF PROFITS), HOWEVER CAUSED AND WHETHER MADE UNDER CONTRACT, TORT OR OTHER LEGAL THEORY, ARISING IN ANY WAY OUT OF THE USE OR DISTRIBUTION OF THE OTHER CODES OR THE EXERCISE OF ANY RIGHTS GRANTED UNDER EITHER OR BOTH THIS LICENSE AND THE LEGAL TERMS APPLICABLE TO ANY SEPARATE FILES, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

5.2.6. Waiver of Liability

IN NO EVENT WILL TELIT AND ITS AFFILIATES BE LIABLE FOR AY DIRECT, INDIRECT, SPECIAL, GENERAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY INDIRECT DAMAGE OF ANY KIND WHATSOEVER, INCLUDING BUT NOT LIMITED TO REIMBURSEMENT OF COSTS, COMPENSATION OF ANY DAMAGE, LOSS OF PRODUCTION, LOSS OF PROFIT, LOSS OF USE, LOSS OF BUSINESS, LOSS OF DATA OR REVENUE, WHETHER OR NOT THE POSSIBILITY OF SUCH DAMAGES COULD HAVE BEEN REASONABLY FORESEEN, CONNECTD IN ANY WAY TO THE USE OF THE PRODUCT/S OR TO THE INFORMATION CONTAINED IN THE PRESENT DOCUMENTATION, EVEN IF TELIT AND/OR ITS AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES OR THEY ARE FORESEEABLE OR FOR CLAIMS BY ANY THIRD PARTY.

5.3. Safety Recommendations

Make sure the use of this product is allowed in your country and in the environment required. The use of this product may be dangerous and has to be avoided in areas where:

- it can interfere with other electronic devices, particularly in environments such as hospitals, airports, aircrafts, etc.
- there is a risk of explosion such as gasoline stations, oil refineries, etc. It is the responsibility of the user to enforce the country regulation and the specific environment regulation.

Do not disassemble the product; any mark of tampering will compromise the warranty validity. We recommend following the instructions of the hardware user guides for correct wiring of the product. The product has to be supplied with a stabilized voltage source and the wiring has to be conformed to the security and fire prevention regulations. The product has to be handled with care, avoiding any contact with the pins because electrostatic discharges may damage the product itself. Same cautions have to be taken for the SIM, checking carefully the instruction for its use. Do not insert or remove the SIM when the product is in power saving mode.

The system integrator is responsible for the functioning of the final product. Therefore, the external components of the module, as well as any project or installation issue, have to be handled with care. Any interference may cause the risk of disturbing the GSM network or external devices or having an impact on the security system. Should there be any doubt, please refer to the technical documentation and the regulations in force. Every module has to be equipped with a proper antenna with specific characteristics. The antenna has to be installed carefully in order to avoid any interference with other electronic devices and has to guarantee a minimum distance from the body (20 cm). In case this requirement cannot be satisfied, the system integrator has to assess the final product against the SAR regulation.

The equipment is intended to be installed in a restricted area location.

The equipment must be supplied by an external specific limited power source in compliance with the standard EN 62368-1.

The European Community provides some Directives for the electronic equipment introduced on the market. All of the relevant information is available on the European Community website:

https://ec.europa.eu/growth/sectors/electrical-engineering_en

6. GLOSSARY

AP	Access Point
API	Application Programming Interface
ATCOP	Access Terminal Command Processor
DM	Device Management
L2TP	Layer 2 Tunneling Protocol
MCM	Mobile Connection Manager
PPTP	Point-To-Point Tunneling Protocol
SMS	Simple Messaging Service
TLB	Translation Board
VPN	Virtual Private Network
WLAN	Wireless Local-Area Network
WWAN	Wireless Wide Area Network

7. RELATED DOCUMENTS

[1]	80624ST10996A	FN980M AT command Reference Guide
[2]	80624ST11005A	FN980M QMI Command Reference Guide_
[3]	1VV0301615	Telit EVB (Evaluation Board) User Guide

8. DOCUMENT HISTORY

Revision	Date	Changes
2	2020-06-17	Template updated Minor Editorial changes
1	2021-05-07	Initial Revision

From Mod.0818 rev.6

Connect to our site and contact our technical support team for any question

www.telit.com

Telit

Telit reserves all rights to this document and the information contained herein. Products, names, logos and designs described herein may in whole or in part be subject to intellectual property rights. The information contained herein is provided "as is". No warranty of any kind, either express or implied, is made in relation to the accuracy, reliability, fitness for a particular purpose or content of this document. This document may be revised by Telit at any time. For most recent documents, please visit www.telit.com