

Welink your smart



ME3630

WelinkOpen™ API编程指南

Version: V1.3

Date: 2018-04-29

LTE Module



Website: www.gosuncnwelink.com

E-mail: welink@gosuncn.com

REVISION HISTORY

Version	Date	Note
V1.0	2018-02-09	First version
V1.1	2018-03-28	更新 SMS 枚举类型及数据结构 新增 BSP 相关 API 函数
V1.2	2018-04-19	新增 WeFOTA 章节
V1.3	2018-04-29	新增 gsdk_device_gpio_isr 更新文档模板

CONTENTS

1	简介.....	6
1.1	范围.....	6
1.2	读者.....	6
1.3	规范.....	6
2	功能概述.....	7
3	客户端域.....	11
3.1	客户端函数.....	11
3.2	客户端消息标识符.....	13
3.3	客户端消息结构体.....	13
3.4	常见枚举.....	15
3.5	常见的数据结构.....	18
4	设备管理 (DM)	19
4.1	DM 消息标识.....	19
4.2	DM 消息结构.....	19
4.3	DM 相关常量.....	21
4.4	DM 相关枚举.....	21
5	网络注册.....	23
5.1	网络消息标识符.....	23
5.2	网络消息数据结构体.....	24
5.3	NW相关常量.....	33
5.4	NW 相关枚举.....	34
5.5	NW数据结构.....	37
6	数据业务呼叫.....	42
6.1	数据业务消息标识符.....	42
6.2	数据业务消息结构体.....	43
6.3	数据相关常量.....	52
6.4	Data 相关枚举.....	53
6.5	Data 数据结构.....	59
7	SMS.....	62
7.1	SMS消息标识符.....	62
7.2	SMS消息结构体.....	63
7.3	SMS 相关常量.....	68
7.4	SMS 相关枚举.....	68
7.5	SMS数据结构.....	70
8	移动接入点(Mobile AP)	73
8.1	Mobile AP消息标识符.....	73
8.2	Mobile AP消息结构体.....	76
8.3	Mobile AP相关常量.....	105
8.4	Mobile Ap相关枚举.....	107
8.5	Mobile AP数据结构.....	109
9	Voice.....	113

9.1	Voice 消息标识符	113
9.2	Voice消息结构体	115
9.3	Voice Constants	130
9.4	Voice相关枚举	131
9.5	Voice相关数据结构	137
10	SIM(Subscriber Identity Module)	139
10.1	SIM消息标识符.....	139
10.2	SIM消息结构体.....	140
10.3	SIM常量.....	154
10.4	SIM枚举类型.....	156
10.5	SIM数据结构.....	159
11	AT命令处理机	166
11.1	ATCoP Message Identifiers.....	166
11.2	ATCoP消息结构体	166
11.3	ATCoP 常量	167
12	定位模块	168
12.1	定位消息体标识	168
12.2	定位消息体的结构	169
12.3	定位相关常量	179
12.4	定位枚举类型	181
12.5	定位数据结构	184
13	BSP相关API函数	188
13.1	gsdk_device_set_rtc_timer	188
13.2	gsdk_device_get_rtc_timer	188
13.3	gsdk_device_cancel_rtc_timer.....	188
13.4	gsdk_device_force_sleep	189
13.5	gsdk_mcu_wakeup_module	189
13.6	gsdk_modem_wakeup_mcu	190
13.7	gsdk_device_get_usb_state	190
13.8	gsdk_device_set_usb_state	190
13.9	gsdk_device_get_adc	191
13.10	gsdk_device_conf_gpio	192
13.11	gsdk_device_read_gpio.....	192
13.12	gsdk_device_write_gpio.....	192
14	WeFOTA相关API函数及使用	194
14.1	数据类型定义	194
14.2	函数接口定义	194
14.3	示例:	195
A	MCM APIs使用说明	199
A.1	初始化MCM客户端	199
A.2	用参数初始化请求对象	199
A.3	创建响应对象并分配内存	199
A.4	打电话	200
A.5	定义异步回调函数	200
A.6	定义指示回调函数（可选）	201
A.7	释放客户端句柄	201

A.8	编译代码	201
-----	------------	-----

827455031@qq.com 47.95.190.67 2019/7/18 11:49:59
GOSUNCN Confidential

1 简介

1.1 范围

本文描述了模块产品ME3630支持的OpenLinux MCM(Mobile Connection Manager)接口，该接口使得Linux 应用程序可以直接访问高通MDM9X07平台提供的服务。

1.2 读者

该文档适用于使用MCM接口进行OpenLinux开发的软件开发人员，并需要读者对Linux编程有一定程度的了解。

1.3 规范

本文代码变量使用尖括号包围。例如: `<number>`。

函数参数传递方向如下所示：

[in]表示该参数是输入参数。

[out] 表示该参数是输出参数。

[in,out]表示该参数既可以用于输入，也可以用于输出。

2 功能概述

MCM是Linux用户面的框架服务，提供了丰富的功能接口供Linux用户程序调用，并支持多客户端并发操作。

MCM接口是基于消息机制的，客户端程序填充MCM消息结构（例如：建立数据拨号的消息类型），然后调用消息发送函数发送消息到MCM框架，该消息请求执行完成后，MCM框架发送执行结果到客户端。客户端也可以注册主动上报回调函数，MCM框架通过回调函数将上报事件传给客户端。

MCM消息类型有请求、回复和主动上报三种类型。消息结构定义在C头文件中，客户端可以通过MCM接口访问。例如：mcm_data_start_data_call_req_msg 消息建立数据连接。消息类型通常带有版本前缀便于版本控制。例如：mcm_data_start_data_call_req_msg_v01。

每个消息可以通过同步或者异步的方式发送给MCM框架，区别在于，同步操作会阻塞客户端线程，直到接收返回结果；调用异步函数发送消息会立即返回。执行结果通过回调函数返回。

客户程序发送MCM 消息前，必须先调用mcm_client_init函数初始化获取客户端句柄，结束前需调用mcm_client_release函数释放该句柄。图2-1说明客户端初始化和释放流程。

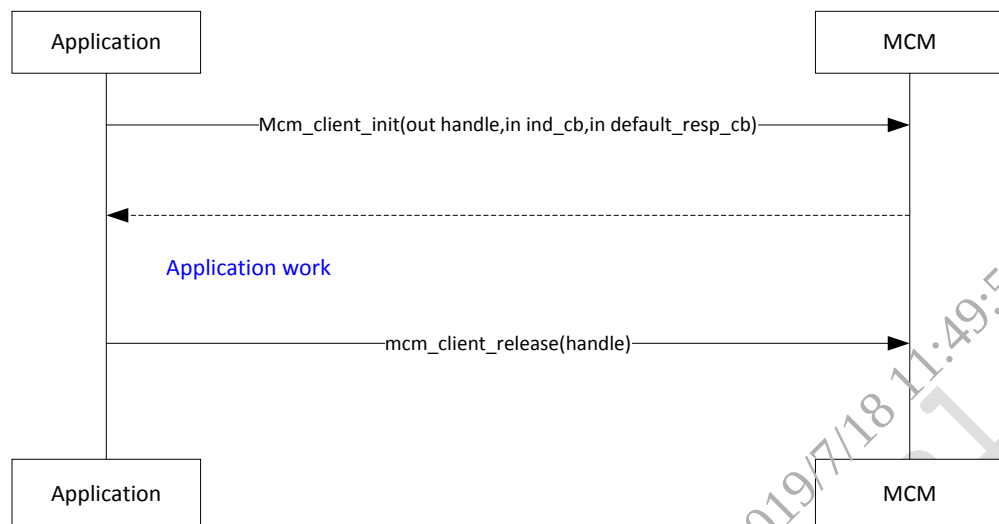


图 2-1 客户端初始化和释放流程

客户程序初始化时，需提供一组回调函数，MCM框架通过主动上报回调函数通知客户端相关信息，或者通过异步请求回调函数传递消息执行结果给客户程序。

异步请求回调函数如下所示：

```

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

主动上报回调函数如下所示：

```

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

在异步请求回调函数中，hndl句柄表示发送请求的客户程序，在主动上报回调函数中，hndl句柄表示该上报消息要发往的客户程序。

参数msg_id标识主动上报的类型或回复结果的类型。第三个参数与该类型对应。例如，当MCM框架调用语音报号请求(MCM_VOICE_DIAL_REQ_V01)的回调函数时，回调函数的msg_id被设置为MCM_VOICE_DIAL_RESP_V01，而resp_c_struct (ind_c_struct for indications)是指向mcm_voice_dial_resp_msg_v01结构的指针，该结构的长度由参数resp_len (ind_len for indications) 提供。

需要注意的是：所有的回调函数由MCM框架的接收者线程调用，所以客户程序应该注意以下两点：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:

- 客户程序应该避免长时间阻塞回调函数，否则主动上报或回复在回调函数返回以前不能发送给客户程序
- 客户程序不能在回调函数中发送任何请求，否则会导致死锁

请求消息可以通过以下三种方式发送：

```
mcm_client_execute_command_async
mcm_client_execute_command_sync
mcm_client_execute_command_sync_ex
```

mcm_client_execute_command_async 函数是异步处理函数，应用程序调用该函数发送消息时，会立即返回。该消息的执行结果通过回调函数（如果非空）或初始化时提供的默认回调函数返回。调用该函数时，应该传入请求消息结构体和返回结果结构体。消息执行结束后，MCM填充返回结果结构体，并通过回调函数传给应用程序。

mcm_client_execute_command_sync 函数是同步处理函数，该调用流程会被一直阻塞，直到请求的消息执行完成。mcm_client_execute_command_sync_ex函数增加超时参数，该函数超时也会返回，即使请求消息未处理完成。

作为简化这些函数调用的一种方式，函数结构体的每个结构指针，必须传入相应的长度，下面的宏提供了更简短的调用：

- 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

图 2-2 说明了如何使用同步和异步的方式发送MCM_VOICE_DIAL_REQ_V01 消息。

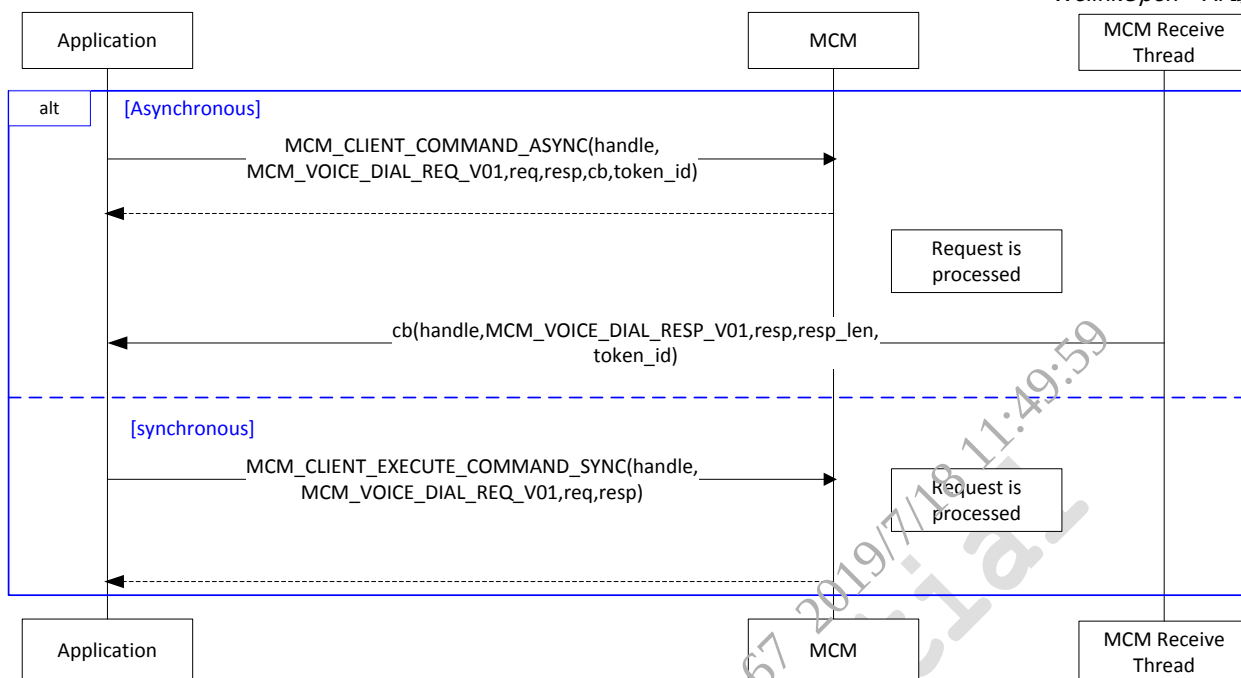


图 2-2 同步和异步语音拨号调用流程

3 客户端域

本章包含MCM面向客户端的接口和常见类型。

- [Client Functions](#)
- [Client Message Identifiers](#)
- [Client Message Structures](#)
- [Common Enumerations](#)
- [Common Data Structures](#)

3.1 客户端函数

本节包含MCM客户端函数。

3.1.1 函数文档

3.1.1.1 `uint32 mcm_client_init ( mcm_client_handle_type * hdl, mcm_client_ind_cb ind_cb, mcm_client_async_cb default_resp_cb )`

初始化MCM客户端库。

该函数在调用任何其他函数之前被调用。

参数

<i>out</i>	<i>hdl</i>	客户端句柄。
<i>in</i>	<i>ind_cb</i>	主动上报回调函数指针。
<i>in</i>	<i>default_resp_cb</i>	异步处理的默认响应回调函数指针。

返回值

MCM_SUCCESS – 0 表示成功。

MCM_ERROR_ – 查看 [3.4.1.2](#) 节错误返回类型。

3.1.1.2 `uint32 mcm_client_execute_command_async ( mcm_client_handle_type hdl, 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 )`

异步发送一个命令到MCM框架服务。

参数

<i>in</i>	<i>hndl</i>	客户端句柄。
<i>in</i>	<i>msg_id</i>	消息 ID.
<i>in</i>	<i>req_c_struct</i>	命令请求结构体指针。
<i>in</i>	<i>req_c_struct_len</i>	命令请求结构体长度。
<i>in</i>	<i>resp_c_struct</i>	命令响应结构体指针。
<i>in</i>	<i>resp_c_struct_len</i>	命令响应结构体长度。
<i>in</i>	<i>async_resp_cb</i>	异步响应回调函数指针。
<i>in</i>	<i>token_id</i>	Token ID.

返回值

MCM_SUCCESS – 0表示成功。

MCM_ERROR_ – 查看 3.4.1.2 节错误返回类型。

3.1.1.3 **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*)

同步发送一个命令到MCM框架服务。

参数

<i>in</i>	<i>hndl</i>	客户端句柄。
<i>in</i>	<i>msg_id</i>	消息 ID.
<i>in</i>	<i>req_c_struct</i>	命令请求结构体指针。
<i>in</i>	<i>req_c_struct_len</i>	命令请求结构体长度。
<i>in</i>	<i>resp_c_struct</i>	命令响应结构体指针。
<i>in</i>	<i>resp_c_struct_len</i>	命令响应结构体长度。

返回值

MCM_SUCCESS – 0表示成功。

MCM_ERROR_ – 查看 3.4.1.2 节错误返回类型。

3.1.1.4 **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)

同步发送一个命令到MCM框架服务，可指定超时参数。

参数

<i>in</i>	<i>hndl</i>	客户端句柄。
<i>in</i>	<i>msg_id</i>	消息 ID.
<i>in</i>	<i>req_c_struct</i>	命令请求结构体指针。
<i>in</i>	<i>req_c_struct_len</i>	命令请求结构体长度。
<i>in</i>	<i>resp_c_struct</i>	命令响应结构体指针。
<i>in</i>	<i>resp_c_struct_len</i>	命令响应结构体长度。
<i>in</i>	<i>timeout</i>	超时参数，单位：ms。

返回值

MCM_SUCCESS – 0表示成功。

MCM_ERROR_ – 查看 [3.4.1.2](#) 节错误返回类型。

3.1.1.5 uint32 mcm_client_release (mcm_client_handle_type hndl)

释放客户端句柄。

参数

<i>in</i>	<i>hndl</i>	客户端句柄。
-----------	-------------	--------

返回值

MCM_SUCCESS – 0表示成功。

MCM_ERROR_ – 查看 [3.4.1.2](#) 节错误返回类型。

3.2 客户端消息标识符

本小节包含MCM客户端消息标识符

```
#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
```

3.3 客户端消息结构体

本小节包含MCM客户端消息结构体。

3.3.1 数据结构体文档

3.3.1.1 struct mcm_client_require_req_msg_v01

请求消息; 所需的客户端服务

数据成员

Type	参数	描述
uint16_t	require_service	要在设备上加载的首选服务; mcm_client_service_type位掩码。

3.3.1.2 struct mcm_client_require_resp_msg_v01

响应消息; 所需的客户端服务

数据成员

Type	参数	描述
mcm_response-_t_v01	response	响应。

3.3.1.3 struct mcm_client_not_require_req_msg_v01

请求消息; 可选的客户端服务

数据域

Type	参数	描述
uint16_t	not_require_service	要在设备上加载的首选服务; mcm_client_service_type位掩码。

3.3.1.4 struct mcm_client_not_require_resp_msg_v01

响应消息; 可选的客户端服务

数据域

Type	参数	描述
mcm_response-_t_v01	response	响应。

3.4 常见枚举

本小节包含MCM常见枚举类型。

3.4.1 枚举类型文档

3.4.1.1 enum mcm_result_t_v01

成员:

MCM_RESULT_SUCCESS_V01 Success.
MCM_RESULT_FAILURE_V01 Failure.

3.4.1.2 enum mcm_error_t_v01

可能的返回值。

成员:

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_BADPARAM_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.

3.5 常见的数据结构

本小节包含MCM常见数据结构。

3.5.1 数据结构文档

3.5.1.1 struct mcm_response_t_v01

数据成员

Type	参数	描述
mcm_result_t_v01	result	结果码: • MCM_RESULT_SUCCESS • MCM_RESULT_FAILURE
mcm_error_t_v01	error	错误码。每个消息定义的错误码描述了可能的错误代码值。

4 设备管理（DM）

本章规定了MCM的DM模块数据类型，用于管理移动联网设备，比如modem，融合modem或者多modem。

- DM 消息标识
- DM 消息结构
- DM相关常量
- DM相关枚举类型

4.1 DM 消息标识

本节规定了 MCM DM相关的消息标识。

```
#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
```

4.2 DM 消息结构

本节规定了 MCM DM相关的消息标识结构。

4.2.1 数据结构集

4.2.1.1 struct mcm_dm_get_radio_mode_resp_msg_v01

结构类型：回复；

结构功能：回复网络状态。

数据域

Type	参数	描述
------	----	----

mcm_response- _t_v01	response	结果码
uint8_t	radio_mode_ valid	如果有网络状态上报，必须设置为真。
mcm_dm_ radio_mode_t_ v01	radio_mode	当前网络状态，必须是mcm_dm_radio_mode_t_v01类型中的一种。

4.2.1.2 struct mcm_dm_set_radio_mode_req_msg_v01

结构类型：请求；

结构功能：设置网络状态。

数据域

Type	参数	描述
mcm_dm_ radio_mode_t_ v01	radio_mode	要设置的网络状态

4.2.1.3 struct mcm_dm_set_radio_mode_resp_msg_v01

结构类型：回复；

结构功能：网络状态设置的结果。

数据域

Type	参数	描述
mcm_response- _t_v01	response	结果码。
uint8_t	no_change_ valid	如果有no_change的参数传递，必须设置为TRUE
uint8_t	no_change	No change.

4.2.1.4 struct mcm_dm_event_register_req_msg_v01

结构类型：请求；

结构功能：网络状态设置的结果。

数据域

Type	参数	描述
------	----	----

uint8_t	register_radio_mode_changed_event_valid	如果有register_radio_mode_changed_event 传过来，此项必须设置为TRUE
uint8_t	register_radio_mode_changed_event	网络状态变化事件。

4.2.1.5 struct mcm_dm_event_register_resp_msg_v01

结构类型：主动上报；

结构功能：网络状态设置的结果。

数据域

Type	参数	描述
mcm_response_t_v01	response	Response.

4.2.1.6 struct mcm_dm_radio_mode_changed_event_ind_msg_v01

数据域

Type	参数	描述
uint8_t	radio_mode_valid	如果有radio_mode值传递，此值设置为TRUE。
mcm_dm_radio_mode_t_v01	radio_mode	网络状态

4.3 DM 相关常量

本节规定了MCM DM功能的相关常量。

4.3.1 宏定义集

4.3.1.1 #define MCM_MAX_ARRAY_LIMIT_V01 252

数组大小上限。

4.4 DM 相关枚举

本节规定了 MCM DM 相关枚举类型。

4.4.1 枚举类型集

4.4.1.1 enum mcm_dm_radio_mode_t_v01

成员:

MCM_DM_RADIO_MODE_OFFLINE_V01 无线模块下电或状态未知.

MCM_DM_RADIO_MODE_ONLINE_V01 Radio 无线功能正常.

MCM_DM_RADIO_MODE_UNAVAILABLE_V01 无线功能不可用.

827455031@qq.com 47.95.190.67 2019/7/18 11:49:59
GOSUNCN Confidential

5 网络注册

本章包含了消息、常量、数据结构、管理和报告 移动网络(NW)的连接,状态和统计数据的枚举类型。

- [Network MessageIdentifiers](#)
- [Network Message Structures](#)
- [NW Constants](#)
- [NW Enumerations](#)
- [NW Data Structures](#)

5.1 网络消息标识符

本节包含了MCM网络消息标识符:

```
#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
```

5.2 网络消息数据结构体

本节包含了网络消息数据结构体。

5.2.1 数据结构文档

5.2.1.1 struct mcm_nw_set_config_req_msg_v01

请求消息; 定义MCM网络接口的配置项

数据域

Type	参数	描述
uint8_t	preferred_nw_mode_valid	如果要使用参数preferred_nw_mode, 该参数必须设置为TRUE
uint64_t	preferred_nw_mode	优选网络模式, 是mcm_nw_mode的位掩码
uint8_t	roaming_pref_valid	如果要使用参数roaming_pref, 该参数必须设置为TRUE
mcm_nw_roam_state_t_v01	roaming_pref	漫游偏好

5.2.1.2 struct mcm_nw_set_config_resp_msg_v01

响应消息; 定义MCM网络接口的配置项

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.

5.2.1.3 struct mcm_nw_get_config_resp_msg_v01

响应消息; 获取这个网络接口的配置状态

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
uint8_t	preferred_nw_-mode_valid	如果要使用参数preferred_nw_mode, 该参数必须设置为TRUE
uint64_t	preferred_nw_-mode	优选网络模式, 是mcm_nw_mode的位掩码
uint8_t	roaming_pref_-valid	如果要使用参数roaming_pref, 该参数必须设置为TRUE
mcm_nw_-roam_state_t_v01	roaming_pref	漫游偏好

5.2.1.4 struct mcm_nw_get_registration_status_resp_msg_v01

响应消息; 获取与连接<id>相关联的状态。

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
uint8_t	voice_-registration_-valid	如果要使用参数voice_registration, 该参数需要设置为TRUE,
mcm_nw_-common_-registration_t_v01	voice_-registration	语音业务注册
uint8_t	data_-registration_-valid	如果要使用参数 data_registration, 该参数需要设置为TRUE

mcm_nw_-common_-registration_-t_v01	data_-registration	数据业务注册
uint8_t	voice_-registration_-details_3gpp_-valid	如果要使用参数 voice_registration_details_3gpp，该参数需要设置为TRUE
mcm_nw-_3gpp_-registration_-t_v01	voice_-registration_-details_3gpp	3GPP语音业务注册信息
uint8_t	data_-registration_-details_3gpp_-valid	如果要使用参数 data_registration_details_3gpp，该参数需要设置为TRUE
mcm_nw-_3gpp_-registration_-t_v01	data_-registration_-details_3gpp	3GPP数据业务注册信息
uint8_t	voice_-registration_-details_3gpp2_-valid	如果要使用参数 voice_registration_details_3gpp2，该参数需要设置为TRUE
mcm_nw-_3gpp2_-registration_-t_v01	voice_-registration_-details_3gpp2	3GPP2语音业务注册信息
uint8_t	data_-registration_-details_3gpp2_-valid	如果要使用参数 data_registration_details_3gpp2，该参数需要设置为TRUE
mcm_nw-_3gpp2_-registration_-t_v01	data_-registration_-details_3gpp2	3GPP2数据业务注册信息

5.2.1.5 struct mcm_nw_scan_resp_msg_v01

响应消息：获取与连接< id >相关联的状态

数据域

Type	参数	描述
mcm_response_-t_v01	response	结果码.
uint8_t	entry_valid	如果要使用参数entry，该参数需要设置为TRUE

uint32_t	entry_len	必须设置为参数entry的数据长度
mcm_nw_scan_entry_t_v01	entry	扫描入口

5.2.1.6 struct mcm_nw_get_operator_name_resp_msg_v01

响应消息; 获取与连接< id >相关联的运营商名称

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.
uint8_t	operator_name_valid	如果要使用参数operator_name, 该参数需要设置为TRUE
mcm_nw_operator_name_t_v01	operator_name	运营商名称。

5.2.1.7 struct mcm_nw_screen_on_off_req_msg_v01

请求消息; 关闭/打开屏幕, 以节省电量

数据域

Type	参数	描述
uint8_t	turn_off_screen	关闭屏幕。

5.2.1.8 struct mcm_nw_screen_on_off_resp_msg_v01

响应消息; 关闭/打开屏幕, 以节省电量

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.

5.2.1.9 struct mcm_nw_selection_req_msg_v01

请求消息; 网络选择 (手动或自动)

数据域

Type	参数	描述
mcm_nw_selection_t_v01	nw_selection_info	网络选择消息

5.2.1.10 struct mcm_nw_selection_resp_msg_v01

响应消息; 网络选择（手动或自动）

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.

5.2.1.11 struct mcm_nw_get_signal_strength_resp_msg_v01

响应消息; 获取信号强度信息

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.
uint8_t	gsm_sig_info_valid	如果要使用参数 gsm_sig_info, 该参数需要设置为TRUE
mcm_nw_gsm_sig_info_t_v01	gsm_sig_info	GSM信号
uint8_t	wcdma_sig_info_valid	如果要使用参数 wcdma_sig_info, 该参数需要设置为TRUE
mcm_nw_wcdma_sig_info_t_v01	wcdma_sig_info	WCDMA信号
uint8_t	tdscdma_sig_info_valid	如果要使用参数 tdscdma_sig_info, 该参数需要设置为TRUE
mcm_nw_tdscdma_sig_info_t_v01	tdscdma_sig_info	TDSCDMA 信号
uint8_t	lte_sig_info_valid	如果要使用参数 lte_sig_info, 该参数需要设置为TRUE
mcm_nw_lte_sig_info_t_v01	lte_sig_info	LTE 信号
uint8_t	cdma_sig_info_valid	如果要使用参数 cdma_sig_info, 该参数需要设置为TRUE

mcm_nw_- cdma_sig_info- _t_v01	cdma_sig_info	CDMA 信号
uint8_t	hdr_sig_info_- valid	如果要使用参数 hdr_sig_info，该参数需要设置为TRUE
mcm_nw_hdr_- sig_info_t_v01	hdr_sig_info	HDR信号

5.2.1.12 struct mcm_nw_get_cell_access_state_resp_msg_v01

响应消息; 获取小区接入状态

数据域

Type	参数	描述
mcm_response- _t_v01	response	结果码.
uint8_t	nw_cell_- access_state_- valid	如果要使用参数nw_cell_access_state，该参数需要设置为TRUE
mcm_nw_cell- _access_state_- t_v01	nw_cell_- access_state	网络小区接入状态

5.2.1.13 struct mcm_nw_get_nitz_time_info_resp_msg_v01

响应消息; 获取NITZ时间信息

数据域

Type	参数	描述
mcm_response- _t_v01	response	结果码.
uint8_t	nw_nitz_time_- valid	如果要使用参数nw_nitz_time，该参数需要设置为TRUE
mcm_nw_nitz- _time_t_v01	nw_nitz_time	网络NITZ时间

5.2.1.14 struct mcm_nw_event_register_req_msg_v01

请求消息; 事件注册

数据域

Type	参数	描述
------	----	----

uint8_t	register_voice_registration_event_valid	如果要使用参数 register_voice_registration_event，该参数需要设置为TRUE
uint8_t	register_voice_registration_event	注册语音业务注册事件
uint8_t	register_data_registration_event_valid	如果要使用参数 register_data_registration_event，该参数需要设置为TRUE
uint8_t	register_data_registration_event	注册数据业务注册事件
uint8_t	register_signal_strength_event_valid	如果要使用参数 register_signal_strength_event，该参数需要设置为TRUE
uint8_t	register_signal_strength_event	注册信号强度事件
uint8_t	register_cell_access_state_change_event_valid	如果要使用参数 register_cell_access_state_change_event，该参数需要设置为TRUE
uint8_t	register_cell_access_state_change_event	注册接入小区状态变化事件
uint8_t	register_nitz_time_update_event_valid	如果要使用参数 register_nitz_time_update_event，该参数需要设置为TRUE
uint8_t	register_nitz_time_update_event	注册NITZ时间更新事件

5.2.1.15 struct mcm_nw_event_register_resp_msg_v01

响应消息; 事件注册

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.

5.2.1.16 struct mcm_nw_voice_registration_event_ind_msg_v01

指示消息，用来指示MCM_NW_VOICE_REGISTRATION_EV对应的注册事件。

数据域

Type	参数	描述
uint8_t	voice_ - registration_ - valid	如果要使用参数 voice_registration，该参数需要设置为TRUE
mcm_nw_ - common_ - registration_ - t_v01	voice_ - registration	语音业务注册
uint8_t	voice_ - registration_ - details_3gpp_ - valid	如果要使用参数voice_registration_details_3gpp，该参数需要设置为TRUE
mcm_nw_ - _3gpp_ - registration_ - t_v01	voice_ - registration_ - details_3gpp	3GPP语音业务注册
uint8_t	voice_ - registration_ - details_3gpp2_ - valid	如果要使用参数voice_registration_details_3gpp2，该参数需要设置为TRUE
mcm_nw_ - _3gpp2_ - registration_ - t_v01	voice_ - registration_ - details_3gpp2	3GPP2语音业务注册

5.2.1.17 struct mcm_nw_data_registration_event_ind_msg_v01

指示消息，用来指示MCM_NW_DATA_REGISTRATION_EV对应的注册事件。

数据域

Type	参数	描述
uint8_t	data_ - registration_ - valid	如果要使用参数data_registration，该参数需要设置为TRUE
mcm_nw_ - common_ - registration_ - t_v01	data_ - registration	数据业务注册

uint8_t	data_registration_details_3gpp_valid	如果要使用参数data_registration_details_3gpp，该参数需要设置为TRUE
mcm_nw_3gpp_registration_t_v01	data_registration_details_3gpp	3GPP数据业务注册
uint8_t	data_registration_details_3gpp2_valid	如果要使用参数data_registration_details_3gpp2，该参数需要设置为TRUE
mcm_nw_3gpp2_registration_t_v01	data_registration_details_3gpp2	3GPP2数据业务注册

5.2.1.18 struct mcm_nw_signal_strength_event_ind_msg_v01

指示消息; 用来指示MCM_NW_SIGNAL_STRENGTH_EV对应的注册事件。

数据域

Type	参数	描述
uint8_t	gsm_sig_info_valid	如果要使用参数gsm_sig_info，该参数需要设置为TRUE
mcm_nw_gsm_sig_info_t_v01	gsm_sig_info	GSM信号
uint8_t	wcdma_sig_info_valid	如果要使用参数wcdma_sig_info，该参数需要设置为TRUE
mcm_nw_wcdma_sig_info_t_v01	wcdma_sig_info	WCDMA信号
uint8_t	tdscdma_sig_info_valid	如果要使用参数tdscdma_sig_info，该参数需要设置为TRUE
mcm_nw_tdscdma_sig_info_t_v01	tdscdma_sig_info	TDSCDMA 信号
uint8_t	lte_sig_info_valid	如果要使用参数lte_sig_info，该参数需要设置为TRUE
mcm_nw_lte_sig_info_t_v01	lte_sig_info	LTE信号
uint8_t	cdma_sig_info_valid	如果要使用参数cdma_sig_info，该参数需要设置为TRUE

mcm_nw_- cdma_sig_info- _t_v01	cdma_sig_info	CDMA信号
uint8_t	hdr_sig_info_- valid	如果要使用参数hdr_sig_info，该参数需要设置为TRUE
mcm_nw_hdr_- sig_info_t_v01	hdr_sig_info	HDR信号

5.2.1.19 struct mcm_nw_cell_access_state_change_event_ind_msg_v01

指示消息; 指示小区接入状态变更，如仅紧急模式，CS call only

数据域

Type	参数	描述
mcm_nw_cell- _access_state- _t_v01	nw_cell_- access_state	小区接入状态

5.2.1.20 struct mcm_nw_nitz_time_ind_msg_v01

指示消息; 指示NITZ时间更新

数据域

Type	参数	描述
mcm_nw_nitz- _time_t_v01	nw_nitz_time	NITZ时间更新

5.3 NW相关常量

本节包含MCM 网络注册常量。

网络选择模式(从 5.3.1.2节至5.3.1.9)用来指定首选的用于搜索和注册的网络。例如，MCM_NW_MODE_GSM| MCM_NW_MODE_WCDMA 选择 GSM/WCDMA 网络,且WCDMA作为优选值。以上设置与MCM_NW_MODE_GSM| MCM_NW_MODE_WCDMA|MCM_NW_MODE_PRL一致，只是该设置以漫游列表作为优选值。

5.3.1 宏定义集

5.3.1.1 #define MCM_NW_SCAN_LIST_MAX_V01 40

网络扫描列表的最大条目数

5.3.1.2 #define MCM_MODE_NONE_V01 ((mcm_nw_mode_type_v01)0x00ull)

无网络

5.3.1.3 #define MCM_MODE_GSM_V01 ((mcm_nw_mode_type_v01)0x01ull)

包含GSM网络

5.3.1.4 #define MCM_MODE_WCDMA_V01 ((mcm_nw_mode_type_v01)0x02ull)

包含WCDMA网络

5.3.1.5 #define MCM_MODE_CDMA_V01 ((mcm_nw_mode_type_v01)0x04ull)

包含CDMA网络

5.3.1.6 #define MCM_MODE_EVDO_V01 ((mcm_nw_mode_type_v01)0x08ull)

包含EVDO网络

5.3.1.7 #define MCM_MODE_LTE_V01 ((mcm_nw_mode_type_v01)0x10ull)

包含LTE网络

5.3.1.8 #define MCM_MODE_TDSCDMA_V01 ((mcm_nw_mode_type_v01)0x20ull)

包含TDSCDMA网络

5.3.1.9 #define MCM_MODE_PRL_V01 ((mcm_nw_mode_type_v01)0x10000ull)

根据优选漫游列表确定优先级

5.4 NW 相关枚举

该节包含MCM网络注册的相关枚举类型。

5.4.1 枚举类型集

5.4.1.1 enum mcm_nw_service_t_v01

成员:

MCM_NW_SERVICE_NONE_V01 未注册或者无数据

MCM_NW_SERVICE_LIMITED_V01 已注册，仅用于紧急业务

MCM_NW_SERVICE_FULL_V01 已注册，所有业务可用

5.4.1.2 enum mcm_nw_selection_type_t_v01

成员:

MCM_NW_SELECTION_AUTOMATIC_V01 自动选网

MCM_NW_SELECTION_MANUAL_V01 手动选网

5.4.1.3 enum mcm_nw_network_status_t_v01

成员:

MCM_NW_NETWORK_STATUS_NONE_V01 网络状态不可用

MCM_NW_NETWORK_STATUS_CURRENT_SERVING_V01 当前服务网络

MCM_NW_NETWORK_STATUS_PREFERRED_V01 优选网络

MCM_NW_NETWORK_STATUS_NOT_PREFERRED_V01 非优选网络

MCM_NW_NETWORK_STATUS_AVAILABLE_V01 服务可用

MCM_NW_NETWORK_STATUS_FORBIDDEN_V01 服务不可用

5.4.1.4 enum mcm_nw_radio_tech_t_v01

成员:

MCM_NW_RADIO_TECH_GSM_V01 GSM; 仅支持语音业务

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.

5.4.1.5 enum mcm_nw_cell_access_state_t_v01

成员:

MCM_NW_CELL_ACCESS_NONE_V01 未知小区接入状态
MCM_NW_CELL_ACCESS_NORMAL_ONLY_V01 小区准入，仅用于普通的呼叫
MCM_NW_CELL_ACCESS_EMERGENCY_ONLY_V01 小区准入，仅用于紧急呼叫
MCM_NW_CELL_ACCESS_NO_CALLS_V01 小区准入，不能用于任何类型的呼叫
MCM_NW_CELL_ACCESS_ALL_CALLS_V01 小区准入，可用于任何类型的呼叫

5.4.1.6 enum mcm_nw_roam_state_t_v01

成员:

MCM_NW_ROAM_STATE_OFF_V01 无漫游，或漫游指示关闭
MCM_NW_ROAM_STATE_ON_V01 漫游指示打开

5.4.1.7 enum mcm_nw_tech_domain_t_v01

成员:

MCM_NW_TECH_DOMAIN_NONE_V01 None.
MCM_NW_TECH_DOMAIN_3GPP_V01 3GPP.
MCM_NW_TECH_DOMAIN_3GPP2_V01 3GPP2.

5.4.1.8 enum mcm_nw_deny_reason_t_v01

成员:

MCM_NW_IMSI_UNKNOWN_HLR_DENY_REASON_V01 HLR中未知IMSI.
MCM_NW_ILLEGAL_MS_DENY_REASON_V01 非法 MS.
MCM_NW_IMSI_UNKNOWN_VLR_DENY_REASON_V01 VLR中未知IMSI.
MCM_NW_IMEI_NOT_ACCEPTED_DENY_REASON_V01 IMEI不接受.
MCM_NW_ILLEGAL_ME_DENY_REASON_V01 非法ME.
MCM_NW_PLMN_NOT_ALLOWED_DENY_REASON_V01 PLMN不允许
MCM_NW_LA_NOT_ALLOWED_DENY_REASON_V01 位置区域不允许
MCM_NW_ROAMING_NOT_ALLOWED_LA_DENY_REASON_V01 该区域不允许漫游
MCM_NW_NO_SUITABLE_CELLS_LA_DENY_REASON_V01 在位置区域内无合适小区
MCM_NW_NETWORK_FAILURE_DENY_REASON_V01 网络失败
MCM_NW_MAC_FAILURE_DENY_REASON_V01 MAC 失败.
MCM_NW_SYNCH_FAILURE_DENY_REASON_V01 Sync 失败.
MCM_NW_CONGESTION_DENY_REASON_V01 拥塞
MCM_NW_GSM_AUTHENTICATION_UNACCEPTABLE_DENY_REASON_V01 不接受GSM认证
MCM_NW_NOT_AUTHORIZED_CSG_DENY_REASON_V01 该CGS未授权
MCM_NW_SERVICE_OPTION_NOT_SUPPORTED_DENY_REASON_V01 不支持服务选项
MCM_NW_REQ_SERVICE_OPTION_NOT_SUBSCRIBED_DENY_REASON_V01 没有订
 阅请求的服务选项
MCM_NW_CALL_CANNOT_BE_IDENTIFIED_DENY_REASON_V01 Call cannot be identified. 呼叫
 不能被识别

MCM_NW_SEMANTICALLY_INCORRECT_MSG_DENY_REASON_V01 语义上错误的信息

MCM_NW_INVALID_MANDATORY_INFO_DENY_REASON_V01 无效的必备信息

MCM_NW_MSG_TYPE_NON_EXISTENT_DENY_REASON_V01 消息类型不存在或者没有实现。

MCM_NW_INFO_ELEMENT_NON_EXISTENT_DENY_REASON_V01 消息类型与协议状态不匹配

MCM_NW_CONDITIONAL_IE_ERR_DENY_REASON_V01 条件IE错误

MCM_NW_MSG_INCOMPATIBLE_PROTOCOL_STATE_DENY_REASON_V01 消息与协议状态不兼容

MCM_NW_PROTOCOL_ERROR_DENY_REASON_V01 未指定的协议错误

5.5 NW数据结构

该节包含MCM网络注册数据结构。

5.5.1 数据结构文档

5.5.1.1 struct mcm_nw_operator_name_t_v01

数据域

Type	参数	描述
char	long_eons	Long EONS.
char	short_eons	Short EONS.
char	mcc	Mobile country code.
char	mnc	Mobile network code.

5.5.1.2 struct mcm_nw_scan_entry_t_v01

数据域

Type	参数	描述
mcm_nw_operator_name_t_v01	operator_name	运营商名字
mcm_nw_network_status_t_v01	network_status	网络状态
mcm_nw_radio_tech_t_v01	rat	无线电技术

5.5.1.3 struct mcm_nw_common_registration_t_v01

数据域

Type	参数	描述
mcm_nw_tech_domain_t_v01	tech_domain	技术，用于决定mcm_tech结构体类型： 0 – None, 1 – 3GPP, 2 – 3GPP2.
mcm_nw_radio_tech_t_v01	radio_tech	无线电技术, 参见mcm_nw_radio_tech_t_v01
mcm_nw_roam_state_t_v01	roaming	0 – Off 1 – Roaming (3GPP2 有扩展值).
mcm_nw_deny_reason_t_v01	deny_reason	当注册状态是mcm_nw_deny_reason_t_v01时，设置该值
mcm_nw_service_t_v01	registration_state	注册状态

5.5.1.4 struct mcm_nw_3gpp_registration_t_v01
数据域

Type	参数	描述
mcm_nw_tech_domain_t_v01	tech_domain	技术，用于决定mcm_tech结构体类型： 0 – None, 1 – 3GPP, 2 – 3GPP2.
mcm_nw_radio_tech_t_v01	radio_tech	无线电技术, 参见mcm_nw_radio_tech_t_v01.
char	mcc	Mobile country code(MCC)
char	mnc	Mobile network code(MNC)
mcm_nw_roam_state_t_v01	roaming	0 – Off, 1 – Roaming (3GPP2 有扩展值).
uint8_t	forbidden	禁止: 0 – No, 1 – Yes.
uint32_t	cid	注册的3GPP系统的cell ID(cid)
uint16_t	lac	注册的3GPP系统的Location area code(lac)
uint16_t	psc	Primary scrambling code (WCDMA only); 0 – None.
uint16_t	tac	LTE的跟踪区号 (tac) 信息

5.5.1.5 struct mcm_nw_3gpp2_registration_t_v01
数据域

Type	参数	描述
------	----	----

mcm_nw_tech_domain_t_v01	tech_domain	技术，用于决定mcm_tech结构体类型： 0 – None, 1 – 3GPP, 2 – 3GPP2.
mcm_nw_radio_tech_t_v01	radio_tech	无线电技术, 参见mcm_nw_radio_tech_t_v01.
char	mcc	Mobile country code(MCC)
char	mnc	Mobile network code(MNC)
mcm_nw_roam_state_t_v01	roaming	漫游状态，参见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	并发服务支持: 0 – No, 1 – Yes.
uint16_t	sid	CDMA system ID.
uint16_t	nid	CDMA network ID.
uint16_t	bsid	Base station ID.

5.5.1.6 struct mcm_nw_selection_t_v01

数据域

Type	参数	描述
mcm_nw_selection_type_t_v01	nw_selection_type	网络选择类型
char	mcc	手动选网的MCC
char	mnc	手动选网的MNC
mcm_nw_radio_tech_t_v01	rat	无线电技术

5.5.1.7 struct mcm_nw_gsm_sig_info_t_v01

数据域

Type	参数	描述
int8_t	rss	RSSI 单位为dBm. 表示接收信号强度。信号值，当≤-125时，表示没有信号

5.5.1.8 struct mcm_nw_wcdma_sig_info_t_v01

数据域

Type	参数	描述
int8_t	rss	RSSI单位为dBm. 表示前向通道pilot Ec, 信号相关参数, 当≤-125时, 表示没有信号
int16_t	ecio	Ec/Io值代表负0.5 dB的增量, 如2表示-1 dbm.

5.5.1.9 struct mcm_nw_tdscdma_sig_info_t_v01

数据域

Type	参数	描述
int8_t	rss	RSSI单位为dBm. 表示前向通道pilot Ec, 信号相关参数, 当≤-125时, 表示没有信号
int8_t	rscp	RSCP单位为dBm.
int16_t	ecio	Ec/Io值代表负0.5 dB的增量, 如2表示-1 dbm.
int8_t	sinr	测量值SINR单位为dB

5.5.1.10 struct mcm_nw_lte_sig_info_t_v01

数据域

Type	参数	描述
int8_t	rss	RSSI单位为dBm. 表示前向通道pilot Ec, 信号相关参数, 当≤-125时, 表示没有信号
int8_t	rsrq	RSRQ单位为dB (整型), 如L1的测量值. 范围: -3 to -20 (-3 equals -3 dB, -20 equals -20 dB).
int8_t	rsrp	当前 RSRP 单位为dBm, , 如L1的测量值. 范围: -44 to -140 (-44 equals -44 dBm, -140 equals -140 dBm).
int8_t	snr	SNR值作为整型数, 单位为0.1dB, 如-160表示-16 dB, 246表示值 24.6dB

5.5.1.11 struct mcm_nw_cdma_sig_info_t_v01

数据域

Type	参数	描述
int8_t	rss	RSSI单位为dBm. 表示前向通道pilot Power (AGC) + Ec/Io. 信号相关参数, 当≤-125时, 表示没有信号
int16_t	ecio	Ec/Io值代表负0.5 dB的增量, 如2表示-1 dbm.

5.5.1.12 struct mcm_nw_hdr_sig_info_t_v01

数据域

Type	参数	描述
int8_t	rssi	RSSI单位为dBm. 表示前向通道pilot Power (AGC) + Ec/Io. 信号相关参数, 当 ≤ -125 时, 表示没有信号
int16_t	ecio	Ec/Io值代表负0.5 dB的增量, 如2表示-1 dbm.
int8_t	sinr	SINR
int32_t	io	接收的 IO, 单位为dBm.

5.5.1.13 struct mcm_nw_nitz_time_t_v01

数据域

Type	参数	描述
char	nitz_time	NITZ 时间

6 数据业务呼叫

该章主要描述MCM数据业务调用

- [Data Message Identifiers](#)
- [Data Message Structures](#)
- [Data Constants](#)
- [Data Enumerations](#)
- [Data Structures](#)

6.1 数据业务消息标识符

该节包含MCM数据业务消息标识符

```
#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
```

6.2 数据业务消息结构体

该节包含MCM数据业务消息相关结构体

6.2.1 数据结构文档

6.2.1.1 struct mcm_data_start_data_call_req_msg_v01

请求消息;发送请求，用于建立一个指定call ID的数据业务呼叫连接。

数据域

Type	参数	描述
uint8_t	ip_family_valid	如需使用ip_family，该参数需要设置为TRUE
int8_t	ip_family	IP family 要求: • 4 – IPv4 • 6 – IPv6 • 10 – IPv4/v6
uint8_t	apn_name_valid	如需使用apn_name，该参数需要设置为TRUE
char	apn_name	所需的APN，一个字符串，表示运营商定义的PDN
uint8_t	user_name_valid	如需使用user_name，该参数需要设置为TRUE
char	user_name	APN的用户名
uint8_t	password_valid	如需使用password，该参数需要设置为TRUE
char	password	APN的密码.
uint8_t	tech_pref_valid	如需使用tech_pref，该参数需要设置为TRUE
int8_t	tech_pref	Technology preference: • 0 – CDMA • 1 – UMTS

uint8_t	umts_profile_valid	如需使用umts_profile, 该参数需要设置为TRUE
int8_t	umts_profile	UMTS/LTE profile ID.
uint8_t	cdma_profile_valid	如需使用CDMA_profile, 该参数需要设置为TRUE
int8_t	cdma_profile	CDMA profile ID.

6.2.1.2 struct mcm_data_start_data_call_rsp_msg_v01

响应消息;发送用于建立一个指定call ID的数据业务呼叫连接请求

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.
mcm_data_call_status_t_v01	call_status	MCM数据呼叫状态
uint8_t	call_id_valid	如需使用call_id, 该参数需要设置为TRUE
int32_t	call_id	成功建立的数据呼叫Call ID
uint8_t	vce_reason_valid	如需使用vce_reason, 该参数需要设置为TRUE
mcm_data_verbose_call_end_reason_t_v01	vce_reason	长字符串格式的呼叫结束原因

6.2.1.3 struct mcm_data_stop_data_call_req_msg_v01

请求消息;发送一个用于终止指定call ID的数据业务呼叫连接请求

数据域

Type	参数	描述
int32_t	call_id	需要终止的数据呼叫call ID

6.2.1.4 struct mcm_data_stop_data_call_rsp_msg_v01

响应消息;发送一个用于终止指定call ID的数据业务呼叫连接请求

数据域

Type	参数	描述
mcm_response-_t_v01	resp	响应

6.2.1.5 struct mcm_data_get_pkt_stats_req_msg_v01

请求消息;获取指定call id连接的数据包统计

数据域

Type	参数	描述
int32_t	call_id	需要获取数据的call id

6.2.1.6 struct mcm_data_get_pkt_stats_rsp_msg_v01

响应消息; 获取指定call id连接的数据包统计

数据域

Type	参数	描述
mcm_response-_t_v01	resp	响应
uint8_t	pkt_stats_valid	如需使用pkt_stats, 该参数需要设置为TRUE
mcm_data_pkt-_stats_t_v01	pkt_stats	数据包统计

6.2.1.7 struct mcm_data_reset_pkt_stats_req_msg_v01

请求消息;复位指定call id连接的数据包统计数据

数据域

Type	参数	描述
int32_t	call_id	需要复位统计数据的数据包统计的call id

6.2.1.8 struct mcm_data_reset_pkt_stats_rsp_msg_v01

响应消息;复位指定call id连接的数据包统计数据

数据域

Type	参数	描述
mcm_response-t_v01	resp	响应结果码.

6.2.1.9 struct mcm_data_get_device_name_req_msg_v01

请求消息;获取指定call id连接的路由信息

数据域

Type	参数	描述
int32_t	call_id	需要获取连接的路由信息的call id

6.2.1.10 struct mcm_data_get_device_name_rsp_msg_v01

响应消息;获取指定call id连接的路由信息

数据域

Type	参数	描述
mcm_response-t_v01	resp	响应
uint8_t	device_name_valid	如需使用device_name, 该参数需要设置为TRUE
uint32_t	device_name_len	必须设置为device_name参数的长度
char	device_name	设备名称

6.2.1.11 struct mcm_data_get_device_addr_count_req_msg_v01

请求消息; 获取到当前call ID连接的可用的IP地址数量

数据域

Type	参数	描述
int32_t	call_id	需要获取可用IP地址数量的call ID连接

6.2.1.12 struct mcm_data_get_device_addr_count_rsp_msg_v01

响应消息;获取到当前call ID连接的可用的IP地址数量

数据域

Type	参数	描述
mcm_response-t_v01	resp	响应
uint8_t	addr_count-valid	如需使用addr_count, 该参数需要设置为TRUE
int8_t	addr_count	可用IP地址数量

6.2.1.13 struct mcm_data_get_call_tech_req_msg_v01

请求消息; 获取到当前call ID连接可用的底层技术

数据域

Type	参数	描述
int32_t	call_id	需获取相关信息的指定call id

6.2.1.14 struct mcm_data_get_call_tech_rsp_msg_v01

响应消息;获取到当前call ID连接可用的底层技术

数据域

Type	参数	描述
mcm_response-t_v01	resp	响应
uint8_t	call_tech_valid	如需使用call_tech, 该参数需要设置为TRUE
mcm_data_bearer_tech_info_t_v01	call_tech	数据呼叫承载技术

6.2.1.15 struct mcm_data_get_call_status_req_msg_v01

请求消息; 获取call ID连接的当前状态

数据域

Type	参数	描述
int32_t	call_id	需获取相关信息的指定call id

6.2.1.16 struct mcm_data_get_call_status_rsp_msg_v01

响应消息; 获取call ID连接的当前状态

数据域

Type	参数	描述
mcm_response_t_v01	resp	响应
uint8_t	call_status_valid	如需使用call_tech, 该参数需要设置为TRUE
mcm_data_call_status_t_v01	call_status	数据业务状态

6.2.1.17 struct mcm_data_get_reg_status_rsp_msg_v01

响应消息; 获取当前数据注册状态

数据域

Type	参数	描述
mcm_response_t_v01	resp	响应
uint8_t	reg_status_valid	如需使用reg_status, 该参数需要设置为TRUE
mcm_data_reg_status_t_v01	reg_status	数据调制解调器注册状态。

6.2.1.18 struct mcm_data_get_device_addr_req_msg_v01

请求消息; 获取call ID连接的IP地址列表

数据域

Type	参数	描述
int32_t	call_id	需获取相关信息的指定call id

6.2.1.19 struct mcm_data_get_device_addr_rsp_msg_v01

响应消息;获取call ID连接的IP地址列表

数据域

Type	参数	描述
mcm_response-_t_v01	resp	响应
uint8_t	addr_info_valid	如需使用addr_info, 该参数需要设置为TRUE
uint32_t	addr_info_len	必须设置为addr_info里地址的个数
mcm_data-_addr_t_info_-v01	addr_info	数据设备地址

6.2.1.20 struct mcm_data_get_channel_rate_req_msg_v01

请求消息; 获取call ID连接的当前的最大信道传输速率

数据域

Type	参数	描述
int32_t	call_id	需获取相关信息的指定call id

6.2.1.21 struct mcm_data_get_channel_rate_rsp_msg_v01

响应消息;获取call ID连接的当前的最大信道传输速率

数据域

Type	参数	描述
mcm_response-_t_v01	resp	响应
uint8_t	channel_rate-_valid	如需使用channel_rate, 该参数需要设置为TRUE
mcm_data-_channel_rate_t-_v01	channel_rate	信道传输速率

6.2.1.22 struct mcm_data_event_register_req_msg_v01

请求消息; 事件注册

数据域

Type	参数	描述
uint8_t	register_net_up_event_valid	如需使用register_net_up_event，该参数需要设置为TRUE
uint8_t	register_net_up_event	注册网络组建事件
uint8_t	register_net_down_event_valid	如需使用register_net_down_event，该参数需要设置为TRUE
uint8_t	register_net_down_event	注册网络关断事件
uint8_t	register_net_new_addr_event_valid	如需使用register_net_new_addr_event，该参数需要设置为TRUE
uint8_t	register_net_new_addr_event	注册新地址事件
uint8_t	register_net_del_addr_event_valid	如需使用register_net_del_addr_event，该参数需要设置为TRUE
uint8_t	register_net_del_addr_event	注册地址删除事件
uint8_t	register_reg_srv_status_event_valid	如需使用register_reg_srv_status_event，该参数需要设置为TRUE
uint8_t	register_reg_srv_status_event	注册服务状态事件
uint8_t	register_bearer_tech_status_event_valid	如需使用register_bearer_tech_status_event，该参数需要设置为TRUE
uint8_t	register_bearer_tech_status_event	册承载技术状态事件
uint8_t	register_dormancy_status_event_valid	如需使用register_dormancy_status_event，该参数需要设置为TRUE
uint8_t	register_dormancy_status_event	注册休眠状态事件

6.2.1.23 struct mcm_data_event_register_resp_msg_v01

响应消息; 事件注册

数据域

Type	参数	描述
mcm_response-_t_v01	response	响应

6.2.1.24 struct mcm_data_unsol_event_ind_msg_v01

指示消息;指示相应的主动上报事件

数据域

Type	参数	描述
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	如需使用call_id, 该参数需要设置为TRUE
int32_t	call_id	成功建立的数据连接的call id
uint8_t	call_status_valid	如需使用call_status, 该参数需要设置为TRUE
mcm_data_call_status_t_v01	call_status	数据呼叫状态
uint8_t	call_tech_valid	如需使用call_tech, 该参数需要设置为TRUE
mcm_data_bearer_tech_info_t_v01	call_tech	Call ID对应的数据承载技术
uint8_t	reg_status_valid	如需使用reg_status, 该参数需要设置为TRUE
mcm_data_reg_status_t_v01	reg_status	数据调制解调器注册状态
uint8_t	dorm_status_valid	如需使用dorm_status, 该参数需要设置为TRUE
mcm_data_dormancy_state_t_v01	dorm_status	数据休眠状态
uint8_t	addr_count_valid	如需使用addr_count, 该参数需要设置为TRUE
int8_t	addr_count	数据设备地址个数
uint8_t	addr_info_valid	如需使用addr_info, 该参数需要设置为TRUE
uint32_t	addr_info_len	必须设置为addr_info里地址元素个数

mcm_data_- addr_t_info_ - v01	addr_info	数据设备地址
uint8_t	vce_reason_ valid	如需使用vce_reason，该参数需要设置为TRUE
mcm_data_- verbose_call_ end_reason_t _- v01	vce_reason	长字符串格式的数据呼叫终止原因

6.3 数据相关常量

该节主要包含数据业务相关常量

6.3.1 宏定义集

6.3.1.1 **#define MCM_DATA_NET_UP_EV_V01 0x00005001**

表示数据呼叫连接上的EVENT ID

6.3.1.2 **#define MCM_DATA_NET_DOWN_EV_V01 0x00005002**

表示数据呼叫断开的Event ID

6.3.1.3 **#define MCM_DATA_NET_NEW_ADDR_EV_V01 0x00005003**

表示接口配置新IP地址的Event ID

6.3.1.4 **#define MCM_DATA_NET_DEL_ADDR_EV_V01 0x00005004**

表示其中一个IP地址丢失的Event ID

6.3.1.5 **#define MCM_DATA_REG_SRVC_STATUS_EV_V01 0x00005005**

表示当前调制解调器服务状态的Event ID

6.3.1.6 **#define MCM_DATA_BEARER_TECH_STATUS_EV_V01 0x00005006**

表示当前呼叫承载技术的Event ID

6.3.1.7 **#define MCM_DATA_DORMANCY_STATUS_EV_V01 0x00005007**

表示当前呼叫休眠状态的Event ID

6.3.1.8 **#define MCM_DATA_MAX_APN_LEN_V01 150**

APN的最大长度

6.3.1.9 **#define MCM_DATA_MAX_USERNAME_LEN_V01 127**

user name的最大长度

6.3.1.10 **#define MCM_DATA_MAX_PASSWORD_LEN_V01 127**

password 的最大长度

6.3.1.11 **#define MCM_DATA_MAX_ADDR_LEN_V01 128**

address 的最大长度

6.3.1.12 **#define MCM_DATA_MAX_DEVICE_NAME_LEN_V01 13**

device name的最大长度

6.3.1.13 **#define MCM_DATA_MAX_ADDR_COUNT_V01 10**

IP addresses 的最大个数

6.4 **Data 相关枚举**

该节主要包含MCM data相关枚举

6.4.1 **枚举类型集**

6.4.1.1 **enum mcm_data_call_status_t_v01**

成员:

MCM_DATA_CALL_STATE_INVALID_V01 呼叫状态无效
MCM_DATA_CALL_STATE_CONNECTING_V01 呼叫正在建立连接
MCM_DATA_CALL_STATE_CONNECTED_V01 呼叫已连接
MCM_DATA_CALL_STATE_DISCONNECTING_V01 呼叫正在断开
MCM_DATA_CALL_STATE_DISCONNECTED_V01 呼叫已断开

6.4.1.2 **enum mcm_data_srv_status_t_v01**

成员:

MCM_DATA_MODEM_STATE_OOS_V01 调制解调器停止服务

MCM_DATA_MODEM_STATE_IN_SERVICE_V01 调制解调器正在运行

6.4.1.3 enum mcm_data_bearer_tech_info_t_v01

成员:

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 DC HSDPA+.

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.

6.4.1.4 enum mcm_data_dormancy_state_t_v01

成员:

MCM_DATA_DORMANCY_STATE_PHYSLINK_ACTIVE_V01 呼叫未休眠

MCM_DATA_DORMANCY_STATE_PHYSLINK_DORMANT_V01 呼叫休眠

6.4.1.5 enum mcm_data_call_end_reason_type_t_v01

成员:

MCM_DATA_TYPE_UNSPECIFIED_V01 Unspecified.

MCM_DATA_TYPE_MOBILE_IP_V01 Mobile IP.

MCM_DATA_TYPE_INTERNAL_V01 Internal.

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 Point-to-Point protocol.

MCM_DATA_TYPE_EHRPD_V01 EHRPD.

MCM_DATA_TYPE_IPV6_V01 IPV6.

6.4.1.6 enum mcm_data_call_end_reason_code_t_v01

成员:

MCM_DATA_CE_INVALID_V01 无效的呼叫终止原因

MCM_DATA_CE_MIP_FA_ERR_REASON_UNSPECIFIED_V01 Mobile IP; 未知错误.

MCM_DATA_CE_MIP_FA_ERR_ADMINISTRATIVELY_PROHIBITED_V01 Mobile IP; 管理禁止

MCM_DATA_CE_MIP_FA_ERR_INSUFFICIENT_RESOURCES_V01 Mobile IP; 资源不足

MCM_DATA_CE_MIP_FA_ERR_MOBILE_NODE_AUTHENTICATION_FAILURE_V01 Mobile IP; 移动节点身份验证失败

MCM_DATA_CE_MIP_FA_ERR_HA_AUTHENTICATION_FAILURE_V01 Mobile IP; HA 认证失败

MCM_DATA_CE_MIP_FA_ERR_REQUESTED_LIFETIME_TOO_LONG_V01 Mobile IP;
请求的生命周期过长

MCM_DATA_CE_MIP_FA_ERR_MALFORMED_REQUEST_V01 Mobile IP; 有缺陷的请求

MCM_DATA_CE_MIP_FA_ERR_MALFORMED_REPLY_V01 Mobile IP; 有缺陷的回复

MCM_DATA_CE_MIP_FA_ERR_ENCAPSULATION_UNAVAILABLE_V01 Mobile IP; 封装不可用

MCM_DATA_CE_MIP_FA_ERR_VJHC_UNAVAILABLE_V01 Mobile IP; VJHC 不可用

MCM_DATA_CE_MIP_FA_ERR_REVERSE_TUNNEL_UNAVAILABLE_V01 Mobile IP; 反向通道
不可用

MCM_DATA_CE_MIP_FA_ERR_REVERSE_TUNNEL_IS_MANDATORY_AND_T_BIT_NOT_SET_V01
Mobile IP; 反向通道是强制性的, T-bit 没有设置

MCM_DATA_CE_MIP_FA_ERR_DELIVERY_STYLE_NOT_SUPPORTED_V01 Mobile IP;
交互方式不支持

MCM_DATA_CE_MIP_FA_ERR_MISSING_NAI_V01 Mobile IP; NAI 丢失.

MCM_DATA_CE_MIP_FA_ERR_MISSING_HA_V01 Mobile IP; HA 丢失.

MCM_DATA_CE_MIP_FA_ERR_MISSING_HOME_ADDR_V01 Mobile IP; 本地地址丢失

MCM_DATA_CE_MIP_FA_ERR_UNKNOWN_CHALLENGE_V01 Mobile IP; unknown challenge.

MCM_DATA_CE_MIP_FA_ERR_MISSING_CHALLENGE_V01 Mobile IP; missing challenge.

MCM_DATA_CE_MIP_FA_ERR_STALE_CHALLENGE_V01 Mobile IP; stale challenge.

MCM_DATA_CE_MIP_HA_ERR_REASON_UNSPECIFIED_V01 Mobile IP; 未知原因

MCM_DATA_CE_MIP_HA_ERR_ADMINISTRATIVELY_PROHIBITED_V01 Mobile IP; 管理禁止

MCM_DATA_CE_MIP_HA_ERR_INSUFFICIENT_RESOURCES_V01 Mobile IP; 资源不足

MCM_DATA_CE_MIP_HA_ERR_MOBILE_NODE_AUTHENTICATION_FAILURE_V01 Mobile IP; 移动节点身份验证失败

MCM_DATA_CE_MIP_HA_ERR_FA_AUTHENTICATION_FAILURE_V01 Mobile IP; FA 认证失败

MCM_DATA_CE_MIP_HA_ERR_REGISTRATION_ID_MISMATCH_V01 Mobile IP; 注册 ID 不匹配

MCM_DATA_CE_MIP_HA_ERR_MALFORMED_REQUEST_V01 Mobile IP; 有缺陷的请求

MCM_DATA_CE_MIP_HA_ERR_UNKNOWN_HA_ADDR_V01 Mobile IP; 未知 HA 地址

MCM_DATA_CE_MIP_HA_ERR_REVERSE_TUNNEL_UNAVAILABLE_V01 Mobile IP; 反向通道不可用

MCM_DATA_CE_MIP_HA_ERR_REVERSE_TUNNEL_IS_MANDATORY_AND_T_BIT_NOT_SET_V01
Mobile IP; 反向通道是强制性的, 且 T-bit 没有设置

MCM_DATA_CE_MIP_HA_ERR_ENCAPSULATION_UNAVAILABLE_V01 Mobile IP; 封装不可用
MCM_DATA_CE_MIP_ERR_REASON_UNKNOWN_V01 Mobile IP; 未知原因
MCM_DATA_CE_INTERNAL_ERROR_V01 内部错误
MCM_DATA_CE_CALL_ENDED_V01 内部错误, 呼叫终止
MCM_DATA_CE_INTERNAL_UNKNOWN_CAUSE_CODE_V01 内部错误, 未知原因
MCM_DATA_CE_UNKNOWN_CAUSE_CODE_V01 内部错误, 未知原因
MCM_DATA_CE_CLOSE_IN_PROGRESS_V01 内部错误, 正在关闭
MCM_DATA_CE_NW_INITIATED_TERMINATION_V01 内部错误, NW-initiated 终止
MCM_DATA_CE_APP_PREEMPTED_V01 内部错误, 应用程序被抢占
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; 不允许 GPRS和 non-GPRS 服务
MCM_DATA_CE_GPRS_SERVICES_NOT_ALLOWED_V01 CDMA; 不允许 GPRS services
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 CDMA; implicitly detached.
MCM_DATA_CE_PLMN_NOT_ALLOWED_V01 CDMA; PLMN is not allowed.
MCM_DATA_CE_LA_NOT_ALLOWED_V01 CDMA; LA is not allowed.
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 noncompatible 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.

6.5 Data 数据结构

该节主要包含MCM data数据结构

6.5.1 数据结构文档

6.5.1.1 struct mcm_data_reg_status_t_v01

数据域

Type	参数	描述
mcm_data_srv_status_t_v01	srv_status	标识调制解调器的服务状态
mcm_data_bearer_tech_info_t_v01	tech_info	标识优选技术类型

6.5.1.2 struct mcm_data_pkt_stats_t_v01

数据域

Type	参数	描述
uint32_t	pkts_tx	传输的数据包数量
uint32_t	pkts_rx	接收的数据包数量
uint64_t	bytes_tx	传输的字节数
uint64_t	bytes_rx	接收的字节数
uint32_t	pkts_dropped_tx	丢弃的传输数据包数量
uint32_t	pkts_dropped_rx	丢弃的接收数据包数量

6.5.1.3 struct mcm_data_channel_rate_t_v01

数据域

Type	参数	描述
uint32_t	current_tx_rate	当前信道Tx数据速率
uint32_t	current_rx_rate	当前信道Rx数据速率
uint32_t	max_tx_rate	最大Tx信道数据速率
uint32_t	max_rx_rate	最大Rx信道数据速率

6.5.1.4 struct mcm_data_addr_t_v01

数据域

Type	参数	描述
char	valid_addr	表示一个有效的地址是否可用
uint8_t	addr	存储的IP地址

6.5.1.5 struct mcm_data_addr_t_info_v01

数据域

Type	参数	描述
mcm_data_addr_t_v01	iface_addr_s	网络接口的地址
uint32_t	iface_mask	子网掩码
mcm_data_addr_t_v01	gtwy_addr_s	网关服务器地址
uint32_t	gtwy_mask	网关掩码
mcm_data_addr_t_v01	dnsp_addr_s	主DNS服务器地址
mcm_data_addr_t_v01	dnss_addr_s	辅助DNS服务器地址

6.5.1.6 struct mcm_data_verbose_call_end_reason_t_v01

数据域

Type	参数	描述
------	----	----

mcm_data_call_end_reason_type_t_v01	call_end_reason_type	呼叫终止原因类型如下： • 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	长字符串格式的数据呼叫终止原因

827455031@qq.com 47.95.190.67 2019/7/18 11:49:59
 GOSUNCN Confidential

7 SMS

本章主要描述SMS相关的数据结构，宏，枚举类型。

- [SMS Message Identifiers](#)
- [SMS Message Structures](#)
- [SMS Constants](#)
- [SMS Enumerations](#)
- [SMS Data Structures](#)

7.1 SMS消息标识符

本节主要包含MCM SMS相关消息标识符

```
#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
```

7.2 SMS消息结构体

本节主要包含SMS相关消息结构体。

7.2.1 数据结构文档

7.2.1.1 struct mcm_sms_set_service_center_cfg_type_req_msg_v01

请求消息;设置服务中心配置类型

数据域

Type	参数	描述
char	service_center- _addr	服务中心地址
uint8_t	validity_time_ - valid	如需使用validity_time, 该参数需要设置为TRUE
int64_t	validity_time	有效时间

7.2.1.2 struct mcm_sms_set_service_center_cfg_type_resp_msg_v01

响应消息; 设置服务中心配置类型

数据域

Type	参数	描述
mcm_response- _t_v01	response	结果码.

7.2.1.3 struct mcm_sms_get_service_center_cfg_type_resp_msg_v01

响应消息; 获取服务中心配置类型

数据域

Type	参数	描述
mcm_response- _t_v01	response	结果码.
uint8_t	service_center- _addr_valid	如需使用service_center_addr, 该参数需要设置为TRUE
char	service_center- _addr	服务中心地址

uint8_t	validity_time_-_valid	如需使用validity_time，该参数需要设置为TRUE
int64_t	validity_time	有效时间

7.2.1.4 struct mcm_sms_send_mo_msg_req_msg_v01

请求消息; 发送MO消息

数据域

Type	参数	描述
mcm_sms_-_msg_format_t_-_v01	message_-_format	消息格式
char	message_-_content	消息内容
char	destination	目的地址
uint8_t	size_validation_-_valid	如需使用size_validation，该参数需要设置为TRUE
mcm_sms_-_msg_size_-_validation_-_mode_t_v01	size_validation	大小确认

7.2.1.5 struct mcm_sms_send_mo_msg_resp_msg_v01

响应消息; 发送MO消息

数据域

Type	参数	描述
mcm_response_-_t_v01	response	结果码.

7.2.1.6 struct mcm_sms_set_msg_config_req_msg_v01

请求消息; 设置消息配置项

数据域

Type	参数	描述
------	----	----

uint8_t	default_size_validation_mode_valid	如需使用default_size_validation_mode，该参数需要设置为TRUE
mcm_sms_msg_size_validation_mode_t_v01	default_size_validation_mode	默认size验证模式
uint8_t	enable_cb_valid	如需使用enable_cb，该参数需要设置为TRUE
uint8_t	enable_cb	使能回调

7.2.1.7 struct mcm_sms_set_msg_config_resp_msg_v01

响应消息;设置消息配置项

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.

7.2.1.8 struct mcm_sms_get_msg_config_resp_msg_v01

响应消息;获取消息配置项

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.
uint8_t	default_size_validation_mode_valid	如需使用default_size_validation_mode，该参数需要设置为TRUE
mcm_sms_msg_size_validation_mode_t_v01	default_size_validation_mode	默认size验证模式
uint8_t	enable_cb_valid	如需使用enable_cb，该参数需要设置为TRUE
uint8_t	enable_cb	使能回调

7.2.1.9 struct mcm_sms_set_reception_mode_req_msg_v01

请求消息; 设置接收模式。

数据域

Type	参数	描述
mcm_sms_reception_mode_t_v01	reception_mode	接收模式
uint8_t	last_absorbed_message_id_valid	如需使用last_absorbed_message_id, 该参数需要设置为TRUE
int64_t	last_absorbed_message_id	Last absorbed message ID.

7.2.1.10 struct mcm_sms_set_reception_mode_resp_msg_v01

响应消息; 设置接收模式。

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.

7.2.1.11 struct mcm_sms_event_register_req_msg_v01

请求消息; 注册一个指示事件

数据域

Type	参数	描述
uint8_t	register_sms_pp_event_valid	如需使用register_sms_pp_event, 该参数需要设置为TRUE
uint8_t	register_sms_pp_event	收到一个PP SMM事件

7.2.1.12 struct mcm_sms_event_register_resp_msg_v01

响应消息; 注册一个指示事件

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.

7.2.1.13 struct mcm_sms_pp_ind_msg_v01

指示消息;点对点消息指示

数据域

Type	参数	描述
mcm_sms_msg_format_t_v01	message_format	消息格式
Uint16_t	Message_len	消息内容长度
char	message_content	消息内容
char	source_address	Source address.
int64_t	message_id	Message ID.
uint8_t	message_class_valid	如需使用message_class, 该参数需要设置为TRUE
mcm_sms_message_class_t_v01	message_class	Message class.

7.2.1.14 struct mcm_sms_cb_ind_msg_v01

指示消息; 小区广播消息指示

数据域

Type	参数	描述
mcm_sms_msg_format_t_v01	message_format	消息格式
char	message_content	消息内容

7.2.1.15 struct mcm_sms_cb_cmas_ind_msg_v01

指示消息; 小区广播消息指示

数据域

Type	参数	描述
uint8_t	type_0_record_valid	如需使用type_0_record, 该参数需要设置为TRUE

mcm_cbs_- cmae_record_- type_0_t_v01	type_0_record	Type 0 record.
uint8_t	type_1_record- _valid	如需使用type_1_record，该参数需要设置为TRUE
mcm_cbs_- cmae_record_- type_1_t_v01	type_1_record	Type 1 record.
uint8_t	type_2_record- _valid	如需使用type_2_record，该参数需要设置为TRUE
mcm_cbs_- cmae_record_- type_2_t_v01	type_2_record	Type 2 record.

7.3 SMS 相关常量

本节主要包含MCM DM相关常量

7.3.1 宏定义集

7.3.1.1 #define MCM_SMS_MAX_MO_MSG_LENGTH_V01 1440

MO SMS (9*160)的最大长度

7.3.1.2 #define MCM_SMS_MAX_MT_MSG_LENGTH_V01 160

SMS 的最大长度

7.3.1.3 #define MCM_SMS_MAX_ADDR_LENGTH_V01 252

最大字符串长度

7.4 SMS 相关枚举

本节主要包含MCC SMS相关枚举类型

7.4.1 枚举类型集

7.4.1.1 enum mcm_sms_msg_format_t_v01

成员:

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.
MCM_SMS_MSG_FORMAT_PDU_UCS2_V01 Message format UCS2.

7.4.1.2 enum mcm_sms_msg_size_validation_mode_t_v01

成员:

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.

7.4.1.3 enum mcm_sms_reception_mode_t_v01

成员:

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.

7.4.1.4 enum mcm_sms_message_class_t_v01

成员:

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.

7.4.1.5 enum mcm_cbs_cmae_category_type_t_v01

成员:

MCM_CBS_CMAE_CATEGORY_GEO_V01 Geophysical, including landslide.
MCM_CBS_CMAE_CATEGORY_MET_V01 Meteorological, including flood.
MCM_CBS_CMAE_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 threat or attack).

MCM_CBS_CMAE_CATEGORY_OTHER_V01 Other events.

7.4.1.6 enum mcm_cbs_cmae_response_type_t_v01

成员:

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).

7.4.1.7 enum mcm_cbs_cmae_severity_type_t_v01

成员:

MCM_CBS_CMAE_SEVERITY_EXTREME_V01 Extreme (extraordinary threat to life or property).

MCM_CBS_CMAE_SEVERITY_SEVERE_V01 Severe (significant threat to life or property).

7.4.1.8 enum mcm_cbs_cmae_urgency_type_t_v01

成员:

MCM_CBS_CMAE_URGENCY_IMMEDIATE_V01 Immediate (responsive action should be taken immediately).

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

7.4.1.9 enum mcm_cbs_cmae_certainty_type_t_v01

成员:

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

MCM_CBS_CMAE_CERTAINTY_LIKELY_V01 Likely (probability > ~50%).

7.5 SMS数据结构

本节主要包含MCM SMS相关数据结构

7.5.1 数据结构文档

7.5.1.1 struct mcm_cbs_cmae_expire_t_v01

数据域

Type	参数	描述
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).

7.5.1.2 struct mcm_cbs_cmae_record_type_0_t_v01

数据域

Type	参数	描述
uint32_t	message_ content_len	如需使用message_content，该参数需要设置为TRUE
char	message_ content	消息内容

7.5.1.3 struct mcm_cbs_cmae_record_type_1_t_v01

数据域

Type	参数	描述
mcm_cbs_ cmae_category- _type_t_v01	category	CMAS告警的分类
mcm_cbs_ cmae_response- _type_t_v01	response	表明是CMAS告警的响应
mcm_cbs_ cmae_severity- _type_t_v01	severity	严重 CMAS告警.
mcm_cbs_ cmae_urgency- _type_t_v01	urgency	紧急CMAS告警

mcm_cbs_- cmae_certainty- _type_t_v01	certainty	确定性CMAS 告警
---	-----------	------------

7.5.1.4 struct mcm_cbs_cmae_record_type_2_t_v01

数据域

Type	参数	描述
uint16_t	id	消息标识
uint8_t	alert_handling	表明此告警消息是否需要特殊处理
mcm_cbs_- cmae_expire_t- _v01	expire	CMAS警报的到期日期和时间
uint8_t	language	消息内容的使用的语言

8 移动接入点(Mobile AP)

该章节主要包含使能/禁止移动接入点(Mobile Access Point, Mobile AP)功能、连接、获取其他Mobile AP-related配置信息APIs。

- [Mobile AP MessageIdentifiers](#)
- [Mobile AP Message Structures](#)
- [Mobile AP Constants](#)
- [Mobile Ap Enumerations](#)
- [Mobile AP Data Structures](#)

8.1 Mobile AP消息标识符

该节主要包含MCM Mobile AP相关消息标识符

```
#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_MOBILEAP_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_UNSQL_EVENT_IND_V01 0x0427
```

8.2 Mobile AP消息结构体

本节主要包含MCM Mobile AP消息结构体

8.2.1 数据结构文档

8.2.1.1 struct mcm_mobileap_enable_resp_msg_v01

响应消息; 通过一个mobile AP对象使能ARM® Cortex™-A5处理器上mobile AP功能
数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.
uint8_t	mcm_- mobileap_- handle_valid	如需使用mcm_mobileap_handle, 该参数需要设置为TRUE
uint32_t	mcm_- mobileap_- handle	Mobile AP 句柄

8.2.1.2 struct mcm_mobileap_disable_req_msg_v01

请求消息;通过一个mobile AP对象禁止ARM® Cortex™-A5处理器上mobile AP功能
数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.3 struct mcm_mobileap_disable_resp_msg_v01

响应消息;通过一个mobile AP对象禁止ARM® Cortex™-A5处理器上mobile AP功能

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.4 struct mcm_mobileap_bring_up_wwan_req_msg_v01

请求消息;请求从mobile AP建立WWAN

使用存储的网络策略,即通过MCM_MOBILEAP_ENABLE_REQ使能mobile AP来建立呼叫。如果控制面在短时间内发起多个请求,将会返回表示之前请求仍在进行中的MCM_ERROR_NO_EFFECT错误。

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint8_t	ip_version_-valid	如需使用ip_version, 该参数需要设置为TRUE
mcm_-mobileap_ip-_version_t_v01	ip_version	IP version

8.2.1.5 struct mcm_mobileap_bring_up_wwan_resp_msg_v01

响应消息; 请求从mobile AP建立WWAN

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.6 struct mcm_mobileap_tear_down_wwan_req_msg_v01

请求消息;如果WWAN 呼叫已连上, 挂断该连接。

该命令关断回传功能。如果控制面在短时间内发起多个请求, 将会返回表示之前请求仍在进行中的MCM_ERROR_NO_EFFECT错误。

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint8_t	ip_version_-valid	如需使用ip_version，该参数需要设置为TRUE
mcm_-mobileap_ip-_version_t_v01	ip_version	IP version.

8.2.1.7 struct mcm_mobileap_tear_down_wwan_resp_msg_v01

响应消息; 如果WWAN 呼叫已连上，挂断该连接。

数据域

Type	参数	描述
uint8_t	resp_valid	如需使用resp，该参数需要设置为TRUE
mcm_response-_t_v01	resp	结果码.

8.2.1.8 struct mcm_mobileap_add_static_nat_entry_req_msg_v01

请求消息; 添加一个静态网络地址转换(NAT)条目。

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
mcm_-mobileap_-static_nat_-entry_conf_t_v01	nat_entry_-config	Mobile AP静态网络地址转换(NAT)条目配置

8.2.1.9 struct mcm_mobileap_add_static_nat_entry_resp_msg_v01

响应消息; 添加一个静态网络地址转换(NAT)条目。

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

8.2.1.10 struct mcm_mobileap_get_static_nat_entry_req_msg_v01

请求消息; 请求所有静态网络地址转换(NAT)条目

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint32_t	max_entries	客户端请求的最大SNAT条目数

8.2.1.11 struct mcm_mobileap_get_static_nat_entry_resp_msg_v01

响应消息; 请求所有的静态NAT条目数。响应消息包含的条目数量紧随其顺序列出的条目信息之后

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.
uint8_t	snat_entries_valid	如需使用snat_entries, 该参数需要设置为TRUE
uint32_t	snat_entries_len	必须设置为snat_entries中元素的个数
mcm_mobileap_static_nat_entry_conf_t_v01	snat_entries	MCM mobile AP静态NAT的条目配置

8.2.1.12 struct mcm_mobileap_delete_static_nat_entry_req_msg_v01

请求消息;删除静态NAT

数据域

Type	参数	描述
------	----	----

uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
mcm_- mobileap_- static_nat_- entry_conf_t_ -v01	snat_entry	MCM mobile AP静态NAT条目请求消息

8.2.1.13 struct mcm_mobileap_delete_static_nat_entry_resp_msg_v01

响应消息; 删除静态NAT

数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.

8.2.1.14 struct mcm_mobileap_set_nat_timeout_req_msg_v01

请求消息;配置不同NAT类型的超时时间. 该命令句柄会用当前的值覆盖之前配置的所有值。

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
mcm_- mobileap_nat_- timeout_t_v01	timeout_type	使用的NAT超时类型，如下： ● 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	使用的NAT超时时间，单位为秒。

8.2.1.15 struct mcm_mobileap_set_nat_timeout_resp_msg_v01

响应消息; 配置不同NAT类型的超时时间

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.16 struct mcm_mobileap_get_nat_timeout_req_msg_v01

请求消息;获取配置的NAT超时时间

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值.
mcm_-mobileap_nat_timeout_t_v01	timeout_type	使用的NAT超时类型，如下： ● 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.

8.2.1.17 struct mcm_mobileap_get_nat_timeout_resp_msg_v01

响应消息; 获取配置的NAT超时时间

数据域

Type	参数	描述
mcm_response-_t_v01	resp	
uint8_t	timeout_value-_valid	如需使用timeout_value，该参数需要设置为TRUE
uint32_t	timeout_value	使用的NAT超时时间，单位为秒。

8.2.1.18 struct mcm_mobileap_set_nat_type_req_msg_v01

请求消息;配置NAT类型。该命令句柄会用当前的值覆盖之前配置的所有值。

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值。
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(1) – Port-restricted cone NAT ● MCM_MOBILEAP_NAT_FULL_CONE_NAT (2) – Full cone NAT ● MCM_MOBILEAP_NAT_ADDRESS_RESTRICTED_NAT (3) – Address-restricted NAT

8.2.1.19 struct mcm_mobileap_set_nat_type_resp_msg_v01

响应消息; 配置NAT类型。

数据域

Type	参数	描述
mcm_response_- t_v01	resp	结果码。

8.2.1.20 struct mcm_mobileap_get_nat_type_req_msg_v01

请求消息; 获取设置的NAT类型

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值。

8.2.1.21 struct mcm_mobileap_get_nat_type_resp_msg_v01

响应消息; 获取设置的NAT类型

数据域

Type	参数	描述
------	----	----

mcm_response-_t_v01	resp	结果码.
uint8_t	nat_type_valid	如需使用nat_type, 该参数需要设置为TRUE
mcm_-_mobileap_-_nat_type_t_v01	nat_type	NAT类型, 如下: ● MCM_MOBILEAP_NAT_SYMMETRIC_NAT (0) – Symmetric NAT ● MCM_MOBILEAP_NAT_PORT_RESTRICTED_CONE_NAT(1) – Port-restricted cone NAT ● MCM_MOBILEAP_NAT_FULL_CONE_NAT (2) – Full cone NAT ● MCM_MOBILEAP_NAT_ADDRESS_RESTRICTED_NAT (3) – Address-restricted NAT

8.2.1.22 struct mcm_mobileap_add_firewall_entry_req_msg_v01

请求消息;添加基于IP过滤器的防火墙规则

数据域

Type	参数	描述
uint32_t	mcm_-_mobileap_-_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
mcm_-_mobileap_ip_-_family_t_v01	ip_version	防火墙族版本, 如下: ● 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	如需使用next_hdr_prot, 该参数需要设置为TRUE
uint8_t	next_hdr_prot	在IP报头后的下一个协议报头
uint8_t	tcp_udp_src_valid	如需使用tcp_udp_src, 该参数需要设置为TRUE
mcm_-_mobileap_-_tcp_udp_port_-_range_t_v01	tcp_udp_src	TCP_UDP源端口
uint8_t	tcp_udp_dst_valid	如需使用tcp_udp_dst, 该参数需要设置为TRUE
mcm_-_mobileap_-_tcp_udp_port_-_range_t_v01	tcp_udp_dst	TCP_UDP目的端口

uint8_t	icmp_type_valid	如需使用 icmp_type，该参数需要设置为TRUE
uint8_t	icmp_type	如ICMP协议,RFC 792定义的 ICMP类型
uint8_t	icmp_code_valid	如需使用 icmp_code，该参数需要设置为TRUE
uint8_t	icmp_code	如ICMP协议,RFC 792定义的 ICMP码
uint8_t	esp_spi_valid	如需使用 esp_spi，该参数需要设置为TRUE
uint32_t	esp_spi	如ESP协议,RFC 4303定义的安全参数索引
uint8_t	ip4_src_addr_valid	如需使用 ip4_src_addr，该参数需要设置为TRUE
mcm_- mobileap_ip4- _addr_subnet_- mask_t_v01	ip4_src_addr	IPv4源地址和子网掩码。
uint8_t	ip4_dst_addr_valid	如需使用 ip4_dst_addr，该参数需要设置为TRUE
mcm_- mobileap_ip4- _addr_subnet_- mask_t_v01	ip4_dst_addr	IPv4目的地地址和子网掩码。
uint8_t	ip4_tos_valid	如需使用 ip4_tos，该参数需要设置为TRUE
mcm_- mobileap_- ip4_tos_t_v01	ip4_tos	IPv6 TOS值和掩码
uint8_t	ip6_src_addr_valid	如需使用 ip6_src_addr，该参数需要设置为TRUE
mcm_- mobileap_ip6- _addr_prefix_- len_t_v01	ip6_src_addr	IPv6源地址和前缀长度。
uint8_t	ip6_dst_addr_valid	如需使用 ip6_dst_addr，该参数需要设置为TRUE
mcm_- mobileap_ip6- _addr_prefix_- len_t_v01	ip6_dst_addr	IPv6目的地址和前缀长度。
uint8_t	ip6_trf_cls_valid	如需使用 ip6_trf_cls，该参数需要设置为TRUE
mcm_- mobileap_ip6- _traffic_class_t_- v01	ip6_trf_cls	IPv6 traffic class 值和掩码。

8.2.1.23 struct mcm_mobileap_add_firewall_entry_resp_msg_v01

响应消息;添加基于IP过滤器的防火墙规则

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码
uint8_t	firewall_-handle_valid	如需使用firewall_handle, 该参数需要设置为TRUE
uint32_t	firewall_handle	定义一个增加防火墙规则的句柄

8.2.1.24 struct mcm_mobileap_get_firewall_entries_handle_list_req_msg_v01

请求消息;获取所有防火墙规则的句柄

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
mcm_-mobileap_ip-_family_t_v01	ip_version	定义防火墙族版本, 值如下: ● 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

8.2.1.25 struct mcm_mobileap_get_firewall_entries_handle_list_resp_msg_v01

响应消息; 获取所有防火墙规则的句柄

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.
uint8_t	firewall_-handle_list_-valid	如需使用firewall_handle_list, 该参数需要设置为TRUE
uint32_t	firewall_-handle_list_len	必须设置为firewall_handle_list的元素个数
uint32_t	firewall_-handle_list	句柄定义的防火墙入口, 该句柄值必须是之前MCM_MOBILEAP_ADD_FIREWALL_ENTRY_RESP的返回值

8.2.1.26 struct mcm_mobileap_get_firewall_entry_req_msg_v01

请求消息; 获取防火墙规则

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint32_t	firewall_handle	句柄定义的防火墙入口，该句柄值必须是之前MCM_MOBILEAP_ADD_FIREWALL_ENTRY_RESP或MCM_MOBILEAP_GET_FIREWALL_ENTRIES_HANDLE_LIST_RESP的返回值

8.2.1.27 struct mcm_mobileap_get_firewall_entry_resp_msg_v01

响应消息; 获取防火墙规则

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码。
uint8_t	ip_version_valid	如需使用ip_version，该参数需要设置为TRUE
mcm_mobileap_ip_family_t_v01	ip_version	定义防火墙族版本，值如下： ● 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	如需使用next_hdr_prot，该参数需要设置为TRUE
uint8_t	next_hdr_prot	IPv4/IPv6下一个协议报头
uint8_t	tcp_udp_src_valid	如需使用tcp_udp_src，该参数需要设置为TRUE
mcm_mobileap_tcp_udp_port_range_t_v01	tcp_udp_src	TCP, UDP, 和 TCP_UDP源端口
uint8_t	tcp_udp_dst_valid	如需使用tcp_udp_dst，该参数需要设置为TRUE

mcm_- mobileap_- tcp_udp_port_ -range_t_v01	tcp_udp_dst	TCP, UDP, 和 TCP_UDP 目的端口
uint8_t	icmp_type_ -valid	如需使用icmp_type, 该参数需要设置为TRUE
uint8_t	icmp_type	如ICMP协议,RFC 792定义的 ICMP类型
uint8_t	icmp_code_ -valid	如需使用 icmp_code , 该参数需要设置为TRUE
uint8_t	icmp_code	如ICMP协议,RFC 792定义的 ICMP码
uint8_t	esp_spi_valid	如需使用 esp_spi , 该参数需要设置为TRUE
uint32_t	esp_spi	如ESP协议,RFC 4303定义的安全参数索引
uint8_t	ip4_src_addr_ -valid	如需使用 ip4_src_addr , 该参数需要设置为TRUE
mcm_- mobileap_ip4- _addr_subnet_ -mask_t_v01	ip4_src_addr	IPv4源地址和子网掩码。
uint8_t	ip4_dst_addr_ -valid	如需使用 ip4_dst_addr , 该参数需要设置为TRUE
mcm_- mobileap_ip4- _addr_subnet_ -mask_t_v01	ip4_dst_addr	IPv4目的地地址和子网掩码。
uint8_t	ip4_tos_valid	如需使用 ip4_tos, 该参数需要设置为TRUE
mcm_- mobileap_- ip4_tos_t_v01	ip4_tos	IPv4 TOS 值和掩码
uint8_t	ip6_src_addr_ -valid	如需使用 ip6_src_addr , 该参数需要设置为TRUE
mcm_- mobileap_ip6- _addr_prefix_ -len_t_v01	ip6_src_addr	IPv6源地址和前缀长度。
uint8_t	ip6_dst_addr_ -valid	如需使用 ip6_dst_addr , 该参数需要设置为TRUE
mcm_- mobileap_ip6- _addr_prefix_ -len_t_v01	ip6_dst_addr	IPv6目的地地址和前缀长度
uint8_t	ip6_trf_cls_ -valid	如需使用 ip6_trf_cls , 该参数需要设置为TRUE
mcm_- mobileap_ip6_- traffic_class_t_ -v01	ip6_trf_cls	IPv6 traffic class值和掩码

8.2.1.28 struct mcm_mobileap_delete_firewall_entry_req_msg_v01

请求消息;删除由句柄定义的防火墙规则

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint32_t	firewall_handle	句柄定义的防火墙

8.2.1.29 struct mcm_mobileap_delete_firewall_entry_resp_msg_v01

响应消息; 删除由句柄定义的防火墙规则

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

8.2.1.30 struct mcm_mobileap_set_firewall_config_req_msg_v01

请求消息; 配置防火墙。该命令可使能/禁止防火墙。如果防火墙被使能，会设置防火墙状态为接收/放弃数据包。

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint8_t	firewall_enabled	布尔值，表明防火墙是否被使能
uint8_t	pkts_allowed_valid	如需使用pkts_allowed，该参数需要设置为TRUE
uint8_t	pkts_allowed	布尔值，表明防火墙接收或放弃数据包

8.2.1.31 struct mcm_mobileap_set_firewall_config_resp_msg_v01

响应消息;配置防火墙

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.32 struct mcm_mobileap_add_dmz_req_msg_v01

请求消息;为mobile AP设置DMZ(perimeter network) IP地址。

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint32_t	dmz_ip_addr	DMZ IP 地址

8.2.1.33 struct mcm_mobileap_add_dmz_resp_msg_v01

响应消息; 为mobile AP设置DMZ(perimeter network) IP地址。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.34 struct mcm_mobileap_get_dmz_req_msg_v01

请求消息; 请求mobile AP的DMZ IP地址。

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.35 struct mcm_mobileap_get_dmz_resp_msg_v01

响应消息; 请求mobile AP的DMZ IP地址。如果客户端没有设置DMZ, 将返回0.0.0.0

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.
uint8_t	dmz_ip_addr_-valid	如需使用dmz_ip_addr, 该参数需要设置为TRUE
uint32_t	dmz_ip_addr	DMZ IP 地址

8.2.1.36 struct mcm_mobileap_delete_dmz_req_msg_v01

请求消息;删除DMZ条目或DMZ IP地址

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint32_t	dmz_ip_addr	DMZ IP地址

8.2.1.37 struct mcm_mobileap_delete_dmz_resp_msg_v01

响应消息; 删除DMZ条目或DMZ IP地址

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.38 struct mcm_mobileap_get_ipv4_wwan_config_req_msg_v01

请求消息; 请求WWAN IP配置信息。该命令必须在MCM_MOBILEAP_WWAN_STATUS_IND指示成功建议WWAN后才被控制面发出。

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.39 struct mcm_mobileap_get_ipv4_wwan_config_resp_msg_v01

响应消息; 请求WWAN IP配置信息。

数据域

Type	参数	描述
mcm_response-t_v01	resp	结果码.
uint8_t	v4_addr_valid	如需使用v4_addr, 该参数需要设置为TRUE
uint32_t	v4_addr	IPv4 地址.
uint8_t	v4_prim_dns_addr_valid	如需使用v4_prim_dns_addr, 该参数需要设置为TRUE
uint32_t	v4_prim_dns_addr	IPv4主DNS地址
uint8_t	v4_sec_dns_addr_valid	如需使用v4_sec_dns_addr, 该参数需要设置为TRUE
uint32_t	v4_sec_dns_addr	IPv4辅助DNS地址

8.2.1.40 struct mcm_mobileap_get_wwan_stats_req_msg_v01

请求消息; 获取 WWAN数据.

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
mcm_mobileap_ip_family_t_v01	ip_family	定义使用的IP版本, 值如下: ● 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

8.2.1.41 struct mcm_mobileap_get_wwan_stats_resp_msg_v01

响应消息; 获取 WWAN数据

数据域

Type	参数	描述
mcm_response-t_v01	resp	结果码.

uint8_t	wwan_stats_valid	Must be set to TRUE if wwan_stats is being passed. 如需使用message_content, 该参数需要设置为TRUE
mcm_mobileap_wwan_statistics_t_v01	wwan_stats	WWAN 数据

8.2.1.42 struct mcm_mobileap_reset_wwan_stats_req_msg_v01

请求消息;复位WWAN数据

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
mcm_mobileap_ip_family_t_v01	ip_family	定义使用的IP版本, 值如下: ● 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

8.2.1.43 struct mcm_mobileap_reset_wwan_stats_resp_msg_v01

响应消息; 复位WWAN数据

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

8.2.1.44 struct mcm_mobileap_set_dhcpd_config_req_msg_v01

请求消息; 设置DHCPD配置项

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

mcm_- mobileap_- dhcpcd_config_- t_v01	dhcpcd_config	DHCPD配置项
---	---------------	----------

8.2.1.45 struct mcm_mobileap_set_dhcpcd_config_resp_msg_v01

响应消息; 设置DHCPD配置项

数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.

8.2.1.46 struct mcm_mobileap_enable_wlan_req_msg_v01

请求消息;使能WLAN

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.47 struct mcm_mobileap_enable_wlan_resp_msg_v01

响应消息; 使能WLAN

数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.

8.2.1.48 struct mcm_mobileap_disable_wlan_req_msg_v01

请求消息;禁用WLAN

数据域

Type	参数	描述
------	----	----

uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
----------	-------------------------------	---

8.2.1.49 struct mcm_mobileap_disable_wlan_resp_msg_v01

响应消息; 禁用WLAN

数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.

8.2.1.50 struct mcm_mobileap_set_ipsec_vpn_pass_through_req_msg_v01

请求消息; 配置互联网协议的安全(IPSec)的虚拟私有网络 (VPN)

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint8_t	vpn_pass_- through_value	布尔值, 表示IPSec VPN值是否被允许

8.2.1.51 struct mcm_mobileap_set_ipsec_vpn_pass_through_resp_msg_v01

响应消息; 配置 IPSec VPN。该命令句柄会用当前值覆盖之前配置的值

数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.

8.2.1.52 struct mcm_mobileap_get_ipsec_vpn_pass_through_req_msg_v01

请求消息; 查询IPSec VPN配置

数据域

Type	参数	描述
------	----	----

uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
----------	-------------------------------	---

8.2.1.53 struct mcm_mobileap_get_ipsec_vpn_pass_through_resp_msg_v01

响应消息; 查询IPSec VPN配置

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.
uint8_t	vpn_pass_- through_value- _valid	如需使用vpn_pass_through_value, 该参数需要设置为TRUE
uint8_t	vpn_pass_- through_value	布尔值, 表示IPSec VPN值是否被允许

8.2.1.54 struct mcm_mobileap_set_pptp_vpn_pass_through_req_msg_v01

请求消息; 配置PPTP(Point-to-Point Tunneling Protocol) VPN值, 该命令句柄会用当前值覆盖之前配置的值

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint8_t	vpn_pass_- through_value	布尔值, 表示PPTP VPN值是否被允许

8.2.1.55 struct mcm_mobileap_set_pptp_vpn_pass_through_resp_msg_v01

响应消息; 配置PPTP VPN值

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.56 struct mcm_mobileap_get_pptp_vpn_pass_through_req_msg_v01

请求消息; 查询PPTP VPN值

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.57 struct mcm_mobileap_get_pptp_vpn_pass_through_resp_msg_v01

响应消息; 查询PPTP VPN值

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.
uint8_t	vpn_pass_- through_value- _valid	如需使用vpn_pass_through_value, 该参数需要设置为TRUE
uint8_t	vpn_pass_- through_value	VPN passthrough 值

8.2.1.58 struct mcm_mobileap_set_l2tp_vpn_pass_through_req_msg_v01

请求消息; 配置L2TP(the Layer 2 Tunneling Protocol) VPN, 该命令句柄会用当前值覆盖之前配置的值

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint8_t	vpn_pass_- through_value	布尔值, 表示L2TP VPN值是否被允许

8.2.1.59 struct mcm_mobileap_set_l2tp_vpn_pass_through_resp_msg_v01

响应消息; 配置L2TP(the Layer 2 Tunneling Protocol) VPN

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.60 struct mcm_mobileap_get_l2tp_vpn_pass_through_req_msg_v01

请求消息;查询L2TP(the Layer 2 Tunneling Protocol) VPN设置

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.61 struct mcm_mobileap_get_l2tp_vpn_pass_through_resp_msg_v01

响应消息; 查询L2TP(the Layer 2 Tunneling Protocol) VPN设置

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.
uint8_t	vpn_pass_-through_value-_valid	如需使用vpn_pass_through_value, 该参数需要设置为TRUE
uint8_t	vpn_pass_-through_value	布尔值, 表示L2TP VPN值是否被允许

8.2.1.62 struct mcm_mobileap_set_auto_connect_req_msg_v01

请求消息;使能/禁止自动连接

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint8_t	enable	使能/禁止自动连接: TRUE – Enable FALSE – Disable

8.2.1.63 struct mcm_mobileap_set_auto_connect_resp_msg_v01

响应消息; 使能/禁止自动连接

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.64 struct mcm_mobileap_get_auto_connect_req_msg_v01

请求消息; 获取自动连接标志位

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.65 struct mcm_mobileap_get_auto_connect_resp_msg_v01

获取自动连接标志位

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.
uint8_t	auto_conn_-flag_valid	如需使用auto_conn_flag, 该参数需要设置为TRUE
uint8_t	auto_conn_flag	自动连接状态, 值如下: TRUE – Enabled FALSE – Disabled

8.2.1.66 struct mcm_mobileap_set_roaming_pref_req_msg_v01

请求消息;配置QCMAP_MSGR是否在漫游时发起WWAN数据呼叫。

漫游模式决定创建新的数据呼叫的QCMAP_MSGR策略。默认情况下,这是假定不发起呼叫的。如果通过这个接口进行修改, 将会永久保存设定值。

注意:

漫游模式不会影响当前已经建立起来的数据连接。如，假如漫游模式设置为不在漫游时发起WWAN呼叫，但是此时已存在一个建立的漫游数据连接（如通过不同的客户端或者因为在mode等于TURE的实际建立起来的连接），QCMAP_MSGR使用当前已经建立的WWAN数据连接

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
uint8_t	allow_wwan_-calls_while_-roaming	布尔值，漫游模式。表示在漫游时QCMAP_MSGR是否发起WWAN数据呼叫

8.2.1.67 struct mcm_mobileap_set_roaming_pref_resp_msg_v01

响应消息;配置QCMAP_MSGR是否在漫游时发起WWAN数据呼叫。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

8.2.1.68 struct mcm_mobileap_get_roaming_pref_req_msg_v01

请求消息;获取漫游标志位

数据域

Type	参数	描述
uint32_t	mcm_-mobileap_-handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.69 struct mcm_mobileap_get_roaming_pref_resp_msg_v01

响应消息; 获取漫游标志位

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

uint8_t	allow_wwan_calls_while_roaming_valid	如需使用allow_wwan_calls_while_roaming, 该参数需要设置为TRUE
uint8_t	allow_wwan_calls_while_roaming	布尔值, 用来决定在漫游时mobile AP是否建立数据连接

8.2.1.70 struct mcm_mobileap_set_dualap_config_req_msg_v01

请求消息; 设置在漫游时mobile AP是否发起WWAN数据呼叫。

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值
mcm_mobileap_dualap_config_t_v01	dualap_config	移动接入点 (mobile AP) 双重SSID配置

8.2.1.71 struct mcm_mobileap_set_dualap_config_resp_msg_v01

响应消息; 设置在漫游时mobile AP是否发起WWAN数据呼叫

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

8.2.1.72 struct mcm_mobileap_station_mode_enable_req_msg_v01

请求消息; 使能调制解调器Modem上移动接入点实例的STA mode(STA)功能。
 当该请求信息被成功处理后, 所有数据包通过WLAN station连接到外部网络。调制解调器路由引擎会合理的处理从调制解调器发出或者接收的数据包路由

数据域

Type	参数	描述
uint32_t	mcm_mobileap_handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

mcm_- mobileap_- sta_connection- _config_t_v01	cfg	STA mode配置表明是动态或静态IP配置
---	-----	------------------------

8.2.1.73 struct mcm_mobileap_station_mode_enable_resp_msg_v01

响应消息;使能调制解调器Modem上移动接入点实例的STA mode(STA)功能。

数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.

8.2.1.74 struct mcm_mobileap_station_mode_disable_req_msg_v01

请求消息;禁止调制解调器Modem上移动接入点实例的STA mode(STA)功能。当该请求信息被成功处理后,如果使能自动连接,控制面将会请求从mobile AP建立WWAN连接

数据域

Type	参数	描述
uint32_t	mcm_- mobileap_- handle	用来标识mobile AP的呼叫实例的句柄。该句柄必须是之前MCM_MOBILEAP_ENABLE_REQ的返回值

8.2.1.75 struct mcm_mobileap_station_mode_disable_resp_msg_v01

响应消息;禁止调制解调器Modem上移动接入点实例的STA mode(STA)功能。

数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.

8.2.1.76 struct mcm_mobileap_event_register_req_msg_v01

请求消息;事件注册

数据域

Type	参数	描述
------	----	----

uint8_t	register_event_enabled_valid	如需使用register_event_enabled，该参数需要设置为TRUE
uint8_t	register_event_enabled	使能事件注册
uint8_t	register_event_lan_connecting_valid	如需使用register_event_lan_connecting，该参数需要设置为TRUE
uint8_t	register_event_lan_connecting	注册LAN正在连接事件
uint8_t	register_event_lan_connecting_fail_valid	如需使用register_event_lan_connecting_fail，该参数需要设置为TRUE
uint8_t	register_event_lan_connecting_fail	注册LAN连接失败事件
uint8_t	register_event_lan_ipv6_connecting_fail_valid	如需使用register_event_lan_ipv6_connecting_fail，该参数需要设置为TRUE
uint8_t	register_event_lan_ipv6_connecting_fail	注册LAN IPv6连接失败事件
uint8_t	register_event_lan_connected_valid	如需使用 register_event_lan_connected，该参数需要设置为TRUE
uint8_t	register_event_lan_connected	注册LAN已连接事件
uint8_t	register_event_sta_connected_valid	如需使用 register_event_sta_connected，该参数需要设置为TRUE
uint8_t	register_event_sta_connected	注册STA已连接事件
uint8_t	register_event_lan_ipv6_connected_valid	如需使用 register_event_lan_ipv6_connected，该参数需要设置为TRUE
uint8_t	register_event_lan_ipv6_connected	注册 LAN IPv6已连接事件
uint8_t	register_event_wan_connecting_valid	如需使用 register_event_wan_connecting，该参数需要设置为TRUE
uint8_t	register_event_wan_connecting	注册 WAN正在连接事件

uint8_t	register_event_wan_connecting_fail_valid	如需使用 register_event_wan_connecting_fail，该参数需要设置为TRUE
uint8_t	register_event_wan_connecting_fail	注册 WAN连接失败事件
uint8_t	register_event_wan_ipv6_connecting_fail_valid	如需使用 register_event_wan_ipv6_connecting_fail，该参数需要设置为TRUE
uint8_t	register_event_wan_ipv6_connecting_fail	注册 WAN IPv6连接失败事件
uint8_t	register_event_wan_connected_valid	如需使用 register_event_wan_connected，该参数需要设置为TRUE
uint8_t	register_event_wan_connected	注册 WAN 已连接事件
uint8_t	register_event_wan_ipv6_connected_valid	如需使用 register_event_wan_ipv6_connected，该参数需要设置为TRUE
uint8_t	register_event_wan_ipv6_connected	注册 WAN IPv6已连接事件
uint8_t	register_event_wan_disconnected_valid	如需使用 register_event_wan_disconnected，该参数需要设置为TRUE
uint8_t	register_event_wan_disconnected	注册 WAN连接已断开事件
uint8_t	register_event_wan_ipv6_disconnected_valid	如需使用 register_event_wan_ipv6_disconnected，该参数需要设置为TRUE
uint8_t	register_event_wan_ipv6_disconnected	注册 WAN IPv6连接已断开事件
uint8_t	register_event_lan_disconnected_valid	如需使用 register_event_lan_disconnected，该参数需要设置为TRUE

uint8_t	register_event_lan_disconnected	注册 LAN已断开事件
uint8_t	register_event_lan_ipv6_disconnected_valid	如需使用 register_event_lan_ipv6_disconnected，该参数需要设置为TRUE
uint8_t	register_event_lan_ipv6_disconnected	注册 LAN IPv6已断开事件
uint8_t	register_event_disabled_valid	如需使用 register_event_disabled，该参数需要设置为TRUE
uint8_t	register_event_disabled	禁止事件注册

8.2.1.77 struct mcm_mobileap_event_register_resp_msg_v01

响应消息; 事件注册

数据域

Type	参数	描述
mcm_response_t_v01	response	结果码.

8.2.1.78 struct mcm_mobileap_unsol_event_ind_msg_v01

- 请求消息; 指示对应的已注册的主动上报事件

数据域

Type	参数	描述
------	----	----

int32_t	event_id	事件ID,值如下:
		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

8.3 Mobile AP相关常量

该节主要包含MCM 移动接入点的相关常量信息

8.3.1 宏定义集

8.3.1.1 #define MCM_MOBILEAP_ENABLED_EV_V01 0x8000

事件使能

8.3.1.2 #define MCM_MOBILEAP_LAN_CONNECTING_EV_V01 0x8001

LAN 连接中

8.3.1.3 #define MCM_MOBILEAP_LAN_CONNECTING_FAIL_EV_V01 0x8002

LAN连接失败

8.3.1.4 #define MCM_MOBILEAP_LAN_IPv6_CONNECTING_FAIL_EV_V01 0x8003

LAN IPv6 连接失败

8.3.1.5 #define MCM_MOBILEAP_LAN_CONNECTED_EV_V01 0x8004

LAN 已连接

8.3.1.6 **#define MCM_MOBILEAP_STA_CONNECTED_EV_V01 0x8005**

Station connected.

8.3.1.7 **#define MCM_MOBILEAP_LAN_IPv6_CONNECTED_EV_V01 0x8006**

LAN IPv6已连接

8.3.1.8 **#define MCM_MOBILEAP_WAN_CONNECTING_EV_V01 0x8007**

WAN连接中

8.3.1.9 **#define MCM_MOBILEAP_WAN_CONNECTING_FAIL_EV_V01 0x8008**

WAN连接失败

8.3.1.10 **#define MCM_MOBILEAP_WAN_IPv6_CONNECTING_FAIL_EV_V01 0x8009**

WAN IPv6连接失败

8.3.1.11 **#define MCM_MOBILEAP_WAN_CONNECTED_EV_V01 0x800A**

WAN已连接

8.3.1.12 **#define MCM_MOBILEAP_WAN_IPv6_CONNECTED_EV_V01 0x800B**

WAN IPv6已连接

8.3.1.13 **#define MCM_MOBILEAP_WAN_DISCONNECTED_EV_V01 0x800C**

WAN未连接

8.3.1.14 **#define MCM_MOBILEAP_WAN_IPv6_DISCONNECTED_EV_V01 0x800D**

WAN IPv6未连接

8.3.1.15 **#define MCM_MOBILEAP_LAN_DISCONNECTED_EV_V01 0x800E**

LAN 未连接

8.3.1.16 **#define MCM_MOBILEAP_LAN_IPv6_DISCONNECTED_EV_V01 0x800F**

LAN IPv6 未连接

8.3.1.17 #define MCM_MOBILEAP_DISABLED_EV_V01 0x8010

事件禁止

8.3.1.18 #define MCM_MOBILEAP_MAX_FIREWALL_ENTRIES_V01 50

防火墙最大条目

8.3.1.19 #define MCM_MOBILEAP_MAX_STATIC_NAT_ENTRIES_V01 50

静态NAT条目

8.3.1.20 #define MCM_MOBILEAP_IPV6_ADDR_LEN_V01 16

IPv6地址长度

8.3.1.21 #define MCM_MOBILEAP_MAC_ADDR_LEN_V01 6

MAC地址长度

8.3.1.22 #define MCM_MOBILEAP_DEVICE_NAME_MAX_V01 100

设备名称最大长度

8.3.1.23 #define MCM_MOBILEAP_LEASE_TIME_LEN_V01 100

Maximum lease time length.

8.3.1.24 #define MCM_MOBILEAP_MSG_TIMEOUT_VALUE_V01 90000

SYNC msgs 的最大超时时间(单位：毫秒).

8.4 Mobile Ap相关枚举

该节主要包含MCM移动接入点的相关枚举类型

8.4.1 枚举类型集

8.4.1.1 enum mcm_mobileap_nat_type_t_v01

成员:

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).

8.4.1.2 enum mcm_mobileap_ip_family_t_v01

成员:

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.

8.4.1.3 enum mcm_mobileap_nat_timeout_t_v01

成员:

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).

8.4.1.4 enum mcm_mobileap_ip_version_t_v01

成员:

MCM_MOBILEAP_IP_V4_V01 IPv4.
MCM_MOBILEAP_IP_V6_V01 IPv6.

8.4.1.5 enum mcm_mobileap_wwan_status_t_v01

成员:

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.

8.4.1.6 enum mcm_mobileap_wwan_call_end_type_t_v01

成员:

MCM_MOBILEAP_WWAN_CALL_END_TYPE_INVALID_V01 Unknown.

MCM_MOBILEAP_WWAN_CALL_END_TYPE_MOBILE_IP_V01 Mobile IP.

MCM_MOBILEAP_WWAN_CALL_END_TYPE_INTERNAL_V01 Internal.

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.

8.4.1.7 enum mcm_mobileap_sta_connection_t_v01

成员:

MCM_MOBILEAP_STA_CONNECTION_DYNAMIC_V01 Dynamic station connection.

MCM_MOBILEAP_STA_CONNECTION_STATIC_V01 Static station connection.

8.5 Mobile AP数据结构

该节主要包含MCM移动接入点的相关数据结构

8.5.1 数据结构文档

8.5.1.1 struct mcm_mobileap_wwan_call_end_reason_t_v01

数据域

Type	参数	描述
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.

8.5.1.2 struct mcm_mobileap_ip4_addr_subnet_mask_t_v01

数据域

Type	参数	描述
uint32_t	addr	如IPv4协议，RFC 791定义的IPv4地址
uint32_t	subnet_mask	如IPv4协议，RFC 791定义的IPv4子网掩码

8.5.1.3 struct mcm_mobileap_ip6_addr_prefix_len_t_v01

数据域

Type	参数	描述
uint8_t	addr	如IPv6协议，RFC 2460定义的IPv6地址
uint8_t	prefix_len	如IPv6协议，RFC 3513定义的IPv6 前缀长度

8.5.1.4 struct mcm_mobileap_tcp_udp_port_range_t_v01

数据域

Type	参数	描述
uint16_t	port	如TCP/UDP协议，RFC 793 和RFC 768定义的TCP/UDP端口
uint16_t	range	如TCP/UDP协议，RFC 793 和RFC 768定义的TCP/UDP端口范围

8.5.1.5 struct mcm_mobileap_ip4_tos_t_v01

数据域

Type	参数	描述
uint8_t	value	如IPv4协议，RFC 791定义的TOS值
uint8_t	mask	IPv4 TOS mask

8.5.1.6 struct mcm_mobileap_ip6_traffic_class_t_v01

数据域

Type	参数	描述
uint8_t	value	如IPv6协议，RFC 2460定义的IPv6 traffic class值
uint8_t	mask	IPv6 traffic class mask

8.5.1.7 struct mcm_mobileap_static_nat_entry_conf_t_v01

数据域

Type	参数	描述
------	----	----

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.

8.5.1.8 struct mcm_mobileap_wwan_statistics_t_v01

数据域

Type	参数	描述
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.

8.5.1.9 struct mcm_mobileap_dhcpd_config_t_v01

数据域

Type	参数	描述
uint32_t	intf	Interface.
uint32_t	start	Start.
uint32_t	end	End.
char	leasetime	Lease time length.

8.5.1.10 struct mcm_mobileap_dualap_config_t_v01

数据域

Type	参数	描述
uint8_t	enable	Enable dual AP.
uint32_t	a5_ip_address	A5 IP address.
uint32_t	sub_net_mask	Subnet mask.

8.5.1.11 struct mcm_mobileap_sta_static_ip_config_t_v01

数据域

Type	参数	描述
uint32_t	ip_addr	IP address.
uint32_t	gw_ip	GW IP.
uint32_t	netmask	Net mask.
uint32_t	dns_addr	DNS address.

8.5.1.12 struct mcm_mobileap_sta_connection_config_t_v01

数据域

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

9 Voice

本章主要提供与Voice模块相关的消息体、常量、枚举类型和结构体。

- [Voice Message Identifiers](#)
- [Voice Message Structures](#)
- [Voice Constants](#)
- [Voice Enumerations](#)
- [Voice Data Structures](#)

9.1 Voice 消息标识符

本节主要包含MCM voice相关的消息标识符

- #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
    
```

9.2 Voice消息结构体

本节主要包含Voice消息结构体

9.2.1 数据结构文档

9.2.1.1 struct mcm_voice_get_calls_resp_msg_v01

响应消息; 获取当前呼叫列表

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
uint8_t	calls_valid	如需使用calls, 该参数需要设置为TRUE
uint32_t	calls_len	必须设置为calls内参数的个数
mcm_voice-_call_record_t_v01	calls	Calls

9.2.1.2 struct mcm_voice_dial_req_msg_v01

请求消息; 拨打特定的号码并返回连接ID

数据域

Type	参数	描述
uint8_t	address_valid	如需使用address, 该参数需要设置为TRUE
char	address	需要拨打的号码, 即需要连接的端点地址
uint8_t	call_type_valid	如需使用call_type, 该参数需要设置为TRUE
mcm_voice-_call_type_t_v01	call_type	Connection (call) details, or NULL.

uint8_t	uusdata_valid	如需使用uusdata，该参数需要设置为TRUE
mcm_voice_-uusdata_t_v01	uusdata	Token ID used to track this command; NULL is OK.
uint8_t	emergency_cat-_valid	如需使用emergency_cat，该参数需要设置为TRUE
mcm_voice_-emergency_cat-_t_v01	emergency_cat	Emergency call category.
uint8_t	ecall_msdc-_valid	如需使用ecall_msdc，该参数需要设置为TRUE
uint32_t	ecall_msdc_len	必须设置为ecall_msdc的长度
uint8_t	ecall_msdc	Minimum set of data. 只有当call_type 是MCM_VOICE_CALL_TYPE_ECALL_AUTO或MCM_VOICE_CALL_TYPE_ECALL_MANUAL时，是有效的，不然该值被忽略。

9.2.1.3 struct mcm_voice_dial_resp_msg_v01

响应消息; 拨打特定的号码并返回连接ID

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
uint8_t	call_id_valid	如需使用call_id，该参数需要设置为TRUE
uint32_t	call_id	Call ID.

9.2.1.4 struct mcm_voice_get_call_status_req_msg_v01

请求消息; 获取到连接call ID的状态

数据域

Type	参数	描述
uint32_t	call_id	需要获取其连接状态的call ID

9.2.1.5 struct mcm_voice_get_call_status_resp_msg_v01

响应消息; 获取到连接call ID的状态

数据域

Type	参数	描述
------	----	----

mcm_response-_t_v01	response	结果码.
uint8_t	status_valid	如需使用status, 该参数需要设置为TRUE
mcm_voice_-call_record_t_v01	status	Call status.

9.2.1.6 struct mcm_voice_dtmf_req_msg_v01

请求消息; Sends a DTMF character over the connection ID.

数据域

Type	参数	描述
char	dtmf	DTMF character to be sent. Valid DTMF characters are 0-9, A-D, '*', '#'.

9.2.1.7 struct mcm_voice_dtmf_resp_msg_v01

响应消息;在连接ID上发送DTMF字符

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.

9.2.1.8 struct mcm_voice_start_dtmf_req_msg_v01

请求消息;在连接ID上开始发送DTMF字符

数据域

Type	参数	描述
uint32_t	call_id	Call ID.
char	digit	发送的DTMF字符, 有效DTMF字符如下: 0-9, A-D, '*', '#'

9.2.1.9 struct mcm_voice_start_dtmf_resp_msg_v01

响应消息; 在Call ID上开始发送DTMF字符

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
uint8_t	call_id_valid	如需使用call_id, 该参数需要设置为TRUE
uint32_t	call_id	Call ID.

9.2.1.10 struct mcm_voice_stop_dtmf_req_msg_v01

请求消息;在Call ID上停止发送DTMF字符

数据域

Type	参数	描述
uint32_t	call_id	Call ID.

9.2.1.11 struct mcm_voice_stop_dtmf_resp_msg_v01

响应消息; 在Call ID上停止发送DTMF字符

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
uint8_t	call_id_valid	如需使用call_id, 该参数需要设置为TRUE
uint32_t	call_id	Call ID.

9.2.1.12 struct mcm_voice_mute_req_msg_v01

请求消息; 使语音呼叫静音/非静音

数据域

Type	参数	描述
uint32_t	call_id	使call ID的连接静音/非静音
mcm_voice_-mute_type_t-_v01	mute_type	使语音呼叫静音/非静音

9.2.1.13 struct mcm_voice_mute_resp_msg_v01

响应消息; 使语音呼叫静音/非静音

数据域

Type	参数	描述
mcm_response-t_v01	response	结果码.

9.2.1.14 struct mcm_voice_flash_req_msg_v01

请求消息; 在一个连接的call ID上发送一个flash字符序列。

数据域

Type	参数	描述
char	flash_string	需要发送的flash字符串, 最多82个字符

9.2.1.15 struct mcm_voice_flash_resp_msg_v01

响应消息; 在一个连接的call ID上发送一个flash字符序列。

数据域

Type	参数	描述
mcm_response-t_v01	response	结果码.

9.2.1.16 struct mcm_voice_hangup_req_msg_v01

请求消息;挂断或者断开一特定call ID的语音呼叫连接。

数据域

Type	参数	描述
uint32_t	call_id	特定call ID的语音呼叫连接

9.2.1.17 struct mcm_voice_hangup_resp_msg_v01

响应消息; 挂断或者断开一特定call ID的语音呼叫连接。

数据域

Type	参数	描述
mcm_response-t_v01	response	结果码.

9.2.1.18 struct mcm_voice_command_req_msg_v01

请求消息; 为语音呼叫提供各种操作。

数据域

Type	参数	描述
mcm_voice_- call_operation- _t_v01	call_operation	Call operation.
uint8_t	call_id_valid	如需使用call_id, 该参数需要设置为TRUE
uint32_t	call_id	Call ID.
uint8_t	cause_valid	如需使用cause, 该参数需要设置为TRUE
uint32_t	cause	Cause.

9.2.1.19 struct mcm_voice_command_resp_msg_v01

响应消息; 为语音呼叫提供各种操作。

数据域

Type	参数	描述
mcm_response- _t_v01	response	结果码。

9.2.1.20 struct mcm_voice_auto_answer_req_msg_v01

请求消息; 使能/禁止语音呼入。

数据域

Type	参数	描述
mcm_voice_- auto_answer_- type_t_v01	auto_answer_- type	Auto-answer type.
uint8_t	auto_answer_- timer_valid	如需使用auto_answer_timer, 该参数需要设置为TRUE
uint32_t	auto_answer_- timer	Auto-answer timer.

9.2.1.21 struct mcm_voice_auto_answer_resp_msg_v01

响应消息; 使能/禁止语音呼入。

数据域

Type	参数	描述
mcm_response-t_v01	response	结果码.

9.2.1.22 struct mcm_voice_event_register_req_msg_v01

请求消息; 注册指示事件

数据域

Type	参数	描述
uint8_t	register_voice_call_event-valid	如需使用register_voice_call_event, 该参数需要设置为TRUE
uint8_t	register_voice_call_event	注册语音呼叫事件指示
uint8_t	register_mute_event-valid	如需使用register_mute_event, 该参数需要设置为TRUE
uint8_t	register_mute_event	注册静音事件指示
uint8_t	register_dtmf_event-valid	如需使用register_dtmf_event, 该参数需要设置为TRUE
uint8_t	register_dtmf_event	注册DTMF事件指示
uint8_t	register_e911_state_event-valid	如需使用register_e911_state_event, 该参数需要设置为TRUE
uint8_t	register_e911_state_event	MCM_VOICE_E911_STATE_INDICATION.

9.2.1.23 struct mcm_voice_event_register_resp_msg_v01

响应消息; 注册指示事件

数据域

Type	参数	描述
mcm_response-t_v01	response	结果码.

9.2.1.24 struct mcm_voice_call_ind_msg_v01

指示消息; MCM_VOICE_CONNECTION_EV的指示

数据域

Type	参数	描述
uint32_t	calls_len	必须设置为参数calls中元素的个数
mcm_voice_- call_record_t_- v01	calls	Calls.

9.2.1.25 struct mcm_voice_mute_ind_msg_v01

指示消息; MCM_VOICE_MUTE_EV的指示

数据域

Type	参数	描述
uint8_t	is_mute	指示电话是否已被静音

9.2.1.26 struct mcm_voice_dtmf_ind_msg_v01

指示消息; DTMF指示

数据域

Type	参数	描述
mcm_voice_- dtmf_info_t_- v01	dtmf_info	DTMF information

9.2.1.27 struct mcm_voice_get_call_forwarding_status_req_msg_v01

请求消息; 呼叫转移状态查询。

数据域

Type	参数	描述
mcm_- voice_call_- forwarding_- reason_t_v01	reason	呼叫转移原因

9.2.1.28 struct mcm_voice_get_call_forwarding_status_resp_msg_v01

响应消息; 呼叫转移状态查询

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
mcm_voice_call_forwarding_status_t_v01	status	呼叫转移状态
uint8_t	info_valid	如需使用info, 该参数需要设置为TRUE
uint32_t	info_len	必须设置为参数info中元素的个数
mcm_voice_call_forwarding_info_t_v01	info	呼叫转移信息

9.2.1.29 struct mcm_voice_set_call_forwarding_req_msg_v01

请求消息; 设置呼叫转移。

数据域

Type	参数	描述
mcm_voice_call_service_t_v01	fdwservice	呼叫转移服务
mcm_voice_call_forwarding_reason_t_v01	reason	呼叫转移原因
uint8_t	forwarding_number_valid	如需使用forwarding_number, 该参数需要设置为TRUE
char	forwarding_number	呼叫转移号码

9.2.1.30 struct mcm_voice_set_call_forwarding_resp_msg_v01

响应消息; 设置呼叫转移。

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.

9.2.1.31 struct mcm_voice_get_call_waiting_status_resp_msg_v01

响应消息;呼叫等待状态查询

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
mcm_voice_-call_waiting_-service_t_v01	status	呼叫等待状态

9.2.1.32 struct mcm_voice_set_call_waiting_req_msg_v01

请求消息;设置呼叫等待

数据域

Type	参数	描述
mcm_voice_-call_waiting_-service_t_v01	cwservice	呼叫等待服务

9.2.1.33 struct mcm_voice_set_call_waiting_resp_msg_v01

响应消息; 设置呼叫等待

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.

9.2.1.34 struct mcm_voice_get_clir_resp_msg_v01

响应消息; CLIR 状态查询

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.

mcm_voice_ clir_action_t_ v01	action	CLIR action.
mcm_ voice_clir_ presentation_t_ v01	presentation	CLIR presentation.

9.2.1.35 struct mcm_voice_set_clir_req_msg_v01

请求消息; 设置 CLIR.

数据域

Type	参数	描述
mcm_voice_ clir_action_t_ v01	clir_action	CLIR action.

9.2.1.36 struct mcm_voice_set_clir_resp_msg_v01

响应消息; 设置CLIR.

数据域

Type	参数	描述
mcm_response_ t_v01	response	响应

9.2.1.37 struct mcm_voice_set_facility_lock_req_msg_v01

请求消息; 设置功能锁

数据域

Type	参数	描述
mcm_voice_ facility_code_t_ v01	code	参考 3GPPTS 27.997, Section 7.4.
mcm_voice_ facility_lock_ status_t_v01	status	功能锁状态
char	password	功能锁密码

9.2.1.38 struct mcm_voice_set_facility_lock_resp_msg_v01

响应消息; 设置功能锁

数据域

Type	参数	描述
mcm_response-t_v01	response	结果码.

9.2.1.39 struct mcm_voice_change_call_barring_password_req_msg_v01

请求消息; 变更呼叫禁止密码。

数据域

Type	参数	描述
mcm_voice-change-call-barring-password-reason-t_v01	reason	密码变更原因, 参考3GPPTS 27.997, Section7.4.
char	old_password	旧密码
char	new_password	新密码

9.2.1.40 struct mcm_voice_change_call_barring_password_resp_msg_v01

响应消息; 变更呼叫禁止密码

数据域

Type	参数	描述
mcm_response-t_v01	response	结果码.

9.2.1.41 struct mcm_voice_send_usd_req_msg_v01

请求消息; 发送非结构化补充服务数据(Unstructured Supplementary Service Data, USSD)。

数据域

Type	参数	描述
------	----	----

mcm_voice- _ussd_msg_ type_t_v01	type	USSD 消息类型
mcm_voice- _ussd_encoding- _t_v01	encoding	USSD 编码
char	ussd_string	USSD 字符串

9.2.1.42 struct mcm_voice_send_ussd_resp_msg_v01

响应消息; 发送非结构化补充服务数据(USSD).

数据域

Type	参数	描述
mcm_response- _t_v01	response	结果码.

9.2.1.43 struct mcm_voice_cancel_ussd_resp_msg_v01

响应消息; 取消 USSD.

数据域

Type	参数	描述
mcm_response- _t_v01	response	结果码.

9.2.1.44 struct mcm_voice_receive_ussd_ind_msg_v01

指示消息; 收到USSD指示

数据域

Type	参数	描述
mcm_voice- _ussd_ind_ notification_t_ v01	notification	USSD指示通知
char	ussd	USSD 指示消息.

9.2.1.45 struct mcm_voice_common_dial_req_msg_v01

请求消息; Voice/SS/USSD 普通拨号API

数据域

Type	参数	描述
char	request	请求
uint8_t	call_type_valid	如需使用call_type, 该参数需要设置为TRUE
mcm_voice_-call_type_t_v01	call_type	连接(呼叫)的细节,或空
uint8_t	uusdata_valid	如需使用uusdata, 该参数需要设置为TRUE
mcm_voice_-uusdata_t_v01	uusdata	用于追踪这个命令的令牌ID, 可以为NULL
uint8_t	emergency_cat-_valid	如需使用emergency_cat, 该参数需要设置为TRUE
mcm_voice_-emergency_cat-_t_v01	emergency_cat	紧急呼叫类别
uint8_t	ecall_msd_-valid	如需使用ecall_msd, 该参数需要设置为TRUE
uint32_t	ecall_msd_len	必须设置为参数ecall_msd中元素的个数
uint8_t	ecall_msd	最小的数据集(minimum set of data, MSD),仅在call_type为MCM_VOICE_CALL_TYPE_ECALL_AUTO或MCM_VOICE_CALL_TYPE_ECALL_MANUAL时有意义, 其他情况下参数被忽略

9.2.1.46 struct mcm_voice_common_dial_resp_msg_v01

响应消息; Voice/SS/USSD普通拨号API

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
uint8_t	req_changed_-to_type_valid	如需使用 req_changed_to_type , 该参数需要设置为TRUE
mcm_voice_-common_dial_-type_t_v01	req_changed_-to_type	如果SS,检查可选的SS字段。如果无, 拨出语音呼叫
uint8_t	call_id_valid	如需使用 call_id , 该参数需要设置为TRUE
uint32_t	call_id	Call ID
uint8_t	ss_get_cf_-status_valid	如需使用 ss_get_cf_status , 该参数需要设置为TRUE

mcm_voice_call_forwarding_status_t_v01	ss_get_cf_status	获取呼叫转移状态
uint8_t	ss_get_cf_info_valid	如需使用 ss_get_cf_info，该参数需要设置为TRUE
uint32_t	ss_get_cf_info_len	必须设置为参数 ss_get_cf_info的数据元素个数
mcm_voice_call_forwarding_info_t_v01	ss_get_cf_info	呼叫转移信息
uint8_t	ss_get_cw_status_valid	如需使用 ss_get_cw_status，该参数需要设置为TRUE
mcm_voice_call_waiting_service_t_v01	ss_get_cw_status	呼叫等待状态
uint8_t	ss_get_clir_action_valid	如需使用 ss_get_clir_action，该参数需要设置为TRUE
mcm_voice_clir_action_t_v01	ss_get_clir_action	CLIR action
uint8_t	ss_get_clir_presentation_valid	如需使用 ss_get_clir_presentation，该参数需要设置为TRUE
mcm_voice_clir_presentation_t_v01	ss_get_clir_presentation	CLIR presentation

9.2.1.47 struct mcm_voice_update_msd_req_msg_v01

请求消息;为一个正在进行的或后续的eCall呼叫更新最小数据集(minimum set of dataMSD)

数据域

Type	参数	描述
uint8_t	ecall_msd_valid	如需使用ecall_msd，该参数需要设置为TRUE
uint32_t	ecall_msd_len	必须设置为参数ecall_msd的数据元素个数
uint8_t	ecall_msd	最小数据集，ASN.1 PER unaligned格式。仅在enable_msd设置为TRUE时，该参数才有意义

9.2.1.48 struct mcm_voice_update_msd_resp_msg_v01

响应消息; 为一个正在进行的或后续的eCall呼叫更新最小数据集(minimum set of dataMSD)

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.

9.2.1.49 struct mcm_voice_e911_state_ind_msg_v01

指示消息; MCM_VOICE_E911_STATE_IND指示

数据域

Type	参数	描述
mcm_voice_-e911_state_t_-v01	e911_state	E911 state.

9.2.1.50 struct mcm_voice_get_e911_state_resp_msg_v01

响应消息; MCM_VOICE_GET_E911_STATE指示

数据域

Type	参数	描述
mcm_response-_t_v01	response	结果码.
uint8_t	e911_state_-valid	如需使用e911_state, 该参数需要设置为TRUE
mcm_voice_-e911_state_t_-v01	e911_state	E911 state.

9.3 Voice Constants

本节包含MCM VOICE的常量

9.3.1 宏定义集

9.3.1.1 #define MCM_MAX_VOICE_CALLS_V01 8

GSM最多提供8 个calls, 3GPP2最多提供2个

9.3.1.2 **#define MCM_MAX_PHONE_NUMBER_V01 82**

Maximum length for a phone number or SIP URI (81 + NULL).

9.3.1.3 **#define MCM_MAX_UUS_DATA_V01 20**

Maximum user-to-user data

9.3.1.4 **#define MCM_MAX_DTMF_LENGTH_V01 20**

Maximum DTMF length

9.3.1.5 **#define MCM_MAX_USSD_LENGTH_V01 128**

Maximum USSD length

9.3.1.6 **#define MCM_MAX_PASSWORD_LENGTH_V01 4**

Maximum password length

9.3.1.7 **#define MCM_MAX_CALL_FORWARDING_INFO_V01 13**

Maximum call forwarding information

9.3.1.8 **#define MCM_MAX_ECALL_MSD_V01 140**

Maximum size of the MSD sent to the network with an eCall.

9.4 **Voice相关枚举**

本节包含MCM voice相关枚举值

9.4.1 **枚举类型集**

9.4.1.1 **enum mcm_voice_call_state_t_v01**

成员:

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 alerting.

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.

9.4.1.2 enum mcm_voice_call_type_t_v01

成员:

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.

9.4.1.3 enum mcm_voice_call_direction_type_t_v01

成员:

MCM_VOICE_CALL_MOBILE_ORIGINATED_V01 Mobile-originated.
MCM_VOICE_CALL_MOBILE_TERMINATED_V01 Mobile-terminated.

9.4.1.4 enum mcm_voice_call_number_presentation_type_t_v01

成员:

MCM_VOICE_CALL_NUMBER_ALLOWED_V01 Number allowed.
MCM_VOICE_CALL_NUMBER_RESTRICTED_V01 Number restricted.
MCM_VOICE_CALL_NUMBER_PAYPHONE_V01 Payphone number.

9.4.1.5 enum mcm_voice_reason_t_v01

成员:

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.

9.4.1.6 enum mcm_voice_call_operation_t_v01

成员:

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.

9.4.1.7 enum mcm_voice_uus_type_t_v01

成员:

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.

9.4.1.8 enum mcm_voice_uus_dcs_type_t_v01

成员:

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.

9.4.1.9 enum mcm_voice_tech_t_v01

成员:

MCM_VOICE_TECH_NONE_V01 None.
MCM_VOICE_TECH_3GPP_V01 3GPP.
MCM_VOICE_TECH_3GPP2_V01 3GPP2.

9.4.1.10 enum mcm_voice_call_forwarding_status_t_v01

成员:

MCM_VOICE_CALL_FORWARDING_DISABLED_V01 Disabled.
MCM_VOICE_CALL_FORWARDING_ENABLED_V01 Enabled.

9.4.1.11 enum mcm_voice_call_forwarding_type_t_v01

成员:

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.

9.4.1.12 enum mcm_voice_call_waiting_service_t_v01

成员:

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.

9.4.1.13 enum mcm_voice_call_service_t_v01

成员:

MCM_VOICE_SERVICE_ACTIVATE_V01 Activate.
MCM_VOICE_SERVICE_DEACTIVATE_V01 Deactivate.
MCM_VOICE_SERVICE_REGISTER_V01 Register.
MCM_VOICE_SERVICE_ERASE_V01 Erase.

9.4.1.14 enum mcm_voice_call_forwarding_reason_t_v01

成员:

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.

9.4.1.15 enum mcm_voice_clir_action_t_v01

成员:

MCM_VOICE_CLIR_INVOCATION_V01 Invocation.
MCM_VOICE_CLIR_SUPPRESSION_V01 Suppression.

9.4.1.16 enum mcm_voice_clir_presentation_t_v01

成员:

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.

9.4.1.17 enum mcm_voice_facility_lock_status_t_v01

成员:

MCM_VOICE_FACILITY_LOCK_ENABLE_V01 Enable.

MCM_VOICE_FACILITY_LOCK_DISABLE_V01 Disable.

9.4.1.18 enum mcm_voice_facility_code_t_v01

成员:

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

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

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

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

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

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

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

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

9.4.1.19 enum mcm_voice_change_call_barring_password_reason_t_v01

成员:

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_OUTGOINGINTEXTOHOMES_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.

9.4.1.20 enum mcm_voice_ussd_encoding_t_v01

成员:

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.

9.4.1.21 enum mcm_voice_common_dial_type_t_v01

成员:

MCM_VOICE_COMMON_DIAL_VOICE_V01 Voice.

MCM_VOICE_COMMON_DIAL_SS_V01 Supplementary service.

MCM_VOICE_COMMON_DIAL_USSD_V01 Unstructured supplementary service.

9.4.1.22 enum mcm_voice_mute_type_t_v01

成员:

MCM_VOICE_MUTE_V01 Mute.

MCM_VOICE_UNMUTE_V01 Unmute.

9.4.1.23 enum mcm_voice_auto_answer_type_t_v01

成员:

MCM_VOICE_AUTO_ANSWER_ENABLE_V01 Enable auto-answer.

MCM_VOICE_AUTO_ANSWER_DISABLE_V01 Disable auto-answer.

9.4.1.24 enum mcm_voice_dtmf_event_type_t_v01

成员:

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.

9.4.1.25 enum mcm_voice_ussd_msg_type_t_v01

成员:

MCM_VOICE_USSD_MSG_TYPE_NEW_MESSAGE_V01 Initiate a new USSD session with the network.

MCM_VOICE_USSD_MSG_TYPE_REPLY_TO_IND_V01 Reply to a USSD indication from the network.

9.4.1.26 enum mcm_voice_ussd_ind_notification_t_v01

成员:

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.

9.4.1.27 enum mcm_voice_emergency_cat_t_v01

成员:

MCM_VOICE_EMER_CAT_POLICE_V01 Police.

MCM_VOICE_EMER_CAT_AMBULANCE_V01 Ambulance.

MCM_VOICE_EMER_CAT_FIRE_BRIGADE_V01 Fire brigade.

MCM_VOICE_EMER_CAT_MARINE_GUARD_V01 Marine guard.

MCM_VOICE_EMER_CAT_MOUNTAIN_RESCUE_V01 Mountain rescue.

9.4.1.28 enum mcm_voice_e911_state_t_v01

成员:

MCM_VOICE_E911_INACTIVE_V01 E911 INACTIVE.

MCM_VOICE_E911_ACTIVE_V01 E911 ACTIVE.

9.5 Voice相关数据结构

本节主要包含Voice相关数据结构

9.5.1 数据结构文档

9.5.1.1 struct mcm_voice_uusdata_t_v01

数据域

Type	参数	描述
mcm_voice_uus_type_t_v01	type	UUS 类型;范围: 0 to 6.
mcm_voice_uus_dcs_type_t_v01	dcs	UUS数据编码方案; 范围: 0 to 4.
uint32_t	uus_data_len	必须设置为参数uus_data元素个数
uint8_t	uus_data	语音呼叫 UUS 数据

9.5.1.2 struct mcm_voice_call_record_t_v01

数据域

Type	参数	描述
uint32_t	call_id	该呼叫相关的call ID
mcm_voice_call_state_t_v01	state	当前呼叫状态(mcm_voice_call_state).
mcm_voice_tech_t_v01	tech	技术(mcm_tech).
char	number	电话号码
mcm_voice_call_number_presentation_type_t_v01	number_presentation	语音呼叫号码描述类型
mcm_voice_call_direction_type_t_v01	direction	语音呼叫方向：来电、去电
uint8_t	uusdata_valid	表明UUS 数据是否有效
mcm_voice_uusdata_t_v01	uusdata	UUS数据（User-to-user.signaling）

9.5.1.3 struct mcm_voice_dtmf_info_t_v01
数据域

Type	参数	描述
uint32_t	call_id	与该DTMF事件相关的Call ID
mcm_voice_dtmf_event_type_t_v01	dtmf_event	DTMF事件类型
uint32_t	digit_len	必须设置为参数digit元素个数
char	digit	DTMF 字符

9.5.1.4 struct mcm_voice_call_forwarding_info_t_v01
数据域

Type	参数	描述
mcm_voice_call_forwarding_type_t_v01	type	呼叫转移类型
char	number	呼叫转移号码

10 SIM(Subscriber Identity Module)

该章节提供SIM相关的消息体，常量，枚举类型及数据结构

- [SIM Message Identifiers](#)
- [SIM Message Structures](#)
- [SIM Constants](#)
- [SIM Enumerations](#)
- [SIM Data Structures](#)

10.1 SIM消息标识符

该节包含MCM SIM相关标识符

```
#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
    
```

10.2 SIM消息结构体

该节包含MCM SIM相关消息结构体

10.2.1 数据结构文档

10.2.1.1 struct mcm_sim_get_subscriber_id_req_msg_v01

请求消息;检索存储在特定的应用程序的国际移动用户身份(IMSIs)。

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息

10.2.1.2 struct mcm_sim_get_subscriber_id_resp_msg_v01

响应消息;检索存储在特定的应用程序的IMSI

数据域

Type	参数	描述
mcm_response-_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t –错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	imsi_valid	如需使用imsi，该参数需要设置为TRUE
uint32_t	imsi_len	必须设置为参数imsi的长度
char	imsi	ASCII字符格式的IMSI

10.2.1.3 struct mcm_sim_get_card_id_req_msg_v01

请求消息; 检索存储在卡上的ICCID(Integrated Circuit Card ID)。

数据域

Type	参数	描述
mcm_sim_slot-_id_t_v01	slot_id	表明使用的插槽，值如下： 1 – Slot 1 2 – Slot 2

10.2.1.4 struct mcm_sim_get_card_id_resp_msg_v01

响应消息; 检索存储在卡上的ICCID(Integrated Circuit Card ID)。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t –错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	iccid_valid	如需使用iccid，该参数需要设置为TRUE
uint32_t	iccid_len	必须设置为参数iccid的长度
char	iccid	ASCII字符格式的ICCID

10.2.1.5 struct mcm_sim_get_device_phone_number_req_msg_v01

请求消息; 检索存储在卡上的设备电话号码。

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息

10.2.1.6 struct mcm_sim_get_device_phone_number_resp_msg_v01

响应消息; 检索存储在卡上的设备电话号码。

数据域

Type	参数	描述
mcm_response_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t - MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t - 错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	phone_number_valid	如需使用phone_number，该参数需要设置为TRUE
uint32_t	phone_number_len	必须设置为参数phone_number的长度
char	phone_number	ASCII字符格式的解析了的电话号码

10.2.1.7 struct mcm_sim_get_preferred_operator_list_req_msg_v01

请求消息; 检索存储在卡上的首选运营商列表

注意：该命令仅在3GPP应用时支持

数据域

Type	参数	描述
mcm_sim_slot_id_t_v01	slot_id	表明使用的插槽，值如下： 1 – Slot 1 2 – Slot 2

10.2.1.8 struct mcm_sim_get_preferred_operator_list_resp_msg_v01

响应消息; 检索存储在卡上的首选运营商列表

注意: 该命令仅在3GPP应用时支持

数据域

Type	参数	描述
mcm_response-_t_v01	resp	标准响应类型, 包含如下数据成员: ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码, 可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	preferred_-operator_list-_valid	如需使用preferred_operator_list, 该参数需要设置为TRUE
uint32_t	preferred_-operator_list-_len	必须设置为参数preferred_operator_list中元素的个数
mcm_sim_-plmn_t_v01	preferred_-operator_list	首选运营商列表

10.2.1.9 struct mcm_sim_read_file_req_msg_v01

请求消息; 将数据读取到卡上指定应用程序的特定文件。该类型文件由需要记录数据字段决定, 当该字段为0时代表为透明文件, 当为其他值时代表为记录的文件

数据域

Type	参数	描述
mcm_sim_-application_-identification_-info_t_v01	app_info	应用识别信息
mcm_sim_file-_access_t_v01	file_access	文件访问信息

10.2.1.10 struct mcm_sim_read_file_resp_msg_v01

响应消息; 将数据读取到卡上指定应用程序的特定文件。响应包含收到的卡(SW1 和 SW2) 响应读请求时的状态码, 以及从文件里读取的数据

数据域

Type	参数	描述
------	----	----

mcm_response_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	card_result_valid	如需使用card_result，该参数需要设置为TRUE
mcm_sim_card_result_t_v01	card_result	Card result
uint8_t	data_valid	如需使用data，该参数需要设置为TRUE
uint32_t	data_len	必须设置为data的数据长度
uint8_t	data	从卡中检索数据

10.2.1.11 struct mcm_sim_write_file_req_msg_v01

请求消息;将数据读写到卡上指定应用程序的特定文件。该类型文件由需要记录数据字段决定，当该字段为0时代表为透明文件，当为其他值时代表为记录的文件

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息
mcm_sim_file_access_t_v01	file_access	文件访问信息
uint32_t	data_len	必须设置为data的数据长度
uint8_t	data	需要在卡上更新的数据

10.2.1.12 struct mcm_sim_write_file_resp_msg_v01

响应消息;将数据读写到卡上指定应用程序的特定文件。响应包含收到的卡(SW1 和 SW2) 响应写请求时的状态码

数据域

Type	参数	描述
mcm_response_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码，可能的错误代码值描述在每个消息的错误代码部分定义。

uint8_t	card_result_valid	如需使用card_result，该参数需要设置为TRUE
mcm_sim_card_result_t_v01	card_result	Card result.

10.2.1.13 struct mcm_sim_get_file_size_req_msg_v01

请求消息;检索卡上指定应用程序的特定文件。

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息
uint32_t	path_len	必须设置为参数path的长度
char	path	ASCII字符的文件路径

10.2.1.14 struct mcm_sim_get_file_size_resp_msg_v01

响应消息; 检索卡上指定应用程序的特定文件的大小。响应包含收到的卡(SW1 和 SW2) 响应获取指定文件大小请求时的状态码，以及与大小相关的文件类型

数据域

Type	参数	描述
mcm_response_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t –错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	card_result_valid	如需使用card_result，该参数需要设置为TRUE
mcm_sim_card_result_t_v01	card_result	Card result
uint8_t	file_info_valid	如需使用file_info，该参数需要设置为TRUE
mcm_sim_file_info_t_v01	file_info	File information

10.2.1.15 struct mcm_sim_verify_pin_req_msg_v01

请求消息; 验证应用的PIN码。同一个PIN码可被多种会话使用（如：PIN码在ICC卡上会被GSM和RUIM共享）。当命名被执行时，在所有会话中，PIN码会被自动验证

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息
mcm_sim_pin_id_t_v01	pin_id	PIN ID
uint32_t	pin_value_len	必须设置为参数pin_value的长度
char	pin_value	PIN value

10.2.1.16 struct mcm_sim_verify_pin_resp_msg_v01

响应消息; 验证应用的PIN码。

数据域

Type	参数	描述
mcm_response_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	retries_left_valid	如需使用retries_left，该参数需要设置为TRUE
uint8_t	retries_left	剩余的重试次数

10.2.1.17 struct mcm_sim_change_pin_req_msg_v01

请求消息; 改变应用的PIN码。该应用必须传递新旧PIN码来完成操作。

同一个PIN码可被多个会话使用（如：PIN码在ICC卡上会被GSM和RUIM共享）。当命名被执行时，在所有会话中，PIN码会被自动验证

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息

mcm_sim_pin-_id_t_v01	pin_id	PIN ID
uint32_t	old_pin_value-_len	必须设置为参数old_pin_value的长度
char	old_pin_value	ASCII字符型的旧PIN码
uint32_t	new_pin_value-_len	必须设置为参数new_pin_value的长度
char	new_pin_value	ASCII字符型的新PIN码

10.2.1.18 struct mcm_sim_change_pin_resp_msg_v01

响应消息; 改变应用的PIN码。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	标准响应类型, 包含如下数据成员: ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码, 可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	retries_left_-_valid	如需使用retries_left, 该参数需要设置为TRUE
uint8_t	retries_left	剩余的重试次数

10.2.1.19 struct mcm_sim_unblock_pin_req_msg_v01

请求消息; 用PUK码解锁锁定的PIN。客户端必须传递PUK1解锁PIN1, 或者PUK2解锁PIN2。

同一个PIN码可被多个会话使用（如: PIN码在ICC卡上会被GSM和RUIM共享）。当命名被执行时, 在所有会话中, PIN码会被自动验证

数据域

Type	参数	描述
mcm_sim_-_application_-_identification_-_info_t_v01	app_info	应用识别信息
mcm_sim_pin-_id_t_v01	pin_id	PIN ID.
uint32_t	puk_value_len	必须设置为参数puk_value的长度
char	puk_value	ASCII字符型的PUK值
uint32_t	new_pin_value-_len	必须设置为参数new_pin_value的长度

char	new_pin_value	ASCII字符型的新PIN码
------	---------------	----------------

10.2.1.20 struct mcm_sim_unblock_pin_resp_msg_v01

响应消息;用PUK码解锁锁定的PIN

数据域

Type	参数	描述
mcm_response_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t –错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	retries_left_valid	如需使用retries_left，该参数需要设置为TRUE
uint8_t	retries_left	剩余的重试次数

10.2.1.21 struct mcm_sim_enable_pin_req_msg_v01

请求消息;使能PIN

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息
mcm_sim_pin_id_t_v01	pin_id	PIN ID.
uint32_t	pin_value_len	必须设置为参数pin_value的长度
char	pin_value	ASCII字符型的PIN码

10.2.1.22 struct mcm_sim_enable_pin_resp_msg_v01

响应消息; 使能PIN

数据域

Type	参数	描述
------	----	----

mcm_response-_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	retries_left_valid	如需使用retries_left，该参数需要设置为TRUE
uint8_t	retries_left	剩余的重试次数

10.2.1.23 struct mcm_sim_disable_pin_req_msg_v01

请求消息;使能/禁止PIN码应用

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息
mcm_sim_pin_id_t_v01	pin_id	PIN ID.
uint32_t	pin_value_len	必须设置为参数pin_value的长度
char	pin_value	ASCII字符型的PIN码

10.2.1.24 struct mcm_sim_disable_pin_resp_msg_v01

响应消息;使能/禁止PIN码应用

数据域

Type	参数	描述
mcm_response-_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码，可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	retries_left_valid	如需使用retries_left，该参数需要设置为TRUE
uint8_t	retries_left	剩余的重试次数

10.2.1.25 struct mcm_sim_depersonalization_req_msg_v01

请求消息; 开启/关闭电话的个性化定制。每个特性与其他特性均能相互独立的开启/关闭

数据域

Type	参数	描述
mcm_sim_depersonalization_t_v01	depersonalization	去个性化

10.2.1.26 struct mcm_sim_depersonalization_resp_msg_v01

响应消息; 开启/关闭电话的个性化定制。

数据域

Type	参数	描述
mcm_response_t_v01	resp	标准响应类型, 包含如下数据成员: ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码, 可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	retries_left_valid	如需使用retries_left, 该参数需要设置为TRUE
mcm_sim_perso_retries_left_t_v01	retries_left	剩余的重试次数

10.2.1.27 struct mcm_sim_personalization_req_msg_v01

请求消息; 激活并设置电话的个性化数据。每个特性与其他特性均能相互独立, 但是为了防止相互冲突的个性化, 激活个性化的网络数据配置必须保持一致, 且每条消息只能激活一个特性。

如果个性化已经激活, 如果需要新的数据配置重新激活, 需要首先使之前的个性化设置无效 (去激活)。

数据域

Type	参数	描述
uint32_t	ck_value_len	必须设置为参数ck_value的长度
char	ck_value	Control key value. 该值为ASCII字符形式
uint8_t	feature_gw_network_perso_valid	如需使用feature_gw_network_perso, 该参数需要设置为TRUE

uint32_t	feature_gw_- network_perso- _len	必须设置为参数 feature_gw_network_perso 的长度
mcm_sim_- network_perso- _t_v01	feature_gw_- network_perso	GW 网络个性化
uint8_t	feature_gw_- network_- subset_perso_- valid	如需使用 feature_gw_network_subset_perso，该参数需要设置为TRUE
uint32_t	feature_gw_- network_- subset_perso_- len	必须设置为参数 feature_gw_network_subset_perso 的长度
mcm_sim_- gw_network_- subset_perso_t- _v01	feature_gw_- network_- subset_perso	GW 网络子集个性化
uint8_t	feature_gw_sp- _perso_valid	如需使用 feature_gw_sp_perso，该参数需要设置为TRUE
uint32_t	feature_gw_sp- _perso_len	必须设置为参数 feature_gw_sp_perso 的长度
mcm_sim_gw- _sp_perso_t_- v01	feature_gw_sp- _perso	GW 服务提供商个性化
uint8_t	feature_gw_- corporate_- perso_valid	如需使用 feature_gw_corporate_perso，该参数需要设置为TRUE
uint32_t	feature_gw_- corporate_- perso_len	必须设置为参数 feature_gw_corporate_perso 的长度
mcm_sim_gw- _corporate_- perso_t_v01	feature_gw_- corporate_perso	GW corporate 个性化
uint8_t	feature_gw_- sim_perso_- valid	如需使用 feature_gw_sim_perso，该参数需要设置为TRUE
uint32_t	feature_gw_- sim_perso_len	必须设置为参数 feature_gw_sim_perso 的长度
mcm_sim_sim- _perso_t_v01	feature_gw_- sim_perso	GW SIM 个性化
uint8_t	feature_1x_- network1_- perso_valid	如需使用 feature_1x_network1_perso，该参数需要设置为TRUE
uint32_t	feature_1x_- network1_- perso_len	必须设置为参数 feature_1x_network1_perso 的长度

mcm_sim_- network_perso- _t_v01	feature_1x_- network1_perso	1X network type 1 personalization.
uint8_t	feature_1x_- network2_- perso_valid	如需使用feature_1x_network2_perso, 该参数需要设置为TRUE
uint32_t	feature_1x_- network2_- perso_len	必须设置为参数 feature_1x_network2_perso的长度
mcm_sim_1x_- network_type2- _perso_t_v01	feature_1x_- network2_perso	1X network type 3 personalization.
uint8_t	feature_1x_- ruim_perso_- valid	如需使用feature_1x_ruim_perso, 该参数需要设置为TRUE
uint32_t	feature_1x_- ruim_perso_len	必须设置为参数feature_1x_ruim_perso的长度
mcm_sim_sim- _perso_t_v01	feature_1x_- ruim_perso	1X RUIM personalization.

10.2.1.28 struct mcm_sim_personalization_resp_msg_v01

响应消息; 激活并设置电话的个性化数据

数据域

Type	参数	描述
mcm_response- _t_v01	resp	标准响应类型, 包含如下数据成员: - mcm_result_t – MCM_RESULT_SUCCESS或 MCM_RESULT_FAILURE - mcm_error_t – 错误码, 可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	retries_left_- valid	如需使用retries_left, 该参数需要设置为TRUE
mcm_sim_- perso_retries_- left_t_v01	retries_left	该值仅在激活和设置个性化失败的时候返回

10.2.1.29 struct mcm_sim_get_card_status_req_msg_v01

请求消息; 检索存储在卡上的卡状态

数据域

Type	参数	描述
------	----	----

mcm_sim_slot-_id_t_v01	slot_id	Slot ID.
------------------------	---------	----------

10.2.1.30 struct mcm_sim_get_card_status_resp_msg_v01

响应消息; 检索存储在卡上的卡状态, 卡状态结果可被客户端用来判断特定目标支持的插槽数。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	标准响应类型, 包含如下数据成员: ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码, 可能的错误代码值描述在每个消息的错误代码部分定义。
uint8_t	card_info_valid	如需使用card_info, 该参数需要设置为TRUE
mcm_sim-_card_info_t_v01	card_info	Card information

10.2.1.31 struct mcm_sim_event_register_req_msg_v01

请求消息; 注册/去注册一个主动上报的SIM事件。

数据域

Type	参数	描述
uint8_t	register_card-_status_event-_valid	如需使用register_card_status_event, 该参数需要设置为TRUE
uint8_t	register_card-_status_event	注册SIM卡状态事件
uint8_t	register-_refresh_event-_valid	如需使用register_refresh_event, 该参数需要设置为TRUE
uint8_t	register-_refresh_event	注册更新事件

10.2.1.32 struct mcm_sim_event_register_resp_msg_v01

响应消息; 注册/去注册主动上报SIM卡事件。只有在请求会话类型所属的文件被更新进程修改时, 客户端才能收到通知。客户端不需要详细指定文档列表。

客户端能够通过一个FALSE的布尔值来去注册卡状态或刷新事件。

数据域

Type	参数	描述
mcm_response_t_v01	resp	标准响应类型，包含如下数据成员： ● mcm_result_t – MCM_RESULT_SUCCESS或MCM_RESULT_FAILURE ● mcm_error_t – 错误码，可能的错误代码值描述在每个消息的错误代码部分定义。

10.2.1.33 struct mcm_sim_card_status_event_ind_msg_v01

指示消息; 与MCM_SIM_CARD_STATUS_EVENT_IND对应的指示

数据域

Type	参数	描述
uint8_t	slot_id_valid	如需使用slot_id，该参数需要设置为TRUE
mcm_sim_slot_id_t_v01	slot_id	卡对应的标识
uint8_t	card_info_valid	如需使用card_info，该参数需要设置为TRUE
mcm_sim_card_info_t_v01	card_info	卡信息显示

10.2.1.34 struct mcm_sim_refresh_event_ind_msg_v01

指示消息; MCM_SIM_REFRESH_EVENT_IND对应的指示

数据域

Type	参数	描述
uint8_t	refresh_event_valid	如需使用refresh_event，该参数需要设置为TRUE
mcm_sim_refresh_event_t_v01	refresh_event	刷新指示信息

10.3 SIM常量

该节包含MCM SIM相关常量

10.3.1 宏定义集

10.3.1.1 #define MCM_SIM_IMSI_LEN_V01 16

IMSI的最大长度

10.3.1.2 #define MCM_SIM_ICCID_LEN_V01 20

ICCID 的最大长度

10.3.1.3 #define MCM_SIM_NUM_PLMN_MAX_V01 24

PLMN的最大长度

10.3.1.4 #define MCM_SIM_CHAR_PATH_MAX_V01 20

ASCII格式的文件全路径的最大长度

10.3.1.5 #define MCM_SIM_DATA_MAX_V01 4096

需要读/写的数据的最大长度

10.3.1.6 #define MCM_SIM_PIN_MAX_V01 8

PIN的最大长度

10.3.1.7 #define MCM_SIM_MAX_NUM_CARDS_V01 2

卡的最大数量

10.3.1.8 #define MCM_SIM_MAX_REFRESH_FILES_V01 35

需刷新文件的最大数目

10.3.1.9 #define MCM_SIM_MAX_BINARY_PATHS_V01 10

二进制格式的文件全路径的最大长度

10.3.1.10 #define MCM_SIM_CK_MAX_V01 16

个性化控制关键（control key）最大长度

10.3.1.11 #define MCM_SIM_MCC_LEN_V01 3

MCC的长度

10.3.1.12 #define MCM_SIM_MNC_MAX_V01 3

MNC的长度

10.3.1.13 #define MCM_SIM_PHONE_NUMBER_MAX_V01 82

电话号码的最大长度

10.3.1.14 #define MCM_SIM_IRM_CODE_LEN_V01 4

IRM码的长度

10.3.1.15 #define MCM_SIM_MSIN_MAX_V01 10

MSIN的最大长度

10.3.1.16 #define MCM_SIM_PERSO_NUM_NW_MAX_V01 85

网络个性化数据集的最大数目

10.3.1.17 #define MCM_SIM_PERSO_NUM_NS_MAX_V01 64

网络子集个性化数据集的最大数目

10.3.1.18 #define MCM_SIM_PERSO_NUM_GW_SP_MAX_V01 64

服务供应商个性化数据集的最大数目

10.3.1.19 #define MCM_SIM_PERSO_NUM_GW_CP_MAX_V01 51

企业个性化数据集的最大数目

10.3.1.20 #define MCM_SIM_PERSO_NUM_SIM_MAX_V01 32

SIM个性化数据集的最大数目

10.3.1.21 #define MCM_SIM_PERSO_NUM_1X_NW2_MAX_V01 128

网络类型2个性化数据集的最大数目

10.4 SIM枚举类型

该节包含MCM SIM相关枚举类型

10.4.1 枚举类型集

10.4.1.1 enum mcm_sim_slot_id_t_v01

成员:

MCM_SIM_SLOT_ID_1_V01 Identify card in slot 1.

MCM_SIM_SLOT_ID_2_V01 Identify card in slot 2.

10.4.1.2 enum mcm_sim_app_type_t_v01

成员:

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.

10.4.1.3 enum mcm_sim_file_type_t_v01

成员:

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.

10.4.1.4 enum mcm_sim_pin_id_t_v01

成员:

MCM_SIM_PIN_ID_1_V01 Level 1 user verification.

MCM_SIM_PIN_ID_2_V01 Level 2 user verification.

10.4.1.5 enum mcm_sim_perso_feature_t_v01

成员:

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.

10.4.1.6 enum mcm_sim_perso_operation_t_v01

成员:

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.

10.4.1.7 enum mcm_sim_card_t_v01

成员:

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.

10.4.1.8 enum mcm_sim_subscription_t_v01

成员:

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.

10.4.1.9 enum mcm_sim_app_state_t_v01

成员:

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_INITIALIZING_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.

10.4.1.10 enum mcm_sim_pin_state_t_v01

成员:

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.

10.4.1.11 enum mcm_sim_card_state_t_v01

成员:

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 Card returned technical problems.

10.4.1.12 enum mcm_sim_refresh_mode_t_v01

成员:

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 initialization and file change notification.
MCM_SIM_REFRESH_NAA_INIT_FULL_FCN_V01 Refresh NAA initialization 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.

10.5 SIM数据结构

该节包含MCM SIM相关数据结构

10.5.1 数据结构文档

10.5.1.1 struct mcm_sim_application_identification_info_t_v01

数据域

Type	参数	描述
mcm_sim_slot_id_t_v01	slot_id	表示使用的插槽，值为： ● 1 – Slot 1 ● 2 – Slot 2
mcm_sim_app_type_t_v01	app_t	表明应用类型，值为： ● 0 – Unknown ● 1 – 3GPP application ● 2 – 3GPP2 application ● 3 – ISIM application 其他值保留

10.5.1.2 struct mcm_sim_plmn_t_v01

数据域

Type	参数	描述
char	mcc	MCC 的 ASCII 字符值
uint32_t	mnc_len	必须设置为 MNC 的长度
char	mnc	MNC 的 ASCII 字符值

10.5.1.3 struct mcm_sim_file_access_t_v01

数据域

Type	参数	描述
uint16_t	offset	当指明文件入口的数据长度后，仅需要 offset 偏移参数
uint8_t	record_num	文件入口相关的记录数。记录数据 0 表示透明文件入口
uint32_t	path_len	必须设置为 参数 path 的长度
char	path	文件路径的 ASCII 字符值

10.5.1.4 struct mcm_sim_card_result_t_v01

数据域

Type	参数	描述
uint8_t	sw1	从卡接收到的 SW1

uint8_t	sw2	从卡接收到的SW2
---------	-----	-----------

10.5.1.5 struct mcm_sim_file_info_t_v01

数据域

Type	参数	描述
mcm_sim_file-_type_t_v01	file_t	文件类型: ● 0xB00 – Unknown ● 0xB01 – Transparent ● 0xB02 – Cyclic ● 0xB03 – Linear fixed
uint16_t	file_size	透明文件的长度
uint16_t	record_size	每个线性或者循环的可配置文件记录尺寸。
uint16_t	record_count	线性或者循环的可配置文件记录数量

10.5.1.6 struct mcm_sim_depersonalization_t_v01

数据域

Type	参数	描述
mcm_sim-_perso_feature-_t_v01	feature	表明开启或去激活的个性化特性，有效值为: ● 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	表明将要执行的操作，值为: ● 0 – Deactivate personalization ● 1 – Unblock personalization
uint32_t	ck_value_len	必须设置为参数ck_value的长度
char	ck_value	ASCII字符型Control key 值

10.5.1.7 struct mcm_sim_network_perso_t_v01

数据域

Type	参数	描述
char	mcc	ASCII字符型的MCC
uint32_t	mnc_len	必须设置为参数MNC的长度
char	mnc	ASCII字符型的MNC

10.5.1.8 struct mcm_sim_gw_network_subset_perso_t_v01

数据域

Type	参数	描述
mcm_sim_- network_perso- _t_v01	network	MCC和MNC网络信息
char	digit6	Digit 6 of the IMSI in ASCII character
char	digit7	Digit 7 of the IMSI in ASCII character

10.5.1.9 struct mcm_sim_gw_sp_perso_t_v01

数据域

Type	参数	描述
mcm_sim_- network_perso- _t_v01	network	MCC和MNC网络信息
uint8_t	gid1	GID1中的服务提供商代码

10.5.1.10 struct mcm_sim_gw_corporate_perso_t_v01

数据域

Type	参数	描述
mcm_sim_- network_perso- _t_v01	network	MCC和MNC网络信息
uint8_t	gid1	GID1中的服务提供商代码
uint8_t	gid2	GID2中的服务提供商代码

10.5.1.11 struct mcm_sim_sim_perso_t_v01

数据域

Type	参数	描述
mcm_sim_- network_perso- _t_v01	network	MCC和MNC网络信息
uint32_t	msin_len	必须设置为参数MSIN的长度

char	msin	卡上存储的ASCII字符型的MSIN值
------	------	---------------------

10.5.1.12 struct mcm_sim_1x_network_type2_perso_t_v01

数据域

Type	参数	描述
char	irm_code	ASCII字符类型的IRM-based MIN of IMSI_M的首四位

10.5.1.13 struct mcm_sim_perso_retries_left_t_v01

数据域

Type	参数	描述
uint8_t	verify_left	验证个性化的剩余可尝试次数
uint8_t	unlock_left	开启个性化的剩余可尝试次数

10.5.1.14 struct mcm_sim_app_info_t_v01

数据域

Type	参数	描述
mcm_sim_subscription_t_v01	subscription	订阅类型（如：primary, secondary等）
mcm_sim_app_state_t_v01	app_state	应用的当前状态
mcm_sim_perso_feature_t_v01	perso_feature	当前个性化状态机特性使能
uint8_t	perso_retries	个性化重试次数
uint8_t	perso_unlock_retries	个性化开启重试次数
mcm_sim_pin_state_t_v01	pin1_state	当前PIN1状态
uint8_t	pin1_num_retries	PIN 1重试次数
uint8_t	puk1_num_retries	PUK 1重试次数
mcm_sim_pin_state_t_v01	pin2_state	当前PIN2状态
uint8_t	pin2_num_retries	PIN 2重试次数
uint8_t	puk2_num_retries	PUK2重试次数

10.5.1.15 struct mcm_sim_card_app_info_t_v01

数据域

Type	参数	描述
mcm_sim_app_info_t_v01	app_3gpp	存储3GPP应用消息
mcm_sim_app_info_t_v01	app_3gpp2	存储3GPP2应用消息
mcm_sim_app_info_t_v01	app_isim	存储ISIM应用消息

10.5.1.16 struct mcm_sim_card_info_t_v01

数据域

Type	参数	描述
mcm_sim_card_state_t_v01	card_state	当前卡机卡错误状态
mcm_sim_card_t_v01	card_t	卡类型
mcm_sim_card_app_info_t_v01	card_app_info	存储所有相关应用消息

10.5.1.17 struct mcm_sim_refresh_file_list_t_v01

数据域

Type	参数	描述
uint32_t	path_value_len	必须设置为参数path_value的长度
char	path_value	路径

10.5.1.18 struct mcm_sim_refresh_event_t_v01

数据域

Type	参数	描述
mcm_sim_application_identification_info_t_v01	app_info	应用识别信息

mcm_sim_- refresh_mode_- t_v01	refresh_mode	刷新模式
uint32_t	refresh_files_- len	必须设置为参数refresh_files的元素个数
mcm_sim_- refresh_file_- list_t_v01	refresh_files	属性文件数据

827455031@qq.com 47.95.190.67 2019/7/18 11:49:59

GOSUNCN Confidential

11 AT命令处理机

本章节主要提供(the Access Terminal Command Processor, ATCoP)相关的消息及常量说明

- [ATCoP Message Identifiers](#)
- [ATCoP Message Structures](#)
- [ATCoP Constants](#)

11.1 ATCoP Message Identifiers

MCM ATCoP相关消息体标识如下:

```
#define MCM_ATCOP_REQ_V01 0x0600
#define MCM_ATCOP_RESP_V01 0x0600
```

11.2 ATCoP消息结构体

本节内容为MCM ATCoP消息体结构说明。

11.2.1 数据结构文档

11.2.1.1 struct mcm_atcop_req_msg_v01

请求消息;与服务器交互的ATCoP消息

数据域

Type	参数	描述
char	cmd_req	请求消息.
uint32_t	cmd_len	请求命令长度

11.2.1.2 struct mcm_atcop_resp_msg_v01

响应消息; 与服务器交互的ATCoP消息

数据域

Type	参数	描述
mcm_response-t_v01	resp	响应码

uint8_t	cmd_resp_valid	如果需要传递参数cmd_resp，该值需要设置为TRUE
char	cmd_resp	命令响应
uint8_t	resp_len_valid	如果需要传递参数resp_len，该值需要设置为TRUE
uint32_t	resp_len	响应消息长度

11.3 ATCoP 常量

该节包含MCM ATCoP相关常量

11.3.1 宏定义集

11.3.1.1 #define MCM_ATCOP_MAX_REQ_MSG_SIZE_V01 513

请求消息最大长度

11.3.1.2 #define MCM_ATCOP_MAX_RESP_MSG_SIZE_V01 4097

响应消息最大长度

12 定位模块

本章规定了与定位（LOC）模块相关的消息体、常量、枚举类型和结构体。

- 定位消息体标识
- 定位消息体结构
- 定位相关常量
- 定位枚举类型

12.1 定位消息体标识

MCM LOC 相关消息体标识如下：

```
#define MCM_LOC_SET_INDICATIONS_REQ_V01 0x0300
#define MCM_LOC_SET_INDICATIONS_RESP_V01 0x0300
#define MCM_LOC_SET_POSITION_MODE_REQ_V01 0x0301
#define MCM_LOC_SET_POSITION_MODE_RESP_V01 0x0301
#define MCM_LOC_START_NAV_REQ_V01 0x0302
#define MCM_LOC_START_NAV_RESP_V01 0x0302
#define MCM_LOC_STOP_NAV_REQ_V01 0x0303
#define MCM_LOC_STOP_NAV_RESP_V01 0x0303
#define MCM_LOC_DELETE_AIDING_DATA_REQ_V01 0x0304
#define MCM_LOC_DELETE_AIDING_DATA_RESP_V01 0x0304
#define MCM_LOC_INJECT_TIME_REQ_V01 0x0305
#define MCM_LOC_INJECT_TIME_RESP_V01 0x0305
#define MCM_LOC_INJECT_LOCATION_REQ_V01 0x0306
#define MCM_LOC_INJECT_LOCATION_RESP_V01 0x0306
#define MCM_LOC_XTRA_INJECT_DATA_REQ_V01 0x0307
#define MCM_LOC_XTRA_INJECT_DATA_RESP_V01 0x0307
#define MCM_LOC_AGPS_DATA_CONN_OPEN_REQ_V01 0x0308
#define MCM_LOC_AGPS_DATA_CONN_OPEN_RESP_V01 0x0308
#define MCM_LOC_AGPS_DATA_CONN_CLOSED_REQ_V01 0x0309
```

```

#define MCM_LOC_AGPS_DATA_CONN_CLOSED_RESP_V01 0x0309
#define MCM_LOC_AGPS_DATA_CONN_FAILED_REQ_V01 0x030A
#define MCM_LOC_AGPS_DATA_CONN_FAILED_RESP_V01 0x030A
#define MCM_LOC_AGPS_SET_SERVER_REQ_V01 0x030B
#define MCM_LOC_AGPS_SET_SERVER_RESP_V01 0x030B
#define MCM_LOC_NI_RESPOND_REQ_V01 0x030C
#define MCM_LOC_NI_RESPOND_RESP_V01 0x030C
#define MCM_LOC_AGPS_RIL_UPDATE_NETWORK_AVAILABILITY_REQ_V01 0x030D
#define
MCM_LOC_AGPS_RIL_UPDATE_NETWORK_AVAILABILITY_RESP_V01 0x030D
#define MCM_LOC_LOCATION_INFO_IND_V01 0x030E
#define MCM_LOC_STATUS_INFO_IND_V01 0x030F
#define MCM_LOC_SV_INFO_IND_V01 0x0310
#define MCM_LOC_NMEA_INFO_IND_V01 0x0311
#define MCM_LOC_CAPABILITIES_INFO_IND_V01 0x0312
#define MCM_LOC_UTC_TIME_REQ_IND_V01 0x0313
#define MCM_LOC_XTRA_DATA_REQ_IND_V01 0x0314
#define MCM_LOC_AGPS_STATUS_IND_V01 0x0315
#define MCM_LOC_NI_NOTIFICATION_IND_V01 0x0316
#define MCM_LOC_XTRA_REPORT_SERVER_IND_V01 0x0317
    
```

12.2 定位消息体的结构

本节内容为MCM LOC消息体结构说明。

12.2.1 数据结构说明

12.2.1.1 struct mcm_loc_set_indications_req_msg_v01

结构类型：请求；

结构功能：注册使能相应的定位消息上报（通知）。

数据域：

Type	参数	描述
uint8_t	register_location_info_ind	注册位置信息主动上报.
uint8_t	register_status_info_ind	注册状态信息主动上报.
uint8_t	register_sv_info_ind	注册卫星信息主动上报.
uint8_t	register_nmea_info_ind	注册NMEA信息主动上报.
uint8_t	register_capabilities_info_ind	注册性能信息主动上报.
uint8_t	register_utc_time_req_ind	UTC时间请求主动上报注册.
uint8_t	register_xtra_data_req_ind	XTRA数据请求主动上报注册.
uint8_t	register_agps_data_conn_cmd_req_ind	AGPS数据链接请求主动上报注册.
uint8_t	register_ni_notify_user_response_req_ind	NI消息用户响应请求主动上报注册.
uint8_t	register_xtra_report_server_ind_valid	若有register_xtra_report_server_ind消息传递则必须设置为TRUE.
uint8_t	register_xtra_report_server_ind	XTRA服务器主动上报注册.

12.2.1.2 struct mcm_loc_set_indications_resp_msg_v01

结构类型：响应；

结构功能：反馈设置定位消息设置的结果。

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

12.2.1.3 struct mcm_loc_set_position_mode_req_msg_v01

结构类型：请求；

结构功能：用于设置定位模式。

数据域：

Type	参数	描述
mcm_gps_position_mode_t_v01	mode	定位模式.
mcm_gps_position_recurrence_t_v01	recurrence	Recurrence.
uint32_t	min_interval	最小时间间隔.
uint32_t	preferred_accuracy	精度期望.
uint32_t	preferred_time	时间期望.

12.2.1.4 struct mcm_loc_set_position_mode_resp_msg_v01

结构类型：响应；

结构功能：反馈设置定位模式的结果。

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

12.2.1.5 struct mcm_loc_start_nav_resp_msg_v01

结构类型：响应；

结构功能：反馈开启导航的结果。

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

12.2.1.6 struct mcm_loc_stop_nav_resp_msg_v01

结构类型：响应；

结构功能：反馈关闭导航的结果。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

12.2.1.7 struct mcm_loc_delete_aiding_data_req_msg_v01

结构类型：请求；

结构功能：请求删除定位辅助数据。

数据域

Type	参数	描述
mcm_gps_aiding_data_t_v01	flags	辅助数据标识.

12.2.1.8 struct mcm_loc_delete_aiding_data_resp_msg_v01

结构类型：响应结构；

结构功能：反馈删除定位辅助数据的结果。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

12.2.1.9 struct mcm_loc_inject_time_req_msg_v01

结构类型：请求；

结构功能：请求注入定位时间。

数据域

Type	参数	描述
int64_t	time	注入时间.
int64_t	time_reference	时间参考.
int32_t	uncertainty	不确定度.

12.2.1.10 struct mcm_loc_inject_time_resp_msg_v01

结构类型：响应；
 结构功能：反馈注入定位时间的结果。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

12.2.1.11 struct mcm_loc_inject_location_req_msg_v01

结构类型：请求；
 结构功能：请求注入初始位置。

数据域

Type	参数	描述
double	latitude	纬度.
double	longitude	经度.
float	accuracy	定位精度.

12.2.1.12 struct mcm_loc_inject_location_resp_msg_v01

结构类型：响应；
 结构功能：反馈为之注入的结果。

数据域

Type	参数	描述
mcm_response-_t_v01	resp	结果码.

12.2.1.13 struct mcm_loc_xtra_inject_data_req_msg_v01

结构类型：请求；
 结构功能：请求注入XTRA数据。由于IPC通信机制限制每次传输的数据不能大于64kB，因此使用时应该把待注入数据分成小块重复调用此接口传输。

数据域

Type	参数	描述
------	----	----

uint32_t	data_len	必须设置为数据总量中的数据单元数.
uint8_t	data	XTRA 数据.

12.2.1.14 struct mcm_loc_xtra_inject_data_resp_msg_v01

结构类型：响应；

结构功能：反馈注入XTRA数据的结果。由于IPC通信机制限制每次传输的数据不能大于64kB，因此使用时应该把待注入数据分成小块重复调用此接口传输数据域

Type	参数	描述
mcm_response-t_v01	resp	结果码.

12.2.1.15 struct mcm_loc_agps_data_conn_open_req_msg_v01

结构类型：请求；

结构功能：请求AGPS网络连接是否打开。

数据域

Type	参数	描述
mcm_agps_t-v01	agps_type	AGPS类型.
char	apn	APN.
mcm_agps-bearer_t_v01	bearer_type	承载类型

12.2.1.16 struct mcm_loc_agps_data_conn_open_resp_msg_v01

结构类型：响应；

结构功能：反馈AGPS网络连接状态请求的结果。

数据域

Type	参数	描述
mcm_response-t_v01	resp	结果码.

12.2.1.17 struct mcm_loc_agps_data_conn_closed_req_msg_v01

结构类型：请求；

结构功能：请求AGPS网络连接是否关闭。

数据域:

Type	参数	描述
mcm_agps_t_v01	agps_type	AGPS 类型.

12.2.1.18 struct mcm_loc_agps_data_conn_closed_resp_msg_v01

结构类型: 响应;

结构功能: 反馈AGPS网络连接状态请求的结果。

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

12.2.1.19 struct mcm_loc_agps_data_conn_failed_req_msg_v01

结构类型: 请求;

结构功能: 请求AGPS数据连接是否启动失败。

数据域

Type	参数	描述
mcm_agps_t_v01	agps_type	AGPS 类型.

12.2.1.20 struct mcm_loc_agps_data_conn_failed_resp_msg_v01

结构类型: 响应;

结构功能: 反馈AGPS数据连接状态请求的结果。

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

12.2.1.21 struct mcm_loc_agps_set_server_req_msg_v01

结构类型：请求；

结构功能：请求设置AGPS服务器参数。

数据域

Type	参数	描述
mcm_agps_t_v01	agps_type	AGPS 类型.
char	host_name	主机名
uint32_t	port	端口.

12.2.1.22 struct mcm_loc_agps_set_server_resp_msg_v01

结构类型：响应；

结构功能：反馈AGPS服务器参数设置结果。

数据域

Type	参数	描述
mcm_response_t_v01	resp	结果码.

12.2.1.23 struct mcm_loc_ni_respond_req_msg_v01

结构类型：请求；

结构功能：发送一条NI消息响应。

数据域

Type	参数	描述
int32_t	notif_id	通知 ID.
mcm_gps_user_response_t_v01	user_response	用户响应.

12.2.1.24 struct mcm_loc_ni_respond_resp_msg_v01

结构类型：响应；

结构功能：反馈发送NI消息响应的结果。

数据域

Type	参数	描述
------	----	----

mcm_response- _t_v01	resp	结果码.
-------------------------	------	------

12.2.1.25 struct mcm_loc_agps_ril_update_network_availability_req_msg_v01

结构类型：请求；

结构功能：更新可用网络状态。

数据域

Type	参数	描述
int32_t	available	可用.
char	apn	APN.

12.2.1.26 struct mcm_loc_agps_ril_update_network_availability_resp_msg_v01

结构类型：响应；

结构功能：反馈可用网络状态更新的结果。

数据域

Type	参数	描述
mcm_response- _t_v01	resp	结果码.

12.2.1.27 struct mcm_loc_location_info_ind_msg_v01

结构类型：通知回调；

结构功能：通知定位结果。

数据域

Type	参数	描述
mcm_gps_ location_t_v01	location	定位信息.

12.2.1.28 struct mcm_loc_status_info_ind_msg_v01

结构类型：通知回调；

结构功能：通知定位状态。

数据域

Type	参数	描述
mcm_gps_status_t_v01	status	GPS 状态.

12.2.1.29 struct mcm_loc_sv_info_ind_msg_v01

结构类型：通知回调；

结构功能：通知可见卫星信息。

数据域

Type	参数	描述
mcm_gps_sv_status_t_v01	sv_info	GPS 卫星状态.

12.2.1.30 struct mcm_loc_nmea_info_ind_msg_v01

结构类型：通知回调；

结构功能：上报NMEA数据。

数据域

Type	参数	描述
int64_t	timestamp	时间戳.
char	nmea	NMEA 消息.
int32_t	length	NMEA 消息长度.

12.2.1.31 struct mcm_loc_capabilities_info_ind_msg_v01

结构类型：通知回调；

结构功能：上报定位能力。

数据域

Type	参数	描述
mcm_gps_capabilities_t_v01	capabilities	性能.

12.2.1.32 struct mcm_loc_agps_status_ind_msg_v01

结构类型：通知回调；

结构功能：上报AGPS状态。

数据域

Type	参数	描述
mcm_agps_status_t_v01	status	AGPS 状态.

12.2.1.33 struct mcm_loc_ni_notification_ind_msg_v01

结构类型：通知回调；

结构功能：上报NI消息内容。

数据域

Type	参数	描述
mcm_gps_ni_notification_t_v01	notification	NI 消息通知.

12.2.1.34 struct mcm_loc_xtra_report_server_ind_msg_v01

结构类型：通知回调；

结构功能：上报XTRA服务器的URL。

数据域

Type	参数	描述
char	server1	Server1.
char	server2	Server2.
char	server3	Server3.

12.3 定位相关常量

本节规定了 MCM LOC 使用的相关常量。

12.3.1 宏定义集

12.3.1.1 #define MCM_LOC_MAX_SEVER_ADDR_LENGTH_CONST_V01 255

服务器最大地址长度.

12.3.1.2 #define MCM_LOC_MAX_GENERIC_STR_LENGTH_CONST_V01 255

通用字符串最大长度.

12.3.1.3 #define MCM_LOC_MAX_APN_NAME_LENGTH_CONST_V01 100

APN最大长度.

12.3.1.4 #define MCM_LOC_MAX_XTRA_DATA_LENGTH_CONST_V01 64512

XTRA数据最大长度.

12.3.1.5 #define MCM_LOC_GPS_LOCATION_MAP_URL_SIZE_CONST_V01 399

定位地图URL长度上限 (用于室内定位).

12.3.1.6 #define MCM_LOC_GPS_LOCATION_MAP_INDEX_SIZE_CONST_V01 16

定位地图编号长度上限 (用于室内定位).

12.3.1.7 #define MCM_LOC_GPS_MAX_SVS_CONST_V01 32

最大可见卫星数.

12.3.1.8 #define MCM_LOC_GPS_SSID_BUF_SIZE_CONST_V01 32

最大SSID缓冲区长度.

12.3.1.9 #define MCM_LOC_IPV6_ADDR_LEN_CONST_V01 16

IPv6地址长度

12.3.1.10 #define MCM_LOC_GPS_NI_SHORT_STRING_MAXLEN_CONST_V01 255

NI 消息短字符串长度上限.

12.3.1.11 #define MCM_LOC_GPS_NI_LONG_STRING_MAXLEN_CONST_V01 2047

NI 消息长字符串长度上限.

12.3.1.12 #define MCM_LOC_GPS_RAW_DATA_MAX_SIZE_CONST_V01 256

原生数据最大长度.

12.3.1.13 #define MCM_LOC_GPS_NMEA_MAX_LENGTH_CONST_V01 255

NMEA 数据最大长度.

12.4 定位枚举类型

本节规定了 MCM LOC 的枚举类型.

12.4.1 枚举类型集

12.4.1.1 enum mcm_gps_position_mode_t_v01

成员:

MCM_LOC_POSITION_MODE_STANDALONE_V01 GPS-standalone模式
MCM_LOC_POSITION_MODE_MS_BASED_V01 AGPS MS-Based 模式
MCM_LOC_POSITION_MODE_MS_ASSISTED_V01 AGPS MS-Assisted 模式.

12.4.1.2 enum mcm_gps_position_recurrence_t_v01

成员:

MCM_LOC_POSITION_RECURRENCE_PERIODIC_V01 GPS在一段时间内连续定位.
MCM_LOC_POSITION_RECURRENCE_SINGLE_V01 请求一次单次定位.

12.4.1.3 enum mcm_gps_aiding_data_t_v01

成员:

MCM_LOC_DELETE_EPHEMERIS_V01 删除星历数据.
MCM_LOC_DELETE_ALMANAC_V01 删除历书数据.
MCM_LOC_DELETE_POSITION_V01 删除位置数据.
MCM_LOC_DELETE_TIME_V01 删除时间数据.
MCM_LOC_DELETE_IONO_V01 删除IONO数据.
MCM_LOC_DELETE_UTC_V01 删除UTC数据.
MCM_LOC_DELETE_HEALTH_V01 删除健康数据.
MCM_LOC_DELETE_SVDIR_V01 删除SVDIR数据.
MCM_LOC_DELETE_SVSTEER_V01 删除 SVSTEER 数据.
MCM_LOC_DELETE_SADATA_V01 删除 SA数据.
MCM_LOC_DELETE_RTI_V01 删除RTI数据.
MCM_LOC_DELETE_CELLDB_INFO_V01 删除 cell DB 信息.
MCM_LOC_DELETE_ALMANAC_CORR_V01 删除星历校正数据.
MCM_LOC_DELETE_FREQ_BIAS_EST_V01 删除频率偏移预估.
MCM_LOC_DELETE_EPHEMERIS_GLO_V01 删除GLO星历数据.
MCM_LOC_DELETE_ALMANAC_GLO_V01 删除GLO历书数据.

MCM_LOC_DELETE_SVDIR_GLO_V01 删除GLO SVDIR数据。
MCM_LOC_DELETE_SVSTEER_GLO_V01 删除GLO SVSTEER数据。
MCM_LOC_DELETE_ALMANAC_CORR_GLO_V01 删除GLO 历书校正数据。
MCM_LOC_DELETE_TIME_GPS_V01 删除GPS时间数据。
MCM_LOC_DELETE_TIME_GLO_V01 Delete time 删除GLO时间数据。
MCM_LOC_DELETE_ALL_V01 删除所有定位数据。

12.4.1.4 enum mcm_agps_t_v01

成员:

MCM_LOC_AGPS_TYPE_INVALID_V01 不可用。
MCM_LOC_AGPS_TYPE_ANY_V01 任何类型。
MCM_LOC_AGPS_TYPE_SUPL_V01 SUPL。
MCM_LOC_AGPS_TYPE_C2K_V01 C2K。
MCM_LOC_AGPS_TYPE_WWAN_ANY_V01 WWAN any。
MCM_LOC_AGPS_TYPE_WIFI_V01 Wi-Fi。
MCM_LOC_AGPS_TYPE_SUPL_ES_V01 SUPL_ES。

12.4.1.5 enum mcm_agps_bearer_t_v01

成员:

MCM_LOC_AGPS_APN_BEARER_INVALID_V01 不可用。
MCM_LOC_AGPS_APN_BEARER_IPV4_V01 IPv4。
MCM_LOC_AGPS_APN_BEARER_IPV6_V01 IPv6。
MCM_LOC_AGPS_APN_BEARER_IPV4V6_V01 IPv4/v6。

12.4.1.6 enum mcm_gps_user_response_t_v01

成员:

MCM_LOC_NI_RESPONSE_ACCEPT_V01 接受。
MCM_LOC_NI_RESPONSE_DENY_V01 拒绝。
MCM_LOC_NI_RESPONSE_NORESP_V01 无响应。

12.4.1.7 enum mcm_gps_location_flag_t_v01

成员:

MCM_LOC_GPS_LOCATION_HAS_LAT_LONG_V01 GPS定位得到可用的经纬度。
MCM_LOC_GPS_LOCATION_HAS_ALTITUDE_V01 GPS定位得到可用的高度。
MCM_LOC_GPS_LOCATION_HAS_SPEED_V01 GPS定位得到可用的速度。
MCM_LOC_GPS_LOCATION_HAS_BEARING_V01 GPS定位得到可用的航向。
MCM_LOC_GPS_LOCATION_HAS_ACCURACY_V01 GPS定位精度可用。
MCM_LOC_GPS_LOCATION_HAS_SOURCE_INFO_V01 GPS定位源信息可用。

MCM_LOC_GPS_LOCATION_HAS_IS_INDOOR_V01 GPS定位“室内标识”可用。
MCM_LOC_GPS_LOCATION_HAS_FLOOR_NUMBE_V01 GPS定位“室内标识”可用。
MCM_LOC_GPS_LOCATION_HAS_MAP_URL_V01 GPS定位地图URL可用。
MCM_LOC_GPS_LOCATION_HAS_MAP_INDEX_V01 GPS定位地图编号可用。

12.4.1.8 enum mcm_gps_status_value_t_v01

成员:

MCM_LOC_GPS_STATUS_NONE_V01 GPS 状态未知。
MCM_LOC_GPS_STATUS_SESSION_BEGIN_V01 GPS导航开启。
MCM_LOC_GPS_STATUS_SESSION_END_V01 GPS 导航停止。
MCM_LOC_GPS_STATUS_ENGINE_ON_V01 GPS 上电没有开启导航。
MCM_LOC_GPS_STATUS_ENGINE_OFF_V01 GPS断电

12.4.1.9 enum mcm_gps_capabilities_t_v01

成员:

MCM_LOC_GPS_CAPABILITY_SCHEDULING_V01 GPS_POSITION_RECURRENCE_PERIODIC模式下的GPS HAL上报时间间隔。如果没有设置，则默认1000ms。
MCM_LOC_GPS_CAPABILITY_MSB_V01 GPS 支持MSB的AGPS模式。
MCM_LOC_GPS_CAPABILITY_MSA_V01 GPS 支持MSA的AGPS模式。
MCM_LOC_GPS_CAPABILITY_SINGLE_SHOT_V01 GPS 支持单次定位。
MCM_LOC_GPS_CAPABILITY_ON_DEMAND_TIME_V01 GPS 支持指令时间注释。

12.4.1.10 enum mcm_agps_status_value_t_v01

成员:

MCM_LOC_GPS_REQUEST_AGPS_DATA_CONN_V01 AGPS请求数据连接。
MCM_LOC_GPS_RELEASE_AGPS_DATA_CONN_V01 AGPS释放数据连接。
MCM_LOC_GPS_AGPS_DATA_CONNECTED_V01 AGPS 数据连接已经启动。
MCM_LOC_GPS_AGPS_DATA_CONN_DONE_V01 AGPS 数据连接已完成。
MCM_LOC_GPS_AGPS_DATA_CONN_FAILED_V01 AGPS 数据连接失败。

12.4.1.11 enum mcm_gps_ni_t_v01

成员:

MCM_LOC_GPS_NI_TYPE_VOICE_V01 Voice。
MCM_LOC_GPS_NI_TYPE_UMTS_SUPL_V01 UMTS SUPL。
MCM_LOC_GPS_NI_TYPE_UMTS_CTRL_PLANE_V01 UMTS control plane。

12.4.1.12 enum mcm_gps_ni_notify_flags_t_v01

成员:

MCM_LOC_GPS_NI_NEED_NOTIFY_V01 NI消息请求通知.
MCM_LOC_GPS_NI_NEED_VERIFY_V01 NI消息请求确认.
MCM_LOC_GPS_NI_PRIVACY_OVERRIDE_V01 NI消息请求无上报通知.

12.4.1.13 enum mcm_gps_ni_encoding_t_v01

成员:

MCM_LOC_GPS_ENC_NONE_V01 None.
MCM_LOC_GPS_ENC_SUPL_GSM_DEFAULT_V01 SUPL GSM default.
MCM_LOC_GPS_ENC_SUPL_UTF8_V01 SUPL UTF8.
MCM_LOC_GPS_ENC_SUPL_UCS2_V01 SUPL UCS2.
MCM_LOC_GPS_ENC_UNKNOWN_V01 Unknown.

12.4.1.14 enum mcm_gps_position_source_t_v01

成员:

MCM_LOC_ULP_LOCATION_IS_FROM_HYBRID_V01 定位源是 ULP.
MCM_LOC_ULP_LOCATION_IS_FROM_GNSS_V01 定位源只有 GNSS.

12.5 定位数据结构

本节规定了 MCM LOC 的数据结构.

12.5.1 数据结构集

12.5.1.1 struct mcm_gps_location_t_v01

数据域:

Type	参数	描述
uint32_t	size	填入mcm_gps_location_t类型的大小
mcm_gps_location_flag_t_v01	flags	包含GPS定位标志位.
mcm_gps_position_source_t_v01	position_source	联合/单GPS 标识.
double	latitude	纬度 (度)
double	longitude	经度 (度)
double	altitude	高度 (米, WGS84坐标系)

float	speed	速度（米/秒）
float	bearing	航向（度）
float	accuracy	定位精度期望值（米）
int64_t	timestamp	定位时间戳
uint32_t	raw_data_len	必须设置为 raw_data 中的成员数目
uint8_t	raw_data	允许HAL传递与定位相关的附加信息
int32_t	is_indoor	室内定位标识
float	floor_number	楼层数目
char	map_url	地图URL.
uint8_t	map_index	地图编号

12.5.1.2 struct mcm_gps_status_t_v01

数据域

Type	参数	描述
uint32_t	size	设置为mcm_gps_status_t结构体的大小.
mcm_gps_status_value_t_v01	status	状态.

12.5.1.3 struct mcm_gps_sv_info_t_v01

数据域

Type	参数	描述
uint32_t	size	设置为mcm_gps_sv_info_t结构体的大小.
int32_t	prn	卫星的伪随机码.
float	snr	信噪比.
float	elevation	卫星仰角（度）
float	azimuth	卫星方位角（度）

12.5.1.4 struct mcm_gps_sv_status_t_v01

数据域

Type	参数	描述
uint32_t	size	设置为mcm_gps_sv_status_t.结构体的大小
int32_t	num_svs	当前可见卫星数.
mcm_gps_sv_info_t_v01	sv_list	卫星信息.
uint32_t	ephemeris_mask	获得星历的卫星掩码标识.

uint32_t	almanac_mask	获得历书的卫星掩码标识.
uint32_t	used_in_fix_mask	最近一次定位用到的卫星掩码标识

12.5.1.5 struct mcm_agps_status_t_v01

数据域

Type	参数	描述
uint32_t	size	设置为mcm_agps_status_t结构体的大小
mcm_agps_t_v01	type	Type.
mcm_agps_status_value_t_v01	status	Status.
int32_t	ipv4_addr	IPv4 address.
char	ipv6_addr	IPv6 address.
char	ssid	SSID.
char	password	Password.

12.5.1.6 struct mcm_gps_ni_notification_t_v01

数据域

Type	参数	描述
uint32_t	size	设置为mcm_gps_ni_notification_t结构体的大小
int32_t	notification_id	HAL生成的ID用来联接NI消息和UI响应
mcm_gps_ni_t_v01	ni_type	NI消息的不同类型
mcm_gps_ni_notify_flags_t_v01	notify_flags	通知/确认 选项。是 GpsNiNotifyFlags类型中的常数组合。
int32_t	timeout	用户响应超时，0代表无超时。
mcm_gps_user_response_t_v01	default_response	响应超时时的默认反馈。
char	requestor_id	Requestor ID.
char	text	NI消息。特定情况下用于存储client ID.
mcm_gps_ni_encoding_t_v01	requestor_id_encoding	Client ID 编码方案
mcm_gps_ni_encoding_t_v01	text_encoding	Client name 编码方案

char	extras	附加数据的指针，格式如下： key_1 = value_1 key_2 = value_2
------	--------	---

827455031@qq.com 47.95.190.67 2019/7/18 11:49:59
GOSUNCN Confidential

13 BSP相关API函数

13.1 gsdk_device_set_rtc_timer

描述：设置RTC定时器，一次性定时器，只能设置一个。

定义：int gsdk_device_set_rtc_timer (unsigned long seconds , gsdk_timer_func fuc, int * timer_id)

参数：

类型	变量	模式	描述
unsigned long	seconds	IN	定时器时长，单位秒
gsdk_timer_func	fuc	IN	定时器回调函数，原型为：typedef(void*gsdk_timer_func) ()
Int*	timer_id	OUT	定时器 ID，可表示设置RTC定时器次数。

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.2 gsdk_device_get_rtc_timer

描述：根据定时器ID，查询RTC定时器剩余时间。

定义：int gsdk_device_get_rtc_timer (int timer_id, unsigned long* seconds)

参数：

类型	变量	模式	描述
Int	timer_id	IN	定时器 ID，必须等于gsdk_device_set_rtc_timer中的*timer_id
unsigned long*	seconds	OUT	定时器剩余时间，单位秒

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.3 gsdk_device_cancel_rtc_timer

描述：根据定时器ID，取消定时器。

定义：int gsdk_device_cancel_rtc_timer (int timer_id)

参数：

类型	变量	模式	描述
Int	timer_id	IN	定时器 ID，必须等于gsdk_device_set_rtc_timer中的*timer_id

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.4 gsdk_device_force_sleep

描述：强制休眠。

定义：int gsdk_device_force_sleep()

参数：无

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.5 gsdk_mcu_wakeup_module

描述：MCU唤醒模块的管脚配置，对应模块PIN72（WAKEUP_IN），上升沿触发可唤醒模块。

定义：int gsdk_mcu_wakeup_module (gsdk_wakeup_func fuc, int * wakeup_id)

参数：

类型	变量	模式	描述
gsdk_wakeup_func	fuc	IN	中断回调函数，原型为： typedef(void*gsdk_wakeup_func) ()
Int*	wakeup_id	OUT	Wakeup_in ID，可表示上升沿中断次数。

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.6 gsdk_modem_wakeup_mcu

描述：模块PIN71 (WAKEUP_OUT) 输出可调信号，可用于唤醒MCU；

定义：int gsdk_modem_wakeup_mcu(int low_delay_ms,int high_delay_ms,int running_time)

参数：

类型	变量	模式	描述
Int	low_delay_ms	IN	低电平持续时间，必须 ≥ 1 ，单位ms
Int	high_delay_ms	IN	高电平持续时间，必须 ≥ 0 ，单位ms
Int	running_time	IN	“低高电平信号”输出次数，必须 ≥ 1

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.7 gsdk_device_get_usb_state

描述：查询USB状态。

定义：int gsdk_device_get_usb_state ()

参数：无

返回值：可参考gsdk_usb_state枚举类型说明。

值	描述
0	USB状态：DISCONNECTED
1	USB状态：CONNECTED
2	USB状态：CONFIGURED
3	USB状态：SUSPENDED
4	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.8 gsdk_device_set_usb_state

描述：设置USB状态，模拟USB热插拔动作。

定义：int gsdk_device_set_usb_state (gsdk_usb_state usb_state)

参数：

类型	变量	模式	描述
gsdk_usb_state	usb_state	IN	0：模拟USB热拔出，进入DISCONNECTED状态。 1：模拟USB热插入，进入CONNECTED状态。 other: error gsdk_usb_state类型参考备注说明。

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

备注：

```
typedef enum{
    USB_DISCONNECTED = 0,
    USB_CONNECTED,
    USB_CONFIGURED,
    USB_SUSPEND,
    USB_RESUMED,
    USB_STATE_ERROR
}gsdk_usb_state;
```

13.9 gsdk_device_get_adc

描述：获取ADC电压值。

定义：int gsdk_device_get_adc(gsdk_adc_channel channel, int* data_uV)

参数：

类型	变量	模式	描述
gsdk_adc_channel	channel	IN	1：ADC1通路，对应模块PIN48。 2：ADC2通路，对应模块PIN47。 gsdk_adc_channel类型参考备注说明。
Int*	Data_uV	OUT	对应通路的电压值，单位uV。

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

备注：

```
typedef enum{
    ADC1 = 1,
```

ADC2

}gSDK_adc_channe;

13.10 gSDK_device_conf_gpio

描述：设置GPIO方向。

定义：int gSDK_device_conf_gpio (int io, int dir)

参数：

类型	变量	模式	描述
int	io	IN	1~11：分别对应模块PIN27、28、29、30、65、57、58、59、60、34、35。
Int	dir	IN	0：输入。 1：输出。

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.11 gSDK_device_read_gpio

描述：获取GPIO电平状态。

定义：int gSDK_device_read_gpio (int io)

参数：

类型	变量	模式	描述
int	io	IN	1~11：分别对应模块PIN27、28、29、30、65、57、58、59、60、34、35。

返回值：

值	描述
0	低电平
1	高电平
-1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.12 gSDK_device_write_gpio

描述：设置输出型的GPIO电平状态。

定义：int gSDK_device_write_gpio (int io, int val)

参数：

类型	变量	模式	描述
int	io	IN	1~11 : 分别对应模块PIN27、28、29、30、65、57、58、59、60、34、35。
Int	val	IN	0 : 低电平。 1 : 高电平。

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

13.13 gsdk_device_gpio_isr

描述：设置输入型GPIO的中断模式。

定义：int gsdk_device_gpio_isr (int io, int irq_flag, gsdk_wakeup_func fun)

参数：

类型	变量	模式	描述
int	io	IN	1~11 : 分别对应模块PIN27、28、29、30、65、57、58、59、60、34、35。 (ME3630对应的中断管脚是： GPIO1\GPIO5\GPIO7\GPIO8\GPIO9，对应模块的 PIN27、65、58、59、60) (ZM5330对应的中断管脚是GPIO2\GPIO3\GPIO5，对应 模块的PIN6、7、61)
int	Irq_flag	IN	0 : 管脚无中断 1 : 上升沿触发中断 2 : 下降沿触发中断 3 : 双边沿触发中断
gsdk_wakeup_func	fun	IN	用户注册回调函数

返回值：

值	描述
0	接口执行成功
1	接口执行失败

函数说明：

同步/异步	超时时间
同步	立即返回

14 WeFOTA相关API函数及使用

14.1 数据类型定义

名称：gsdk_wefota_res_type

含义：wefota相关函数返回值。

类型：枚举

成员及含义：

WEFOTA_ERR_NONE：函数返回值正常，无错误；

WEFOTA_ERR_PATH：差分包路径错误；

WEFOTA_FILE_NONEXIST：差分包路径不存在；

WEFOTA_MSG_SEDD_ERR：进程间通信失败；

WEFOTA_DURING_UPDATE：fota升级正在进行中；

WEFOTA_UPDATE_SUCCESS：fota升级成功；

WEFOTA_UPDATE_FAIL：fota升级失败；

WEFOTA_STATE_FAIL：无法获取fota升级状态；

WEFOTA_IDLE：fota升级未启动。

14.2 函数接口定义

14.2.1 名称：gsdk_wefota_res_type wefota_whole_update(char *whole_pkg_path)

功能：发起一次全版本升级；

参数说明：whole_pkg_path – 全版本升级差分包路径；

返回值：	WEFOTA_ERR_NONE	--无错误；
	WEFOTA_ERR_PATH	--差分包路径不存在；
	WEFOTA_FILE_NONEXIST	--差分包文件不存在；
	WEFOTA_MSG_SEDD_ERR	--进程通信错误；

14.2.2 名称：gsdk_wefota_res_type wefota_local_update(char *app_pkg_path)

功能：发起一次用户app升级；

参数说明：app_pkg_path – 用户app差分包所在路径；

返回值：	WEFOTA_ERR_NONE	--无错误；
	WEFOTA_ERR_PATH	--差分包路径不存在；
	WEFOTA_FILE_NONEXIST	--差分包文件不存在；
	WEFOTA_MSG_SEDD_ERR	--进程通信错误；

14.2.3 名称：gsdk_wefota_res_type get_update_state()

功能：查询升级状态；

参数说明：无参；

返回值：	WEFOTA_DURING_UPDATE	--fota升级正在进行中；
	WEFOTA_UPDATE_SUCCESS	--fota升级成功；
	WEFOTA_UPDATE_FAIL：	--fota升级失败；
	WEFOTA_STATE_FAIL：	--无法获取fota升级状态；
	WEFOTA_IDLE：	--fota升级未启动。

14.3 示例：

14.3.1 全版本升级

(1)将已有文件导入以下两个路径：

\usr\bin\gsdk_client(升级程序，该程序位于我司提供的虚拟机压缩包文件中)

\cache\delta.package(待升级差分包)

(2)运行gsdk_client,依次选择7和1，进入全版本升级，如下图操作：

```

root@mdm9607-perf:/cache# gsdk_client
gsdk_client.c[main, 321]: gsdk_client start!
gsdk_client.c[mcm_gsdk_test_main_menu, 285]:
=====Main Menu=====
1 : gsdk_device_shutdown_system
2 : gsdk_device_reboot_system
3 : gsdk_device_service_manager
4 : gsdk_device_get_adc_value
5 : gsdk_device_get_usb_state_test
6 : gsdk_device_set_usb_state_test
7 : gsdk_wefota_test
x : exit

=====
7
gsdk_client.c[gsdk_wefota_manager, 245]:
===== wefota manager =====
1 : whole update
2 : App update
3 : Get wefota state
x : exit

=====
Enter :
1
    
```

(3)系统开始重启

```

Broadcast message from root@mdm9607-perf (Sun Jan  6 08:06:04 1980):
The system is going down for reboot NOW!
    
```

(4)等待系统重启完成后（重启两次），查看对应分区的内容是否根据预期得到升级、修改。

(5)若升级失败，可查询\cache\logs\logcat，分析日志。

14.3.2 用户app升级

(1)利用adb将gsdk_client(升级程序，该程序位于我司提供的虚拟机压缩包文件中)文件push到/usr/bin目录

eg: adb push C:/Users/Desktop/gsdk/gsdk_client /usr/bin

利用adb将app.package(待升级的ubi格式文件包)push到/cache目录

eg: adb push C:/Users/Desktop/app.package /cache

注意:保证PC机安装adb工具，开发板使能adb端口。

(2)运行gsdk_client

如图1所示,在串口终端输入gsdk_client，回车即可。

```
/ # /usr/bin/gsdk_client
gsdk_client.c[main, 579]: gsdk_client start!
gsdk_client.c[mcm_gsdk_test_main_menu, 459]:
```

图1

(3)菜单项选择

如图2所示,首先选择选项14 gsdk_wefota_test, 然后回车。再选择选项2.App update,回车即可。

```
=====Main Menu=====
0 : exit
1 : gsdk_device_shutdown_system
2 : gsdk_device_reboot_system
3 : gsdk_device_service_manager
4 : gsdk_device_get_adc_value
5 : gsdk_device_get_usb_state_test
6 : gsdk_device_set_usb_state_test
7 : uart test
8 : gsdk_device_get_imei
9 : gsdk_device_get_meid
10 : gsdk_device_get_devicename
11 : gsdk_device_get_swversion
12 : gsdk_device_get_hwversion
13 : gsdk_device_set_syslog_test
14 : gsdk_wefota_test
15 : gsdk_start_soft_timer
```

```
===== wefota manager =====
1 : Whole update
2 : App update
3 : Get wefota state
x : exit
```

图2

(4)等待升级完成

等会儿系统会重启，然后升级，最后重新进入系统.

(5)查询升级结果

两种查询方式:

第一种：读取升级状态

如图3所示.

```
3
gsdk_client.c[gsdk_wefota_manager, 369]: update state = 5
gsdk_client.c[gsdk_wefota_manager, 358]:
===== wefota manager =====
1 : Whole update
2 : App update
3 : Get wefota state
x : exit
```

图3

a.执行gsdk_client

b.选择3.Get wefota state，查看update state确认升级成功

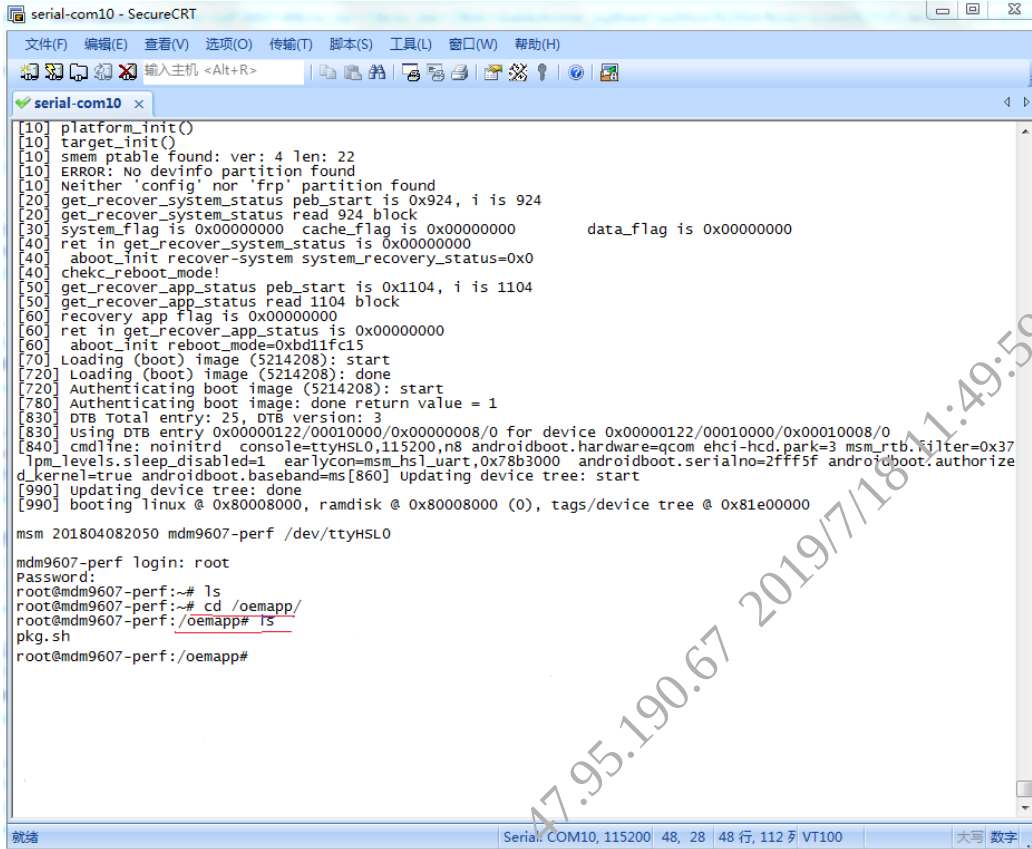
返回值update state的含义：

4	--fota升级正在进行中；
5	--fota升级成功；
6	--fota升级失败；
7	--无法获取fota升级状态；
8	--fota升级未启动。

第二种：直接查看目录

如图4所示.进入oemapp目录，查看升级内容是否为需要的内容。(注意：截图中oemapp是实验所用，客户看到的应该是自己所需升级内容)

注意：重启系统后，如果无法找到gsdk_client，继续按照上面adb操作方式push gsdk_client即可。



```

[10] platform_init()
[10] target_init()
[10] smem ptable found: ver: 4 len: 22
[10] ERROR: No devinfo partition found
[10] Neither 'config' nor 'frp' partition found
[20] get_recover_system_status peb_start is 0x924, i is 924
[20] get_recover_system_status read 924 block
[30] system_flag is 0x00000000 cache_flag is 0x00000000 data_flag is 0x00000000
[40] ret in get_recover_system_status is 0x00000000
[40] about_init recover-system system_recovery_status=0x0
[40] check_reboot_mode!
[50] get_recover_app_status peb_start is 0x1104, i is 1104
[50] get_recover_app_status read 1104 block
[60] recovery app flag is 0x00000000
[60] ret in get_recover_app_status is 0x00000000
[60] about_init reboot_mode=0xbd11fc15
[70] Loading (boot) image (5214208): start
[720] Loading (boot) image (5214208): done
[720] Authenticating boot image (5214208): start
[780] Authenticating boot image: done return value = 1
[830] DTB Total entry: 25, DTB version: 3
[830] using DTB entry 0x00000122/00010000/0x00000008/0 for device 0x00000122/00010000/0x00010008/0
[840] cmdline: nointrd console=ttyHSL0,115200,n8 androidboot.hardware=qcom ehci-hcd.park=3 msm.rtb.filter=0x37
lpm_levels.sleep_disabled=1 earlycon=msm_hsl_uart,0x78b3000 androidboot.serialno=2fff5f androidboot.authorize
d_kernel=true androidboot.baseband=ms[860] Updating device tree: start
[990] updating device tree: done
[990] booting linux @ 0x80008000, ramdisk @ 0x80008000 (0), tags/device tree @ 0x81e00000

msm 201804082050 mdm9607-perf /dev/ttyHSL0

mdm9607-perf login: root
Password:
root@mdm9607-perf:~# ls
root@mdm9607-perf:~# cd /oemapp/
root@mdm9607-perf:/oemapp# ls
pkg.sh
root@mdm9607-perf:/oemapp#
  
```

图4

A MCM APIs使用说明

MCM API是一个面向回调的API，用于设备访问和控制通信。访问MCM框架提供的任何功能函数的主要方法是创建一个请求消息结构体，填充相关的参数，然后通过同步或者异步调用将消息结构体传给MCM框架，该调用会返回一个和请求消息相应的响应消息。另外，可以收到与系统消息或变化相关的指示事件。

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

下面的章节提供了使用IoE MCM框架开发的步骤。

A.1 初始化MCM客户端

在发送其他调用前，MCM客户端必须用下面的代码初始化：

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

形参如下：

ind_cb = 指示回调

async_cb = 异步回调

初始化成功的话，初始化函数返回0。

A.2 用参数初始化请求对象

使用下面的代码创建请求对象并填充相关的参数。例如，创建语音呼叫请求对象：

```
mcm_voice_dial_req_msg_v01req;
```

下面的参数是可选的：

```
req.address_valid=1;

strcpy(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;
```

A.3 创建响应对象并分配内存

使用下面的代码创建响应对象并动态分配内存。例如，创建语音响应对象：

```
mcm_voice_dial_req_msg_v01 req;
rsp = malloc(sizeof(mcm_voice_dial_resp_msg_v01));
memset(rsp, 0, sizeof(mcm_voice_dial_resp_msg_v01));
```

A.4 打电话

该API支持同步和异步调用。例如，拨打异步语音电话：

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

拨打同步语音电话：

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

形参如下：

hndl = A.1节说明了MCM客户端句柄初始化

MCM_VOICE_DIAL_REQ_V01 = 用于区别不同请求的消息ID

req = A.2节说明了请求对象的定义

rsp = A.3说明了响应对象的定义

async_cb = 异步回调函数

token_id = 请求返回Token ID；用于确认后续回调是否是同样的异步请求

A.5 定义异步回调函数

该函数用于接收A-4节所描述的异步回调的响应。例如，定义拨打语音电话的回调函数：

```
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
```

形参如下：

msg_id = 用于区别不同响应类型的消息ID

resp_c_struct = 框架返回的响应对象

token_id = 回调返回的Token ID；该值与之前异步请求返回的值相同

A.6 定义指示回调函数（可选）

该类型的回调通常提供与系统状态变化相关的信息：

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

注册这种类型的回调，必须执行事件注册调用。例如：

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

形参ind_req和int_rsp是请求和响应对象，它们可以用A.3节的方式注册。

A.7 释放客户端句柄

使用下面的代码释放客户端句柄：

```
mcm_client_release(hndl);
```

该函数执行成功会返回0。

A.8 编译代码

按照下面的步骤编译代码：

1.在API文件夹中取到头文件，在存根文件夹中取到mcm_client_stubs.c。

1.用mcm_client_stubs.c文件创建一个共享库（libmcm.so）。例如：

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

参数如下：

arm-none-linux-gnueabi-gcc = 交叉编译器

api = 包含所有MCM头文件的文件夹

3.生成C代码可执行程序的时候链接上面的共享库。例如：

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

参数如下：

sample_code.c = MCM相关的函数的C代码

lmcm = 共享库 libmcm.so

sample_code = 生成的可执行程序的名称