Universal Serial Bus
MultiFlow Extension
for
Mobile Broadband Interface Model

Revision 1.0

July 23, 2017

Revision History

Rev	Date	Filename	Comments
1.0	2017-07-23	MBIM Multiflow v1.0	Initial release

Contributors

Patrik Olesen	Ericsson
Guenter Hildebrandt	Emsys
Reinhold Schneider	Intel
John Bruner	Microsoft
Terry Moore	MCCI
Alan Soloway	Qualcomm
David Sperling	Smith Micro Software
Morten Christiansen	Synopsys

Table of Contents:

1 OVERVIEW 6
1.1 PROBLEM STATEMENT 6
1.2 PROBLEM DEMONSTRATION 7
1.3 ARCHITECTURE 7
1.4 TERMS AND ABBREVIATIONS 8
1.5 SCOPE 9
1.6 EDITORIAL NOTES 9
2 REFERENCES 10
3 MBIM CONTROL MESSAGES 11
3.1 STATUS CODES 11
3.1.1 STATUS CODES 11
4 DATA TRANSPORT 13
4.1 NDP CONTROL 13
4.1.1 NDP16 CONTROL 14
4.1.2 NDP32 CONTROL 14
4.1.3 NDP CONTROL MESSAGE 14
4.2 NDP CONTROL MESSAGE FOR MULTIFLOW 15
4.2.1 MULTIFLOW DISABLED 16
5 MBIM-SPECIFIC COMMAND MECHANISM 17
5.1 MULTIFLOW SERVICE AND CID VALUES 17
5.2 DATA TYPES 18
5.3 MBIM CIDS DETAILED DESCRIPTIONS 19
5.3.1 MBIM_CID_MULTIFLOW_CAPS 19
5.3.2 MBIM_CID_MUTLIFLOW_UPLINK 21
5.3.3 MBIM_CID_MUTLIFLOW_DOWNLINK 23
5.3.4 MBIM_CID_MULTIFLOW_TFT 25
6 MBIM MANDATORY FUNCTIONALITY 27
6.1 MBIM MULTIFLOW MANDATORY BASIC MECHANISMS 27

List of Figures

Figure 1: MultiFlow Architecture 7

List of Tables

Table 1-1: Terms and Abbreviations 8
Table 3-1: Generic Status Codes and their Relevant CIDs 11
Table 3-2: MBIM_STATUS_CODES 12
Table 4-1: Modified MBIM Data Transport signature values for NDP structures 13
Table 4-2: 16-bit NCM Datagram Pointer Signature – Modified for NDP Control 14
Table 4-3: 32-bit NCM Datagram Pointer Table Signature – Modified for NDP Control 14
Table 4-4: NDP Control Message 14
Table 4-5: Control Message Types 15
Table 4-6: MBIM_MULTIFLOW_ STATUS 15
Table 4-7: MultiFlow Control Message 15
Table 5-1: Services Defined by MBIM MultiFlow 17
Table 5-2: Defined CIDs and Message Formats for Commands and Results 17
Table 5-3: Parameters 19
Table 5-4: MBIM_MULTIFLOW_CTRL_CAPS 19
Table 5-5: MBIM_MULTIFLOW_CAPS_INFO 20
Table 5-6: Parameters 21
Table 5-7: MBIM_MULTIFLOW _STATE 21
Table 5-8: MBIM_SET_MULTIFLOW_STATE 22
Table 5-9: MBIM_MULTIFLOW_STATE_INFO 22
Table 5-10: Parameters 23
Table 5-11: Parameters 25
Table 5-12: MBIM_MULTIFLOW_TFT_INFO 26

1 OVERVIEW

This specification is an extension to the Mobile Broadband Interface Model [MBIM] to support MultiFlow.

1.1 PROBLEM STATEMENT

New generations of cellular network technology like LTE Advanced are expected to approach 1 Gigabit per second throughput within a few years. As a result of these advances, networks are imposing new requirements for quality of service and multiple streams. It is clear to the USB Implementers Forum [USB-IF] that these changes will expose technical limitations in the existing MBIM specification related to quality of service and multiple data sessions. The MBIM Multiple Flow Control Initiative ("MBIM MultiFlow") outlined below, which is an approved initiative by USB-IF, will address an expected user experience issue related to flow control with multiple simultaneous data sessions.

The current MBIM 1.0 specification was published late 2011. MBIM 1.0 is based on the NCM specification published early 2009. MBIM is the native mobile broadband connectivity in Win8 onwards, and is supported by multiple other operating systems. NCM (or derived from NCM) is the de-facto connectivity for earlier operating systems. NCM-derived protocols are used for modem to application processor communication in most smartphones with a separate application processor and modem. NCM and MBIM specifications have served the industry well, so far.

MBIM 1.0 uses one bulk USB endpoint pair to multiplex multiple data streams. This was a pragmatic design choice as 3G networks in practice only supported one simultaneous data stream. However, IMS, VoLTE, video, and other new LTE use cases require multiple data streams. Each simultaneous data stream has different Quality of Service ("QoS") requirements and differentiated bandwidth and latency priority in the network.

It is well known in the Internet Protocol world that flow control on multiple TCP streams needs to be end-to-end. Intermediate flow control and buffering can greatly reduce data throughput even if the mobile network is working as expected. Unfortunately, multiplexing all service types over the existing MBIM bulk pipe introduces common flow control and intermediate buffering.

The end user can experience loss of throughput or total loss of service on one data stream due to flowstop on a parallel simultaneous data stream. Intermittent and/or random loss of service on one data stream that does not in itself experience flowstop results in a poor user experience and the cellular network operator is seen as responsible. This problem affects multiple device categories including WAN connected laptops, tethered mobile phones, wireless modems, wireless routers, or any device that can provide multiple simultaneous data sessions to a single cellular operator.

1.2 PROBLEM DEMONSTRATION

The following videos demonstrate the MultiFlow problem:

Demonstration #1 (NCM):

https://youtu.be/GzRWpqzlRaw

Demonstration #2 (MBIM):

https://youtu.be/2d3z-Sf-1QE

1.3 ARCHITECTURE

This MBIM Extension uses a simple buffer watermark design. This allows the host computer to understand the state of the uplink and downlink buffers in the modem to reduce congestion. Figure 1 shows the uplink case.

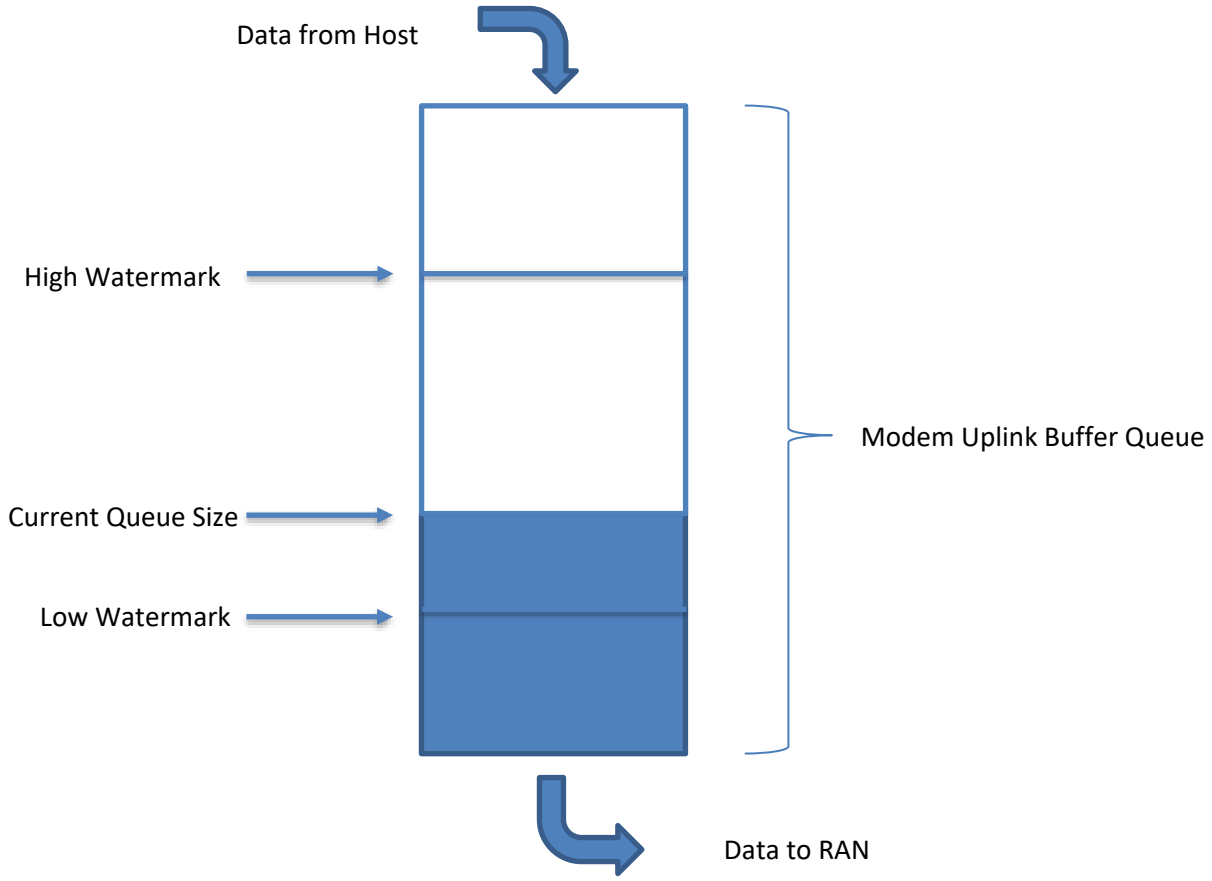

Figure 1: MultiFlow Architecture

1.4 TERMS AND ABBREVIATIONS

Table 1-1: Terms and Abbreviations

CDC	Communications Device Class specification [USBCDC12]
CID	Command Identifier
MB	Mobile Broadband
MBIM	Mobile Broadband Interface Model
MBIM MultiFlow	This specification
MNO	Mobile Network Operator
NA	Not applicable
Transaction	A message sent by the host and the corresponding response from the function.
USB	Universal Serial Bus
Watermark	Used in flow control to determine when data should stop flowing and start flowing

1.5 SCOPE

This document specifies new functionality for mobile broadband based networking devices, defined as an extension of the Mobile Broadband Interface Model specification [MBIM]. This specification defines the following material applicable to MBIM functions:

- The format of data that is exchanged between device and host.
- The required operational behavior.

Behavior that is required of all USB devices and hosts is defined by [USB30]. Behavior that is required of all Communications devices is defined by [USBCDC12]. Some commands and notifications may be based on material defined in [MBIM].

This specification imposes requirements on devices not on hosts. Accordingly, normative language throughout (see section 1.6) applies strictly to devices and not to hosts.

1.6 EDITORIAL NOTES

In this specification, the words ‘shall’ and ‘must’ are used for mandatory requirements, the word ‘should’ is used to express recommendations and the word ‘may’ is used for options.

In some cases, text from [MBIM] is repeated or referred to for clarity. In such cases, the MBIM specification shall be treated as the controlling document.

2 REFERENCES

[USBNCM10]	Universal Serial Bus Subclass Specification for Network control Model (NCM), Version 1.0 http://www.usb.org
[MBIM]	Universal Serial Bus Communications Class Subclass Specification for Mobile Broadband Interface Model, Revision 1.0, November 14, 2011.
[MBIMRegistry]	MBIM Registry at www.usb.org For registration requests, please contact admin@usb.org
[RFC 2396]	T. Berners-Lee, R. Fielding and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.
[USBCDC12]	Universal Serial Bus Class Definitions for Communications Devices, Revision 1.2. http://www.usb.org.
[3GPP TS 24.008]	3GPP specification that defines Traffic Flow Templates. See section 10.5.6.12.

3 MBIM CONTROL MESSAGES

3.1 STATUS CODES

3.1.1 STATUS CODES

For a complete list of the Status codes and their descriptions, please refer to Table 3-2: MBIM_STATUS_CODES.

Table 3-1: Generic Status Codes and their Relevant CIDs

Generic Status Codes	Relevant Commands
MBIM_STATUS_SUCCESS	All CIDs
MBIM_STATUS_BUSY	All CIDs
MBIM_STATUS_FAILURE	All CIDs
MBIM_STATUS_NO_DEVICE_SUPPORT	All CIDs
MBIM_STATUS_NOT_INITIALIZED	MBIM_CID_MULTIFLOW_TFT

Table 3-2: MBIM_STATUS_CODES

Name	Value	Description
MBIM_STATUS_SUCCESS	0	The operation succeeded.
MBIM_STATUS_BUSY	1	The operation failed because the device is busy. In the absence of any explicit information from the function to clear this condition, the host can use subsequent actions by the function (e.g., notifications or command completions) as a hint to retry the failed operation.
MBIM_STATUS_FAILURE	2	The operation failed (a generic failure).
MBIM_STATUS_NO_DEVICE_SUPPORT	9	The operation failed because the device does not support the operation.
MBIM_STATUS_NOT_INITIALIZED	14	The operation failed because the device is in the process of initializing.
Device service specific status commands	80000000h-FFFFFFFFh	

4 DATA TRANSPORT

The data packets will be passed to and from the function using the standard NCM 1.0 transfer headers and datagram pointers are described in section 3.2 of [USBNCM10] and section 7 of [USBMBIM10].

The MBIM data transport is augmented for MultiFlow to also carry the watermark control information. Since the data transport function is responsible for moving packets between the function and the host, it will be also responsible for notifying MultiFlow watermark events.

The mechanism to deliver this control information over the data transport is defined in a general purpose manner such that future extensions can use “NDP Control” for additional use cases.

Since IP data Streams and Device Service Streams have independent identification space, associated control streams will also independently exist.

4.1 NDP CONTROL

NDP Control messages are added to the data transport and are identified by a new textual string and the SessionId in the signature as shown in Figure 3.

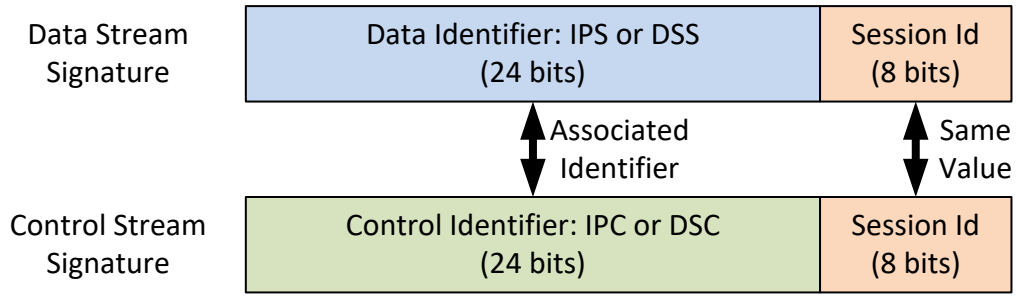

Figure 1 - NDP Control Header Signature

Table 4-1: Modified MBIM Data Transport signature values for NDP structures

Protocol	NDP16	NDP32	Description
Raw IPv4 or IPv6 Control	“IPC”<SessionId>	“ipc”<SessionId>	Raw IPv4 or IPv6 Control payload
Raw IPv4 or IPv6	“IPS”<SessionId>	“ips”<SessionId>	Raw IPv4 or IPv6 payload
Device Service Control	“DSC”<DssSessionId>	“dsc”<DssSessionId>	Device Service Control payload
Device Service Stream	“DSS”<DssSessionId>	“dss”<DssSessionId>	Device Service Stream payload

The *datagramIndex* in the "NDP Control" structure represents a control message index rather than a datagram index.

4.1.1 NDP16 CONTROL

Format of an NDP16 is identical to existing NDP16; only the interpretation of *dwSignature* field has changed:

Table 4-2: 16-bit NCM Datagram Pointer Signature – Modified for NDP Control

Offset	Size	Field	Value	Description
0	4	dwSignature	Number	Signature of an NDP16. Signature corresponds to "IPC"<SessionId> where "IPC" indicates that this carries an IP data Control packet or "DSC"<Session_Id> where "DSC" indicates that this carries a Device Service Control packet and SessionId representing the actual session id value. Signature is transmitted in little- endian form.

4.1.2 NDP32 CONTROL

Format of an NDP32 is identical to existing NDP32; only the interpretation of *dwSignature* field has changed:

Table 4-3: 32-bit NCM Datagram Pointer Table Signature – Modified for NDP Control

Offset	Size	Field	Value	Description
0	4	dwSignature	Number	Signature of a NDP32. Signature corresponds to "ipc"<SessionId> where "ipc" indicates that this carries an IP data Control packet or "dsc"<Session_Id> where "dsc" indicates that this carries a Device Service Control packet and SessionId representing the actual session id value. Signature is transmitted in little- endian form.

4.1.3 NDP CONTROL MESSAGE

Each wControlMessageIndex in the NDP16 or NDP32 points to an in-band NDP Control Message, the format of which is shown below.

Table 4-4: NDP Control Message

Offset	Size	Field	Value	Description
0	2	wType	Number	Control message type. See Table 4-5 for defined type codes.

Offset	Size	Field	Value	Description
2	2	wLength	Number	Size of this message. Should be at least 4.
4	wLength-4	Message		Message content.

4.2 NDP CONTROL MESSAGE FOR MULTIFLOW

Following control message types are currently defined:

Table 4-5: Control Message Types

Types	Value	Length	Description
MBIM_MULTIFLOW_STATUS	0	2	MultiFlow status message. Format of this message is shown in Table 4-6.

Table 4-6: MBIM_MULTIFLOW_ STATUS

Types	Value	Description
MBIMMultiFlowStatusOK	0	Buffer is between high and low watermark
MBIMMultiFlowStatusHighWatermark	1	Buffer is above the high watermark
MBIMMultiFlowStatusLowWatermark	2	Buffer is below the low watermark

Table 4-7: MultiFlow Control Message

Offset	Size	Field	Value	Description
0	2	wMultiFlowStatus	MBIM_MULTIFLOW_STATUS	Watermark status of the reverse link. A value from MBIM_MULTIFLOW_STATUS See Table 4-6.
2	4	dwMultiFlowWatermark	Number	The number of payload bytes in the queue of the reverse link. A value of 0xFFFFFFFF indicates that this is not reported.

4.2.1 MULTIFLOW DISABLED

If MultiFlow is not enabled, then:

- wMultiFlowState MUST equal 0
- dwMultiFlowWatermark MUST equal 0

See MBIM_CID_MULTIFLOW_UPLINK and MBIM_CID_MULTIFLOW_DOWNLINK for enabling/disabling MultiFlow.

5 MBIM-SPECIFIC COMMAND MECHANISM

Each CID allows a Query operation from the host to the function. A Query enables the host to inquire as to the value or status of a given capability at the function. CIDs may also allow indications from the function to the host, both to notify about command completions as well as for unsolicited events (also known as “unsolicited indications”, “notifications” or “events”).

5.1 MULTIFLOW SERVICE AND CID VALUES

This specification defines the following services:

Table 5-1: Services Defined by MBIM MultiFlow

Service Name	UUID	UUID Value
MultiFlow	UUID_MULTIFLOW	8d8f32d9-f7c2-4419-b28b-dedcde205ed8

The following table specifies the command code for each CID. Additionally, it also shows informatively, whether the CID supports Query, Set or Events (notifications). Per-CID details about what (if any) is the structure used for input parameters on a Query (Query InformationBuffer Payload), on a Set (Set InformationBuffer Payload), on command completion (Completion InformationBuffer payload), and on unsolicited Indications (asynchronous events also known as Notifications). Unsolicited indications are delivered via MBIM_INDICATE_STATUS_MSG from the function to the host.

Table 5-2: Defined CIDs and Message Formats for Commands and Results

CID	UUID_	Command Code	Set	Query	Notification
MBIM_CID_MULTIFLOW_CAPS	MULTIFLOW	1	N	Y	N
MBIM_CID_MULTIFLOW_UPLINK	MULTIFLOW	2	Y	Y	N
MBIM_CID_MULTIFLOW_DOWNLINK	MULTIFLOW	3	Y	Y	N
MBIM_CID_MULTIFLOW_TFT	MULTIFLOW	4	N	Y	Y

5.2 DATA TYPES

MBIM MultiFlow uses the Data Types detailed in Table 10.5 of [MBIM].

5.3 MBIM CIDS DETAILED DESCRIPTIONS

This section contains the detailed CID descriptions for the defined device services.

5.3.1 MBIM_CID_MULTIFLOW_CAPS

5.3.1.1 DESCRIPTION

Query only:

The function prepares and returns an MBIM_MULTIFLOW_CAPS_INFO structure in response to a MBIM_COMMAND_MSG with UUID_MULTIFLOW and CID MBIM_CID_MULTIFLOW_CAPS.

5.3.1.2 PARAMETERS

Table 5-3: Parameters

	Set	Query	Notification
Command	NA	Empty	NA
Response	NA	MBIM_MULTIFLOW_CAPS_INFO	NA

5.3.1.3 DATA STRUCTURES

This is a bitmap that represents the control capabilities that the function supports. The following table shows the valid MBIM_MULTIFLOW_CTRL_CAPS settings.

Table 5-4: MBIM_MULTIFLOW_CTRL_CAPS

Types	Mask	Description
MBIMMultiFlowCtrlCapsNone	0h	No flags set
MBIMMultiFlowCtrlCapsUplink	1h	Function supports flow control on the uplink. If this flag is set, the function shall support MBIM_CID_MUTLIFLOW_UPLINK.
MBIMMultiFlowCtrlCapsDownlink	2h	Function supports flow control on the downlink. If this flag is set, the function shall support MBIM_CID_MUTLIFLOW_DOWNLINK.

5.3.1.4 QUERY

The InformationBuffer shall be null and InformationBufferLength shall be zero.

5.3.1.5 RESPONSE

The following structure shall be used in the InformationBuffer

Table 5-5: MBIM_MULTIFLOW_CAPS_INFO

Offset	Size	Field	Type	Description
0	4	ControlCaps	MBIM_MULTIFLOW_CTRL_CAPS	Control capabilities. See Table 5-4: MBIM_MULTIFLOW_CTRL_CAPS

5.3.1.6 STATUS CODES

This CID only uses Generic Status Codes (see Status Codes section 3.1.1).

5.3.2 MBIM_CID_MUTLIFLOW_UPLINK

5.3.2.1 DESCRIPTION

The function prepares and returns an MBIM_MULTIFLOW_STATE_INFO structure in response to an MBIM_COMMAND_MSG with UUID_MULTIFLOW and MBIM_CID_MULTIFLOW_UPLINK. This structure contains the status of Uplink MultiFlow in the function. Functions supporting MultiFlow are in an initial state of MBIMMultiFlowOff. The host should set the state to MBIMMultiFlowOn prior to connecting with MBIM_CID_CONNECT. However, functions shall support the ability to support a Set operation during a connection.

Query:

No input parameters, so InformationBuffer on Query message is not used.

MBIM_MULTIFLOW_STATE_INFO returned in the InformationBuffer of MBIM_COMMAND_DONE.

Set:

The Event InformationBuffer contains an MBIM_MULTIFLOW_STATE_INFO structure.

Unsolicited Event:

Not applicable

5.3.2.2 PARAMETERS

Table 5-6: Parameters

	Set	Query	Notification
Command	MBIM_SET_MULTIFLOW_STATE	Empty	NA
Response	MBIM_MULTIFLOW_STATE_INFO	MBIM_MULTIFLOW_STATE_INFO	NA

5.3.2.3 DATA STRUCTURES

Table 5-7: MBIM_MULTIFLOW _STATE

Types	Value
MBIMMultiFlowOff	0
MBIMMultiFlowOn	1

5.3.2.4 SET

The following structure shall be used in the InformationBuffer

Table 5-8: MBIM_SET_MULTIFLOW_STATE

Offset	Size	Field	Type	Description
0	4	MultiFlowState	MBIM_MULTIFLOW_STATE	Sets the software controlled MultiFlow state.

5.3.2.5 QUERY

The InformationBuffer shall be null and InformationBufferLength shall be zero

5.3.2.6 RESPONSE

The following structure shall be used in the InformationBuffer

Table 5-9: MBIM_MULTIFLOW_STATE_INFO

Offset	Size	Field	Type	Description
0	4	MultiFlowState	MBIM_MULTIFLOW_STATE	One of the values in Table 5-7: MBIM_MULTIFLOW_STATE

5.3.2.7 STATUS CODES

This CID only uses Generic Status Codes (see Status Codes section 3.1.1).

5.3.3 MBIM_CID_MUTLIFLOW_DOWNLINK

5.3.3.1 DESCRIPTION

The function prepares and returns an MBIM_MULTIFLOW_STATE_INFO structure in response to an MBIM_COMMAND_MSG with UUID_MULTIFLOW and MBIM_CID_MULTIFLOW_DOWNLINK. This structure contains the status of Downlink MultiFlow in the function. Hosts supporting MultiFlow are in an initial state of MBIMMultiFlowOff. The host should set the state to MBIMMultiFlowOn prior to connecting with MBIM_CID_CONNECT. However, functions shall support the ability to support a Set operation during a connection.

Functions that support MultiFlow may opt to support Uplink MultiFlow only. In this case, the MBIM_CID_MUTLIFLOW_DOWNLINK shall not be supported.

Query:

No input parameters, so InformationBuffer on Query message is not used.

MBIM_MULTIFLOW_STATE_INFO returned in the InformationBuffer of MBIM_COMMAND_DONE.

Set:

The Event InformationBuffer contains an MBIM_SET_MULTIFLOW_STATE structure.

Unsolicited Event:

Not applicable

5.3.3.2 PARAMETERS

Table 5-10: Parameters

	Set	Query	Notification
Command	MBIM_SET_MULTIFLOW_STATE	Empty	NA
Response	MBIM_MULTIFLOW_STATE_INFO	MBIM_MULTIFLOW_STATE_INFO	NA

5.3.3.3 DATA STRUCTURES

None

5.3.3.4 SET

The same MBIM_SET_MULTIFLOW_STATE data structure is used from MBIM_CID_MULTIFLOW_UPLINK

5.3.3.5 QUERY

The InformationBuffer shall be null and InformationBufferLength shall be zero

5.3.3.6 RESPONSE

The MBIM_MULTIFLOW_STATE_INFO structure shall be used in the InformationBuffer.

5.3.3.7 STATUS CODES

This CID only uses Generic Status Codes (see Status Codes section 3.1.1).

5.3.4 MBIM_CID_MULTIFLOW_TFT

5.3.4.1 DESCRIPTION

This command returns information about the current Traffic Flow Templates (TFT) assigned to the modem for the current data session.

Query:

InformationBuffer of **Query** message is empty. InformationBuffer of MBIM_COMMAND_ DONE contains an MBIM_MULTIFLOW_TFT_INFO.

Unsolicited Event:

The Event InformationBuffer contains an MBIM_MULTIFLOW_TFT_INFO structure.

5.3.4.2 PARAMETERS

Table 5-11: Parameters

	Set	Query	Notification
Command	NA	Empty	NA
Response	NA	MBIM_MULTIFLOW_TFT_INFO	MBIM_MULTIFLOW_TFT_INFO

5.3.4.3 DATA STRUCTURES

5.3.4.4 QUERY

The InformationBuffer shall be null and InformationBufferLength shall be zero.

The function shall be able to present the entire TFT state at any time.

5.3.4.5 RESPONSE

Table 5-12: MBIM_MULTIFLOW_TFT_INFO

Offset	Size	Field	Type	Description
0	4	SessionId	UINT32	TFT's are specified in a per-SessionId basis.
4	4	ElementCount (EC)	UINT32	Count of TFT's following this element
8	8*EC	TFTList	OL_PAIR_LIST	The first element of the pair is a 4 byte Offset in bytes, calculated from the beginning (offset 0) of this MBIM_MULTIFLOW_TFT_INFO structure, to TFT octets. The second element of the pair is a 4 byte size of the record element.
8+8*EC		DataBuffer	DATABUFFER	TFT octets as defined by [3GPP TS 24.008]

5.3.4.6 NOTIFICATION

See Table 5-12: MBIM_MULTIFLOW_TFT_INFO

5.3.4.7 STATUS CODES

This CID only uses Generic Status Codes (see Status Codes section 3.1.1).

6 MBIM MANDATORY FUNCTIONALITY

6.1 MBIM MULTIFLOW MANDATORY BASIC MECHANISMS

The minimum support for MBIM MultiFlow support are the following CID's.

1. MBIM_CID_MULTIFLOW_CAPS
2. MBIM_CID_MULTIFLOW_UPLINK

All other CIDs are optional.