

MMA Technical Standards Board/ AMEI MIDI Committee

Letter of Agreement for Recommend Practice Mobile XMF Content Format Specification (RP-042)

Originated by: <u>MMA</u>	Reference MMA Item #: <u>179b</u>	TSBB #: <u>28</u>
Issue Date: <u>Feb 3 2004</u>	Approval Date: <u>Jan 18 2004</u>	Revision Date: _____
Related item(s):	<u>Mobile DLS (RP-041), XMF Meta File Format 2.0 (RP-043)</u> <u>XMF Specification, SMF Specification, DLS Specification</u>	
Agreed to by MMA: _____	_____	Date: _____
<i>signature</i>	<i>title</i>	
Agreed to by AMEI: _____	_____	Date: _____
<i>signature</i>	<i>title</i>	

Source:

MMA Scalable MIDI Working Group, AMEI Mobile DLS Working Group

Abstract:

The Mobile XMF Content Format Specification describes the method for bundling Standard MIDI Files with Mobile DLS instrument data, including mechanisms for indicating what type of Mobile DLS instruments are used. Player requirements are also included.

Background:

This document specifies in detail all aspects of containment and external signaling that must be addressed for interchange of Mobile DLS compound content (MIDI file plus instruments).

Proposal:

Proposal begins on the following page.

Contents

1.	Introduction	4
2.	Definitions of Terms	4
3.	File Layout and Supported Resource Types.....	5
4.	Interoperable SMF Content.....	6
	SMF Content Format	6
	SMF File Format	6
	MIDI message support.....	6
	MIDI System Exclusive messages.....	6
	SMF Meta Events	6
5.	Interoperable Instrument Content	6
	Instrument Data Format	6
	Constraints on Instrument Content	6
6.	How Instruments are Accessed With MIDI Messages	6
	6.1 Content Conventions	7
	6.2 Synthesizer Behaviors	7
7.	Use of XMF Meta File Format Features.....	7
	7.1 Standard XMF Meta File Format Features	7
	XMF Meta File Format Version.....	8
	ReferenceTypeIDs	8
	UnpackerIDs	8
	Standard Meta-Data Fields.....	8
	Custom Meta-Data Fields	8
	International Meta-Data.....	8
	VLQ Field Maximum Sizes/Values	8
	7.2 Additional Restrictions on Use of XMF Meta File Format	10
8.	Content Handling Behaviors.....	10
	8.1 Clear Instruments from Old Mobile XMF Documents	10
	8.2 Load the Mobile DLS File from the Mobile XMF Document.....	10
	8.3 Prepare to Play SMF	10
	8.4 Load Errors Prevent Playback	10

9. Content Type Indication	10
9.1 Internal Content Type Indication	10
XMF File Type.....	11
Spec Revision Level	11
9.2 External Content Type Indication	11
Filename Extension	11
MIME Media Type.....	11
10. Content Description MetaDataItem	11
10.1 Purpose.....	11
10.2 Usage.....	11
10.3 Contents Format.....	11
10.3.1 MIP Message Reference	12
10.3.2 Number of MIDI Channels Described.....	12
10.3.3 Number of Playback Resources Described	12
10.3.4 Playback Resource List.....	13
10.3.5 Playback Resource Group List	16
10.3.6 MIR Count Table.....	17
10.4 Example Content Description MetaDataItem.....	18
11. References	19

1. Introduction

This document specifies in detail all aspects of containment and external content type indication that must be addressed in Mobile DLS compound documents, i.e. content that bundles SMF with Mobile DLS instruments.

2. Definitions of Terms

Mobile DLS – Specification for waveform-based musical instrument content format. See definition in [7].

Player – Hardware or software that reads files in the specified compound document format, parses the compound document, and plays the content.

Renderer – Hardware or software MIDI synthesizer or sampler for which instrument definition data is being supplied, and to which the SMF's MIDI events will be played.

SMF – Acronym for Standard MIDI File (see below). See definition in [5].

SP-MIDI – Acronym for Scalable Polyphony MIDI. See [10]

Standard MIDI File – Specification for MIDI message content format. See definition in [5].

VLQ – Variable Length Quantity, an integer number representation format used in XMF files. See definition in section 4.1 of [1]. Note that the VLQ definition for XMF is slightly different from the VLQ definition used in Standard MIDI Files (see [5]), however all SMF VLQs will be correctly interpreted by an XMF VLQ reader.

XMF – Acronym for eXtensible Music Format as defined in [1] and modified in [3].

XMF File Type – See definition in [1].

XMF Meta File Format – See definition in [1] and modifications in [3].

3. File Layout and Supported Resource Types

A Mobile XMF document must contain one SMF file, and may contain zero or one #179a Mobile DLS files. Both must be located in the Mobile XMF document's RootNode (see [1]). Any Mobile DLS file must appear before the SMF file. Detached nodes (ReferenceTypeID 3, as defined in [1]) may not be used. An example file layout is shown below.

Start of XMF Document

FileHeader			
Tree	RootNode (a FolderNode)	Mobile DLS FileNode	Mobile DLS Meta-Data
			Mobile DLS File
		SMF FileNode	SMF Meta-Data incl. Content Description
			SMF File

End of XMF Document

Notes:

The Mobile DLS file (and the FileNode that contains it) is optional.

'Mobile DLS Meta-Data' means: One MetaDataItem for Resource Format (the ResourceFormatID must be 5, meaning Mobile DLS).

'SMF Meta-Data' means: One or more MetaDataItems for Content Description (see section 10), and one MetaDataItem for Resource Format (the ResourceFormatID may be either 0 [meaning SMF Format 0] or 1 [meaning SMF Format 1]).

4. Interoperable SMF Content

A Mobile XMF document must contain one SMF file [5]. The SMF file may access custom Mobile DLS instruments provided in the same Mobile XMF document, or GM instruments that the player is required to provide (see section 6). The SMF file must conform to the following requirements:

SMF Content Format	The SMF file must be SP-MIDI-compliant [6].
SMF File Format	The SMF file may be either Format 0 (single track), or Format 1 (multiple tracks playing in parallel). For SMF Format 1, players should support at least 16 tracks. SMF Formats 0 and 1 are specified in [5].
MIDI message support	See SP-MIDI specification [6].
MIDI System Exclusive messages	See SP-MIDI specification [6].
SMF Meta Events	See SP-MIDI specification [6].

5. Interoperable Instrument Content

A Mobile XMF document may contain zero or one Mobile DLS instrument content files [7]. If present, the Mobile DLS file must conform to the following requirements:

Instrument Data Format	The instrument data file must be compliant with the Mobile DLS specification [7].
Constraints on Instrument Content	See the separate Mobile DLS specification document [7].

6. How Instruments are Accessed With MIDI Messages

The SMF content in the Mobile XMF document can use both programs (melodic instruments or percussion sets) contained within the same Mobile XMF document ('custom instruments') and General MIDI programs ([8], [9]) whose data is automatically supplied by the player. This section specifies the content conventions and player behaviors needed to support this functionality.

Bank and Program Addresses – The bank numbers used in the contained Mobile DLS content correspond directly to the bank numbers used in the contained SMF content. For example, to access custom program 12h in bank MSB=42h, LSB=04h, the SMF content would use two Control Change messages (CC0 = 42h, CC32 = 04h) followed by a Program Change message (program 12h). Certain limitations on the use of bank numbers are detailed below.

Overriding a GM Instrument – It is possible, if desired, for a content creator to provide temporary 'override' replacements for the player's built-in GM programs by providing custom programs in the compound document that have bank and program numbers that correspond to the desired target.

6.1 Content Conventions

SMF Content Bank Numbering – The SMF content must comply with the SP-MIDI specifications ([6], [10]), and must also follow General MIDI 2 rules for using MIDI Bank Select and Program Change messages to access the built-in GM instruments (see [9] at sections 3.2 and 3.3.1). Where the SP-MIDI and GM2 specifications diverge with respect to bank selection and program change rules, the player must observe the SP-MIDI specified behavior, not the GM2 rules. The player furnishes a set of GM percussion kit programs at bank MSB 78h LSB 00h, and a set of GM melodic instrument programs at bank MSB 79h and LSB 00h through 09h.

Mobile DLS Content Bank Numbering – Any custom program (melodic instrument or percussion set) may use any bank number (MSB+LSB) and any program number, however content creators should be aware of the following considerations:

Custom programs with bank MSB of 78h and bank LSB of 00h will act as temporary overrides for the player's built-in GM percussion kit programs.

Custom programs with bank MSB of 79h and bank LSB in the range 00h through 09h will act as temporary overrides for the player's built-in GM melodic instrument programs.

No two programs should use the same bank (MSB+LSB) and program number, even if the values of the ulBank bit 31 melodic/percussion bit are different. Programs with the same bank and program will result in an override, and ulBank bit 31 is ignored by the Mobile DLS synthesizer.

All other bank and program numbers are available for use by content authors, without restriction.

6.2 Synthesizer Behaviors

Bank Loading – The player must map the bank select MSB/LSB values in the contained SMF content directly to the identical MSB/LSB bank locations stored in the contained Mobile DLS program (melodic instruments and percussion sets) content, without performing any shifting or other translation. For example, if the MIDI messages in the SMF indicate MSB 12h, LSB 34h, and program 56h, then the player will look for a Mobile DLS instrument with MSB 12h, LSB 34h, and program 56h.

Program Change Processing – When processing a MIDI Program Change message, the player must first seek the requested program (melodic instrument or percussion set) within the contained Mobile DLS content, respecting the current bank select MSB/LSB value for that MIDI channel. If the requested bank and program number is not found in the contained Mobile DLS content, and the bank select MSB+LSB is within the built-in GM instrument range (MSB 78h LSB 00h, or MSB 79h LSB 00h through 09h), then the player must go on to perform a second seek for the indicated program in the player's built-in GM bank(s), using the SP-MIDI rules given in section 2.2.3 of [10] and the GM2 rules given in section 2.6 of [9].

7. Use of XMF Meta File Format Features

7.1 Standard XMF Meta File Format Features

The following tables indicate the XMF Meta File Format features that players must support. Content should not rely upon features that players are not required to support. Players shall properly skip over all unsupported MetaDataItems.

XMF Meta File Format Version	Players must support XMF Meta File Format version 2.00 [11], excepting certain features as detailed below.
ReferenceTypeIDs	Players are required to support only ReferenceTypeIDs 1 (In-Line Resource) and 2 (In-File Resource). Content generated according to this specification shall not rely on players to process any other ReferenceTypeIDs.
UnpackerIDs	Players are not required to support any UnpackerIDs. Content generated according to this specification shall not rely on players to process any UnpackerIDs.
Standard Meta-Data Fields	<p>Players are required to support the following Standard Meta-Data Fields, and are not required to support any others. Content generated according to this specification shall not rely on players to process any Standard Meta-Data FieldIDs other than those listed here:</p> <ul style="list-style-type: none"> • Resource Format (Standard FieldID value 3) – Used to distinguish contained SMF files from contained Mobile DLS instrument files. This field must be attached to every Mobile DLS file and every SMF file in the XMF file. The content is a Standard ResourceFormatID, which must be one of the following three values: <ul style="list-style-type: none"> 0: SMF Format 0 [5] 1: SMF Format 1 [5] 5: Mobile DLS instrument file [7] <p>Note: This specification defines a new Standard ResourceFormatID for Mobile DLS instrument files, value 5, which should be added to the XMF Meta File Format specification.</p> • Content Description (Standard FieldID value 13) – A new field with Universal binary contents (hidden or visible), and an internal structure as described in this specification at section 10. One or more copies of this field collectively describe the playback resource consumption of an SMF in the Mobile XMF document. This MetaDataItem must appear in the FileNode for the corresponding SMF, and nowhere else in the XMF file. <p>Note: This specification defines a new Standard FieldID for XMF meta-data, value 13, which should be added to the XMF Meta File Format specification.</p>
Custom Meta-Data Fields	Players are not required to support Custom Meta-Data fields. Content generated according to this specification shall not rely on players to process any Custom Meta-Data fields.
International Meta-Data	Players are not required to support International meta-data. Content generated according to this specification shall not rely on players to process any international meta-data FieldContents.
VLQ Field Maximum Sizes/Values	See following table.

VLQ Field Maximum Sizes and Values		
VLQ	Max Value	Min Bits Unsigned Integer
FileHeader:		
FileLength	268,435,455	28
TreeStart	268,435,455	28
TreeEnd	268,435,455	28
NodeHeader:		
NodeLength	268,435,455	28
NodeContainedItems	256	8
NodeHeaderLength	65,535	16
NodeMetaData:		
LengthInBytes	65,535	16
Standard FieldID	256	8
Custom Field Specifier (XString length)	256	8
FieldContents:		
length for UniversalContents	65,535	16
NumberOfStrings for InternationalContents	65,535	16
MetaDataType for InternationalContents	65,535	16
VersionData length for InternationalContents	65,535	16
NodeUnpackers:		
ListSizeInBytes	65,535	16
DecodedSize	65,535	16
NodeContents:		
ReferenceTypeID	5	3
MetaDataTypesTable:		
LengthInBytes	65,535	16
NumberOfEntries	65,535	16
MetaDataType in TypeEntry	65,535	16
StringFormatTypeID	7	3

7.2 Additional Restrictions on Use of XMF Meta File Format

Resource blocks (SMF file, Mobile DLS file) must always begin on a word boundary (even byte address, also known as 16-bit boundary). When the resource block start does not naturally fall on a word boundary, this alignment can be achieved in the following manner.

For In-File Resources and In-Line Resources, word alignment can be accomplished by padding, in other words by adding one pad byte (not meaningful data) between the NodeHeader and the NodeContents. It is recommended but not required to use the value 00h for the pad byte.

8. Content Handling Behaviors

Players are required to support the following content handling behaviors.

8.1 Clear Instruments from Old Mobile XMF Documents

To prevent any previously loaded Mobile XMF document instruments from being inadvertently used, and to prevent the build-up of unused instruments in the synthesizer's state, it is necessary to clear any instruments from previously loaded Mobile XMF documents when loading a new Mobile XMF document. All instrument banks should be cleared of any previously loaded instruments that originated in Mobile XMF documents before loading the compound document's Mobile DLS instrument file (if any). Previously loaded instruments that are intended to be treated as permanently installed, including General MIDI instrument permanent replacements, should not be cleared when loading a Mobile XMF document. Any further mechanism for handling persistent instrument content is beyond the scope of this specification.

8.2 Load the Mobile DLS File from the Mobile XMF Document

After clearing old Mobile XMF document instruments, the player must load the Mobile DLS file in the Mobile XMF document (if any) into the synthesizer in preparation for playback. If the Mobile DLS file contain any instruments with the same bank and program numbers as Mobile DLS files loaded at earlier times, then the later programs will overwrite the earlier programs.

8.3 Prepare to Play SMF

Note: This specification only defines playback behavior for a single SMF in a Mobile XMF document. No SMF playlist behavior is defined.

Upon loading a Mobile XMF document, its SMF file should be automatically loaded into the sequencer and prepared for playback, in case the playback has not already started.

8.4 Load Errors Prevent Playback

If any failure is encountered while loading the Mobile XMF document, and the SMF file has not been started, then the SMF file should never be started. If any failure is encountered while loading the Mobile XMF document, and the SMF file has already been started, then the SMF file should be stopped. This is to assure that the content will play as its creator intended. Examples of failure conditions would include: the player does not support the Mobile DLS instrument data format [7]; the Mobile DLS instrument file or SMF file contains invalid data that cannot be parsed; required storage cannot be allocated while parsing the content; etc.

9. Content Type Indication

9.1 Internal Content Type Indication

To provide content type information to XMF file parsers and generic file type recognizers, the XmfFileTypeID field in the FileHeader must indicate the proper XMF File Type number, and the XmfFileTypeRevisionID must indicate the proper spec version:

XMF File Type	2
Spec Revision Level	1 (for version 1.0 of this specification)

9.2 External Content Type Indication

To provide content type information to file systems, file servers, transfer protocols, and services, a Mobile XMF document generated according to this specification must include external indication of the content type:

Filename Extension	<code>.mxmf</code>
MIME Media Type	<code>audio/mobile-xmf</code>

10. Content Description MetaDataItem

This section describes the purpose, format, and usage of the Content Description MetaDataItem, and provides a commented example.

10.1 Purpose

This MetaDataItem is used to characterize the set of resources needed to play an SMF file in the Mobile XMF document, so that the player is able to determine, before commencing playback and without having to analyze the contents of the SMF, whether the SMF can be correctly played using the available resources. If there are not enough resources available to play the SMF file correctly, the player should not start playback of the SMF.

10.2 Usage

This FieldID is valid for File nodes only, not Folder nodes. One or more Content Description MetaDataItem may be attached to any FileNode in the Mobile XMF document that contains an SMF file. There must be one MetaDataItem instance for every MIP message in the SMF file. Typically there may be several of these MetaDataItem for every SMF, all of which must be parsed to fully characterize the SMF's resource consumption.

10.3 Contents Format

The FieldContents format for a Content Description MetaDataItem is Universal, Binary (hidden or visible), and consists of the following sequence of parts; the data format for each part is detailed below.

1. **MIP Message Reference:** The index of the corresponding MIP message in the SMF
2. **Number of Channels:** The count of MIDI channels described in this MetaDataItem
3. **Number of Playback Resources Described** in this MetaDataItem
4. **Playback Resource List (PRL)** identifying the described resources
5. **Playback Resource Group List (PRGL)** identifying the described resources
6. **Maximum Instantaneous Resource (MIR) Count Table:** per-resource usage figures

The described playback resources are identified by a combination of the PRL and the PRGL; every described playback resource belongs to a group (see 5) and has a unique ID within that group (see 4). The interpretation of the MIR Count Table depends on the contents and order of the PRL and PRGL.

Example:

```

00h // Corresponds to the first MIP message in the SMF
04h // 4 MIDI channels in the MIP message
03h // 3 resources described in this MetaDataItem
// Resource column descriptor lists for MIR Count Table:
00h, 01h, 00h, 02h, 00h, 03h // PRL for resources 1, 2, & 3
// 0 = Standard, 1 for Std = "Mobile DLS voices w/o DCF & Vib. LFO"
// 0 = Standard, 2 for Std = "Mobile DLS voices with DCF & Vib. LFO"
// 0 = Standard, 3 for Std = "Total wavetable memory consumption"
00h, 00h, 02h // PRGL for resources 1, 2, & 3
// MIR Count Table: 4 channel rows, 3 resource columns
02h, 00h, 01h, // 1st MIDI channel of the MIP message:
// 2 of 1st rsrc, 0 of 2nd rsrc, 1 or 3rd rsrc
03h, 00h, 01h, // 2nd MIDI channel of the MIP message
05h, 00h, 01h, // 3rd MIDI channel of the MIP message
05h, 02h, 01h // 4th MIDI channel of the MIP message

```

10.3.1 MIP Message Reference

This is a VLQ containing the index of the MIP message in the SMF that this MetaDataItem describes. For example, the first MIP message would be index 0.

Example:

```
02h // Corresponds to the third MIP message in the SMF
```

10.3.2 Number of MIDI Channels Described

This is a VLQ containing the number of MIDI channels described in this MetaDataItem. This count is used to determine the number of rows in the MIR Count Table. The number should agree with the number of MIP channel values present in the corresponding MIP message.

10.3.3 Number of Playback Resources Described

This is a VLQ containing the number of playback resources described in this MetaDataItem. This count is used to determine the number of columns in the MIR Count Table, the number of entries in the Playback Resource List, and the number of entries in the playback resource group list.

10.3.4 Playback Resource List

This is a list of entries that, together with the Playback Resource Groups List, uniquely identify which playback resources are being described in this MetaDataItem. There is one entry per described playback resource (see 10.3.3.). Each entry consists of one VLQ containing a ResourceTypeID prefix of 0 through 5, and a data field (whose interpretation depends on the value of the ResourceTypeID, as described below), describing the resource appearing at the corresponding index in the MIR Count Table entries.

Each defined playback resource belongs to exactly one Playback Resource Group (see 10.3.5).

Example:

```
00h, 01h,    // Prefix 0 = Standard,
              //    1 for Standard = "Mobile DLS voices w/o DCF & Vib. LFO"
00h, 02h,    // Prefix 0 = Standard,
              //    2 for Standard = "Mobile DLS voices with DCF & Vib. LFO"
00h, 03h     // Prefix 0 = Standard,
              //    3 for Standard = "Total wavetable memory consumption"
```

ResourceTypeID (Prefix)	Description	Used with PlaybackResourceGroupIDs
0	Standard	0, 2
1	MMA/AMEI Manufacturer	0, 2
2	Registered	0, 2
3	Non-Registered	0, 2
4	Wavetable Codec wFormatTag	1
5	Wavetable Codec GUID	1

10.3.4.1 Standard PlaybackResourceIDs

Standard PlaybackResourceIDs are prefixed with ResourceTypeID 0 (0x00).

They are used for voice categories, wavetable memory consumption, and may be used in the future for other playback resources. Only MMA/AMEI may define Standard PlaybackResourceIDs. Each PlaybackResourceID value may be used only with its corresponding PlaybackResourceGroupID, as shown in the following table.

Note: Standard PlaybackResourceIDs may not be used for wavetable compression codecs (PlaybackResourceGroupID 1).

PlaybackResourceID Value	Description	For Use with PlaybackResourceGroupID
Synthesizer Voices:		
0	Number of General MIDI 1 voices [8] Note: Any voice layering in the GM bank is not taken into account	0: Synthesizer Voice
1	Number of Mobile DLS voices without DCF & Vib. LFO [7]	0: Synthesizer Voice
2	Number of Mobile DLS voices with DCF & Vib. LFO [7]	0: Synthesizer Voice
Wavetable Memory Consumption:		
3	Total wavetable data consumption (in kilobytes) for Mobile DLS instruments using (16-bit, 8-bit) PCM samples	2: Wavetable Memory Consumption
4	Total uncompressed wavetable data (in kilobytes) for Mobile DLS instruments using compressed samples	2: Wavetable Memory Consumption

Example:

```
00h, // Prefix 0 = Standard
01h // 1 for Standard = Mobile DLS voices w/o DCF & Vib. LFO
```

10.3.4.2 MMA/AMEI Manufacturer PlaybackResourceIDs

MMA/AMEI Manufacturer PlaybackResourceIDs are prefixed with ResourceTypeID 1 (0x01).

They are reserved for manufacturer-specific playback resource types. Each MMA/AMEI Manufacturer is free to create PlaybackResourceIDs based on their own MMA/AMEI Manufacturer ID, subject to the following restrictions. Manufacturers are free to either publish their PlaybackResourceIDs, or keep them proprietary.

A MMA/AMEI Manufacturer PlaybackResourceID contains two parts: first the company's assigned MMA/AMEI Manufacturer ID in ordinary 1-byte or 3-byte form, then a VLQ containing the company's internal ID.

Note that all MMA/AMEI Manufacturer IDs are either one byte long or three bytes long, as indicated by the first byte (0x00 means a three-byte ID).

Note: MMA/AMEI Manufacturer PlaybackResourceIDs may not be used for wavetable compression codecs (PlaybackResourceGroupID 1).

Example:

```
01h,           // Prefix 1 = MMA/AMEI Manufacturer,
00h, 7Ch, 7Fh, // Electric Kazoo MMA/AMEI Mfr. ID
0Ah           // Electric Kazoo's playback resource type 10
```

10.3.4.3 Registered PlaybackResourceIDs

Registered PlaybackResourceIDs are prefixed with ResourceTypeID 2 (0x02).

They are reserved for custom or proprietary playback resources, but are not linked to MMA/AMEI Manufacturer IDs, and may only be defined and assigned by the MMA/AMEI. This allows non-MMA/AMEI manufacturers access to extensibility without incurring the higher byte count of Non-Registered PlaybackResourceIDs.

A Registered PlaybackResourceID consists of one integer in VLQ form.

Note: Registered PlaybackResourceIDs may not be used for wavetable compression codecs (PlaybackResourceGroupID 1).

Example:

```
02h,           // Prefix 2 = Registered
C5h           // 197 = Registered playback resource number 197
              // (must be listed on MMA/AMEI website to be valid).
```

10.3.4.4 Non-Registered PlaybackResourceIDs

Non-Registered PlaybackResourceIDs are prefixed with ResourceTypeID 3 (0x03).

They are reserved to allow non-MMA/AMEI manufacturers to use arbitrary private playback resources without MMA/AMEI registration. To avoid collisions, a Non-Registered PlaybackResourceIDs must be generated as a Globally Unique Identifier (GUID). MMA/AMEI may also use Non-Registered PlaybackResourceIDs for specifying playback resources with GUIDs.

A Non-Registered PlaybackResourceID consists of one 16-byte GUID in binary form without any conversion (for example, to hexadecimal text form or VLQ form).

Note: Non-Registered PlaybackResourceIDs may not be used for wavetable compression codecs (PlaybackResourceGroupID 1).

Example:

```
03h,           // Prefix 3 = Non-Registered
<16 bytes appear here> // GUID in binary form
```

10.3.4.5 Wavetable Codec wFormatTag PlaybackResourceIDs

Wavetable Codec wFormatTag PlaybackResourceIDs are prefixed with ResourceTypeID 4 (0x04).

They are reserved for representing wavetable compression codecs, using the same wFormatTag values as in the Format Chunk of the Mobile DLS Wave File Format. Only MMA/AMEI may register supported Wavetable Codec wFormatTag PlaybackResourceIDs for Mobile DLS and Mobile XMF.

To promote content interoperability, the MMA will maintain a public web page listing wFormatTags in use for Mobile DLS and Mobile XMF. Developers and standards bodies adding new Mobile DLS wavetable codec wFormatTags should always contact the MMA to have their wFormatTags added to this registry, using the email address on the registry web page. The MMA will promptly add all submitted wFormatTags to the registry, so long as the codec definition does not conflict with any pre-existing definition for the submitted wFormatTag value. The wFormatTag value WAVE_FORMAT_EXTENSIBLE should not be registered, as it does not describe a specific codec. New codecs without a pre-existing wFormatTag will be registered using Wavetable Codec GUID PlaybackResourceIDs.

Each Wavetable Codec wFormatTag PlaybackResourceID consists of one VLQ for the wFormatTag value.

Example:

```
04h,           // Prefix 4 = Wavetable Codec wFormatTag
01h           // wFormatTag 1 = Linear PCM
```

10.3.4.6 Wavetable Codec GUID PlaybackResourceIDs

Wavetable Codec GUID PlaybackResourceIDs are prefixed with ResourceTypeID 5 (0x05).

They are reserved for representing wavetable compression codecs, using the same WAVE_FORMAT_EXTENSIBLE GUID values as in the Format Chunk of the Mobile DLS Wave File Format. Only MMA/AMEI may register supported Wavetable Codec GUID PlaybackResourceIDs for Mobile DLS and Mobile XMF.

To promote content interoperability, the MMA will maintain a public web page listing wavetable codec GUIDs in use for Mobile DLS and Mobile XMF. Developers and standards bodies adding new Mobile DLS wavetable codec GUIDs should always contact the MMA to have their GUIDs added to this registry, using the email address on the registry web page. The MMA will promptly add all submitted GUIDs to the registry, so long as the codec definition is not the same as any pre-existing codec definition in the registry.

A Wavetable Codec GUID PlaybackResourceID consists of one 16-byte GUID in binary form without any conversion (for example, to hexadecimal text form or VLQ form).

Example:

```
05h,           // Prefix 5 = Wavetable Codec GUID
<16 bytes appear here> // GUID in binary form
```

10.3.5 Playback Resource Group List

This is a list of entries that, together with the Playback Resource List, uniquely identify which playback resources are being described in this MetaDataItem. There is one entry per described playback resource (see 10.3.3.). Each entry consists of one VLQ containing the PlaybackResourceGroupID for the resource described at the corresponding index in the MIR Count Table entries.

Each playback resource belongs to exactly one playback resource group. A group is defined by a unique PlaybackResourceGroupID, the name of the group, whether the playback resources within the group are mutually exclusive, and whether encountering an unrecognized playback resource in the group is required to cause MIDI channel masking [6].

Mutual Exclusivity means that e.g. any one synthesizer voice resource cannot be listed as a GM voice and Mobile DLS voice at the same time, and that any one wavetable can have only one wavetable compression codec. A group's Mutual Exclusivity property should be set to 'No' if the playback resources are meant to overlap within the playback resource group, in terms of counting the MIR values.

A player shall mask (not process messages on) any MIDI channel if that channel includes any unrecognized playback resources and that resource's group has its Channel Masking property set to 'Yes', unless the channel's MIR value for the unrecognized resource is zero; in that case, the unrecognized resource should not have any effect on the channel's masking.

PlaybackResourceGroupID	Group Name	Mutual Exclusivity	Channel Masking
0	Synthesizer voice	Yes	Yes
1	Wavetable compression codec	Yes	Yes
2	Wavetable memory consumption	No	No

Mutual Exclusivity = Are playback resources mutually exclusive inside this group

Channel Masking = Unrecognized playback resources in this group cause channel masking

Example:

```
00h, 01h,    00h, 02h,    00h, 03h    // PRL for resources 1, 2, & 3
00h,        00h,         02h        // PRGL for resources 1, 2, & 3:
// 0 = Synthesizer voice
// 0 = Synthesizer voice
// 2 = Wavetable memory consumption
```

10.3.6 MIR Count Table

This is a table containing one MIR list for each MIDI channel described in this MetaDataItem. The channels appear in the order of decreasing channel priority, as defined by the corresponding MIP message. All MIR lists in a given MetaDataItem have one entry per playback resource described in this MetaDataItem, a VLQs representing the MIR value for the corresponding playback resources. The MIR counts for each resource column are cumulative. For example, the MIR counts for the second MIDI channel of the MIP message are the total maximum instantaneous resource counts for the first and second MIDI channels of the MIP message combined.

Example:

```
// MIR Count Table: 4 channel rows, 3 resource columns:
02h, 00h, 01h,    // 1st MIDI channel of the MIP message:
// 2 of 1st resource, 0 of 2nd resource, 1 of 3rd resource
03h, 00h, 01h,    // 2nd MIDI channel of the MIP message - adds 1,0,0
05h, 00h, 01h,    // 3rd MIDI channel of the MIP message - adds 2,0,0
05h, 02h, 01h     // 4th MIDI channel of the MIP message - adds 0,2,0
```

10.4 Example Content Description MetaDataItem

```
// MetaDataItem -----
// Field Specifier:
00h // Standard meta-data FieldID follows
0Dh // FieldID 13: This is a Content Description MetaDataItem
// FieldContents:
00h // Universal contents follows
19h // LengthInBytes 25: 25 bytes follow
06h // Visible binary data follow
// UniversalData starts here
00h // This MetaDataItem corresponds to
    // the first MIP message in the SMF
04h // 4 MIDI Channels in the MIP message
03h // 3 resources described in this MetaDataItem

// Resource column descriptor lists for MIR Count Table: PRL & PRGL
// PRL for resources 1, 2, & 3:
00h, 01h,    00h, 02h,    00h, 03h
    // 0 = Standard, 1 for Std = "Mobile DLS voices w/o DCF & Vib. LFO"
    // 0 = Standard, 2 for Std = "Mobile DLS voices with DCF & Vib. LFO"
    // 0 = Standard, 3 for Std = "Total wavetable memory consumption"
// PRGL for resources 1, 2, & 3:
00h,        00h,        02h
    // 0 = Synthesizer voice
    // 0 = Synthesizer voice
    // 2 = Wavetable memory consumption

// MIR Count Table: 4 channel rows, 3 resource columns:
02h, 00h, 01h// 1st MIDI channel of the MIP message:
    // 2 of 1st resource: Synth voice::Mobile DLS voices w/o DCF & Vib. LFO
    // 0 of 2nd resource: Synth voice::Mobile DLS voices with DCF & Vib. LFO
    // 1 of 3rd resource: Wavetable Memory::Total wavetable mem. consumption
03h, 00h, 01h// 2nd MIDI channel of the MIP message - adds 1,0,0
05h, 00h, 01h// 3rd MIDI channel of the MIP message - adds 2,0,0
05h, 02h, 01h// 4th MIDI channel of the MIP message - adds 0,2,0
```

11. References

- [1] “*Specification for XMF Meta File Format*”, RP-030, MIDI Manufacturers Association, Los Angeles, CA, USA, 2001
- [2] “*Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*”, RFC 2048, IETF, <http://www.ietf.org/rfc/rfc2048.txt>, November 1996
- [3] “*XMF Meta File Format Updates 1.01*”, RP-039, MIDI Manufacturers Association, Los Angeles, CA, USA, 2003
- [4] “*Type 0 and Type 1 XMF Files*”, RP-031, MIDI Manufacturers Association, Los Angeles, CA, USA, 2001
- [5] “*Standard MIDI Files*”, RP-001, in *The Complete MIDI 1.0 Detailed Specification*, Document Version 96.1, MIDI Manufacturers Association, Los Angeles, CA, USA, 1996
- [6] “*Scalable Polyphony MIDI Specification, Version 1.0*”, RP-034, MIDI Manufacturers Association, Los Angeles, CA, USA, February 2002
- [7] “*Mobile DLS Specification*”, RP-041, MIDI Manufacturers Association, Los Angeles, CA, USA, 2003
- [8] “*General MIDI System Level 1*”, RP-003, in *The Complete MIDI 1.0 Detailed Specification*, Document Version 96.1, MIDI Manufacturers Association, Los Angeles, CA, USA, 1996
- [9] “*General MIDI Level 2 Specification (Recommended Practice)*”, RP-024, November 1999, MIDI Manufacturers Association, Los Angeles, CA, USA
- [10] “*Scalable Polyphony MIDI Device 5–24 Note Profile For 3GPP, Version 1.0*”, RP-035, MIDI Manufacturers Association, Los Angeles, CA, USA, February 2002
- [11] “*XMF Meta File Format 2.00*”, RP-043, MIDI Manufacturers Association, Los Angeles, CA, USA, 2004