

MMA Technical Standards Board/ AMEI MIDI Committee

February 2001

Type 0 and Type 1 XMF Files (SMF + DLS) (RP-031)

Source: MMA TSB Item #162 (XMF Files)

Primary

Authors: MMA XMF Working Group

Abstract

This document describes the preferred simple method for bundling a collection of any number of SMF and DLS file images into a single XMF file. Player behavior is also specified. This document defines new XMF Standard ResourceFormatIDs, and Standard Meta-Data FieldIDs, amending the separate MMA document Specification for XMF Meta File Format (RP-030). Familiarity with the XMF meta-file format is assumed.

XMF Type 0 and Type 1 are identical, except that the SMF file images in an XMF Type 0 file are guaranteed to be streamable. Every XMF Type 0 and 1 File must contain at least one SMF file; inclusion of DLS files is optional. Reliable playback performance is achieved by requiring DLS and General MIDI capabilities in all players. There are no player API requirements. Any one SMF file image within the XMF file may be optionally designated to 'Autostart' when the XMF file is loaded for playback. SMF and DLS file images may optionally be marked for Preloading. Certain restrictions apply to the contents of the bundled SMF and DLS file images, and to the supported XMF meta-file features.

Note: Future Recommended Practices are expected to address the use of additional instrument formats and further resource types in XMF files. Developers are advised to check the status of this ongoing work before developing their own extensions to XMF Type 0 and 1.

Topics

1. Introduction
2. File Format
3. SMF and DLS Content Requirements
4. Playback System Specification
5. External File Type Identifiers

1. Introduction

MIDI sequences stored in SMF files and performed on a combination of custom wavetable and General MIDI instruments represent a powerful, flexible, potentially interactive, and sometimes space-efficient musical medium. This medium is suitable for a broad range of musical genres, and suitable for unattended playback in end-user terminal devices such as desktop and laptop computers, web browsers, interactive televisions, and more powerful mobile devices.

This class of MIDI application implies a link between the musical event (notes) content and the custom wavetable instruments required to present it. Because the standards defining these two media types – the Standard MIDI File (SMF) and the Downloadable Sounds file (DLS) – conceive each as a separate file, the need for a data-driven bundling and linking solution has arisen.

The MMA has previously addressed this need once, in TSB Item #157, **Recommended Use of the RMID File Format**. However, certain expansion limitations in the approach endorsed in that Recommended Practice have led to the present proposal. The XMF meta-file format used in Type 0 and Type 1 XMF Files is different from the RIFF meta-file format use in the Extended RMID format, and this difference produces several functional differences between the two Recommended Practices. As a result, some developers may find the Type 0 and Type 1 XMF file formats a more general and flexible bundling solution for SMF and DLS data.

Differences between XMF Types 0/1 and Extended RMID include:

- XMF files with equivalent contents will tend to be slightly smaller than Extended RMID.
- Whereas Extended RMID supports one SMF file image and one DLS file image, XMF Type 0 and Type 1 files may optionally contain multiple SMF file images and/or multiple DLS files, arranged by the content creator into arbitrary folder hierarchies. This makes the XMF formats suitable for both archive files and presentation files. If the Extended RMID approach were to be extended to support arbitrary hierarchies or large numbers of contained files, the size difference between the two Recommended Practices would be significantly greater.
- The XMF file may incorporate by reference external SMF and DLS files, as well as SMF file images and DLS file images contained in other XMF files, via `file:` or `http:` access. This facilitates data sharing and dynamically served, netborne files (although some players may not support these URI schemes, and will have to decline to play XMF files that require them).
- XMF meta-data also allows multiple alternate contents for each field, keyed to the end user's country and language, and optionally in Unicode encoding to support all major international character sets; Extended RMID's meta-data system uses ASCII only. We plan to define guidelines for import/export interoperability with major standard meta-data systems, whereas no such guidelines have been proposed for the system used in Extended RMID.
- XMF files may contain custom meta-data fields defined by the content creator, publisher, or distributor. While this would be possible to achieve in Extended RMID, no specific mechanism has been proposed to date.
- The playback system's required capabilities and behavior are specified for XMF Type 0 and Type 1 files, including General MIDI support and Autostart and Preload functions. In Extended RMID, no playback system requirements are specified.
- XMF Type 0 and Type 1 files have a 4 GB maximum size, and a 4 GB maximum resource size limit, whereas Extended RMID files (like all RIFF-based formats) are limited to 2 GB.

2. File Format

XMF Type 0 and XMF Type 1 files are based on the XMF Meta-File Format, with the following further requirements:

- There are certain restrictions on what `ResourceTypes` are allowed to appear in the file,
- Certain newly defined meta-data fields are required, and
- Certain newly defined optional meta-data fields are available for use

To support these requirements, the Specification for XMF Meta File Format (RP-030) is hereby extended with new `ResourceFormatIDs` and new meta-data `Standard FieldIDs`.

Topics

- 2.1. About the XMF Meta File Format
- 2.2. Definition of Type 0 and Type 1 XMF Files
- 2.3. XMF Standard `ResourceFormatID` Definitions for SMF and DLS
- 2.4. XMF Meta-Data Standard Fields for Type 0 and Type 1 Files
- 2.5. VLQ Maximum Values

2.1. About the XMF Meta File Format

XMF Type 0 and XMF Type 1 use the XMF meta-file format, which is defined in detail in the separate MMA document Specification for XMF Meta File Format (RP-030) and its amendments.

XMF is intended to support resource bundling in a low-overhead manner suitable for use in a broad range of playback platforms including very simple mobile devices (cell phones, PDAs, etc.). XMF allows multiple data blocks to be bundled into a single file, in a folder hierarchy similar to a computer file system. Each data block or folder may have any amount of meta-data associated with it, and each data block may be optionally data-compressed and optionally encrypted. Data resources in other files may be included by reference.

2.2. Definition of Type 0 and Type 1 XMF Files

XMF Type 0 and Type 1 are identical, except that the SMF file images in an XMF Type 0 file are guaranteed to be streamable. Differences between the two are described at sections 2.2.1. and 2.2.2., and common requirements are described at section 2.2.3.

Every XMF Type 0 and 1 File must contain at least one SMF file; inclusion of DLS files is optional.

2.2.1. Type 0 XMF File

XMF Type 0 is intended for SMF + DLS collections whose SMF parts can be played by streaming. Note that it is believed that for reliable playback, any needed DLS file images must be completely loaded before SMF playback begins.

A Type 0 XMF File is defined as an XMF file with the following characteristics:

- All SMF file images present in the XMF collection must be playable by streaming.
It is believed that only SMF Type 0 data that does not include embedded looping and branching messages can be streamed; SMF Type 1 file images, and SMF Type 0 file images containing embedded looping and branching messages, cannot be streamed.
- All FileNodes contain either SMF Type 0 file images, or DLS file images, or data blocks in custom formats.
No provision is made for digital audio file images (WAV, AIFF, MP3, etc.) appearing in an XMF Type 0 file, other than those appearing as components of DLS file images. It is expected that future Recommended Practices will define new XMF File Types with guidelines and behaviors for digital audio files.
- The RootNode contains one MetaDataItem (using the XMF File Type FieldID) identifying the file as a Type 0 XMF File. This is Revision 0 of Type 0, so UniversalData is **0x00 0x00**.

Further characteristics are described below at heading 2.2.3.

2.2.2. Type 1 XMF File

XMF Type 1 is intended for SMF + DLS collections, with no expectation that the SMF parts can be played by streaming. Therefore, all needed SMF and DLS file images must be completely loaded before SMF playback begins.

A Type 1 XMF File is defined as an XMF file with the following characteristics:

- All FileNodes contain either SMF Type 1 or Type 0 file images, or DLS file images, or data blocks in custom formats.
No provision is made for digital audio file images (WAV, AIFF, MP3, etc.) appearing in an XMF Type 1 file, other than those appearing as components of DLS file images. It is expected that future Recommended Practices will define new XMF File Types with guidelines and behaviors for digital audio files.
- The RootNode contains one MetaDataItem (using the XMF File Type FieldID) identifying the file as a Type 1 XMF File. This is Revision 0 of Type 1, so UniversalData is **0x01 0x00**.

Further characteristics are described below at heading 2.2.3.

2.2.3. Common Characteristics for Type 0 and Type 1

Further, both Type 0 and Type 1 XMF Files must have the following characteristics:

- Conformance to the separate Specification for XMF Meta File Format (RP-030) document.
- The contained file images may be freely arranged into files and folders however desired, to any nesting depth, as described in the Specification for XMF Meta File Format.
- Subject to the limitations described in this document, each FileNode or FolderNode in the XMF file may optionally include in its NodeMetaData any number of MetaDataItem, as defined in the Specification for XMF Meta File Format.
- As per the Specification for XMF Meta File Format, each FileNode containing resource data must indicate the resource format by including in its NodeMetaData one ResourceFormatID MetaDataItem (Standard Meta-Data field number 2), using an appropriate ResourceFormatID.
 - Note: Use of FileNodes containing data in custom formats may reduce content portability, as not all players will be able to recognize and use that data.
- As per the Specification for XMF Meta File Format, a FileNode may refer to resource data in a separate file via the file: or http: URI schemes (see ReferenceTypeIDs 4 and 5). However, for Type 0 and Type 1 XMF files only the following data file types may be referenced in this way:
 - Separate SMF files
 - SMF file images stored in other XMF files
 - Separate DLS files
 - DLS file images stored in other XMF files
 - Separate custom data files
 - Custom data file images stored in other XMF files

No provision is made for digital audio file images (WAV, AIFF, MP3, etc.) referenced from an XMF Type 0 or Type 1 file, other than those appearing as components of referenced DLS file images. It is expected that future Recommended Practices will define new XMF file Types with guidelines and behaviors for digital audio files.

Note: Use of the file: or http: URI schemes may reduce content portability, as some players may not have file systems or support web access. A player encountering an XMF file which depends on the use of a URI scheme that the player does not support must decline to play that file. If appropriate in the intended user experience, the player should report this error condition to the user (e.g. ‘External http: access not supported’ or ‘External file: access not supported’).

2.3. XMF Standard ResourceFormatID Definitions for SMF and DLS

Every FileNode in an XMF Type 0 or Type 1 file that contains an SMF or DLS file image must include in its NodeMetaData one MetaDataItem identifying its data’s resource format. The FieldSpecifier for this Meta-DataItem must be Standard FieldID 2 (Resource Format), and the FieldContents must be the appropriate Standard ResourceFormatID from the following list. No other Standard ResourceFormatIDs may appear in the XMF file (see previous notes on digital audio files).

Standard ResourceFormatIDs may only be assigned by the MMA.

ResourceFormatID	Interpretation
0	Standard MIDI File (SMF), Type 0
1	SMF, Type 1
2	Downloadable Sounds (DLS), Level 1
3	DLS, Level 2
4	DLS, Level 2.1

Note: This Recommended Practice hereby adds the preceding Standard ResourceFormatIDs to the Specification for XMF Meta File Format (RP-030), amending that document.

2.4. XMF Meta-Data Standard Fields for Type 0 and Type 1 Files

XMF Type 0 and Type 1 files introduce two new XMF Standard Meta-Data FieldIDs, as detailed in this section.

FieldID	Field Name and Notes	Valid for Node Types	Contents Format
11	Autostart Node Name of the FileNode containing the SMF image to autostart when this XMF file loads	File or Folder, but only allowed in the Root-Node	Universal Extended ASCII (hidden or visible)
12	Preload	File	Universal, No contents – just 0x00 0x00

Note: This Recommended Practice hereby adds the preceding Meta-Data Standard FieldIDs to the Specification for XMF Meta File Format (RP-030), amending that document.

2.4.1. Meta-Data FieldID 11: Autostart

The Autostart meta-data field is optional, and is intended to allow the content creator to define what ‘play’ means for an XMF file that contains multiple SMFs, by designating exactly one of the SMFs as the ‘Autostart’ target.

This field may appear in the RootNode of a Type 0 or Type 1 XMF File. This field is only valid in the RootNode; if the reading parser encounters it elsewhere, it must not process it as an autostart indicator.

The FieldContents of an Autostart MetaDataItem is a Universal Extended ASCII string containing the Node Name of one SMF file image to be automatically started when the player loads the XMF file. The desired SMF FileNode must contain one MetaDataItem with the matching Node Name (Standard FieldID 1).

Parsers should ignore any Autostart item that does not target an SMF file image.

2.4.2. Meta-Data FieldID 12: Preload

The Preload meta-data field is optional, and may be used to pre-load specific SMF and DLS file images into the player’s playback sequencer and/or DLS synth device, in order to make them immediately available for use upon a subsequent event from a user or program. For DLS this means making the instruments available in the synthesizer. For SMF this means preparing the sequence for playback, but not actually starting playback.

This field may appear in any FolderNode without restriction, and in any FileNode that describes a DLS or SMF file image. If a Preload field appears in a FolderNode, the player should preload all SMF and DLS files in that folder and in any of its descendents. If a Preload field appears in a FileNode that describes a custom data format then behavior is left to the player’s discretion, in order to preserve flexibility for custom extensions.

The FieldContents of a Preload MetaDataItem is empty, as no parameter is needed. That is, UniversalContents is two VLQs, both with value 0: 0x00 0x00.

2.5. VLQ Maximum Values

In XMF Type 0 and Type 1, VLQs shall not exceed the following maximum values. These limits are intended to guide parser implementors as to what size integer to use when reading each VLQ.

VLQ		Max Value	Min Bits Unsigned Integer
FileHeader:			
	FileLength	4 GB (4,294,967,295)	32
	TreeStart	4 GB (4,294,967,295)	32
	TreeEnd	4 GB (4,294,967,295)	32
NodeHeader:			
	NodeLength	4 GB (4,294,967,295)	32
	NodeContainedItems	65,535	16
	NodeHeaderLength	4 GB (4,294,967,295)	32
NodeMetaData:			
	LengthInBytes	4 GB (4,294,967,295)	32
	Standard FieldID	65,535	16
	Custom Field Specifier (XString length)	65,535	16
FieldContents:			
	length for UniversalContents	4 GB (4,294,967,295)	32
	NumberOfStrings for InternationalContents	65,535	16
	MetaDataType for InternationalContents	65,535	16
	VersionData length for InternationalContents	4 GB (4,294,967,295)	32
NodeUnpackers:			
	ListSizeInBytes	65,535	16
	DecodedSize	4 GB (4,294,967,295)	32
NodeContents:			
	ReferenceTypeID	5	3
MetaDataTypesTable:			
	LengthInBytes	65,535	16
	NumberOfEntries	65,535	16
	MetaDataType in TypeEntry	65,535	16
	StringFormatTypeID	7	3

3. SMF and DLS Content Requirements

The XMF Type 0 and Type 1 instrument access scheme requires slight limitations on the contents of the included SMF and DLS files, as detailed below. XMF file creation tools should enforce these rules where possible.

The General MIDI system is used to support common instruments, and the DLS system is used to support custom instruments and audio premixes including recorded performances. SMF content may be authored for GM1 or GM2, however the presence of a GM2 synth in the XMF player is not guaranteed. DLS content may be authored for DLS Level 1 or DLS Level 2, relying on DLS' fallback mechanism.

3.1. SMF Content

SMF content is unrestricted, except that:

- Every Track in an SMF will use either General MIDI instruments supplied by the playback system, or custom instruments in DLS form, and supplied via the XMF file. Each Track may independently choose either option. Within GM, both GM1 and GM2 can be used. Therefore, every SMF Track must start by specifying that choice using an XMF Patch Type Prefix Meta event (**0xFF 0x60 <len> <param>**). This SMF Meta event is defined in the separate document **SMF Meta Event for XMF Patch Type Prefix (RP-032)**.
 - The General MIDI 1 instrument set (and GM1 system behavior) is chosen by default, so no initial message is required to select GM1. The XMF Patch Type Prefix Meta event selecting GM1 (parameter 0x01) at the start of a Track is permitted, but redundant: **0xFF 0x60 0x01 0x01**.
 - If a Track has been written to take advantage of the General MIDI 2 instrument set and/or controller responses, the Track should begin with an XMF Patch Type Prefix Meta event selecting GM2 (parameter 0x02): **0xFF 0x60 0x01 0x02**.
 - If a Track has been written for the custom DLS instruments supplied via the XMF file, the Track should begin with an XMF Patch Type Prefix Meta event selecting DLS (parameter 0x03): **0xFF 0x60 0x01 0x03**.
- No SMF Track may be reassigned to a different instrument set (GM1, GM2, or DLS) at any time. Therefore, the SysEx messages **GM1 System On**, **GM2 System On**, **Turn DLS On**, **Turn DLS Off** and **GM System Off** should not appear in any SMF Track, and must be ignored by players. XMF Patch Type Prefix Meta events will only be processed if they appear as the first message in an SMF Track; if they appear anywhere else in an SMF Track, they must be ignored.

3.2. DLS Content

Any XMF file that requires any DLS instruments must supply those instruments via the XMF file.

The content of DLS file images in an XMF Type 0 or Type 1 file is unrestricted, except that to avoid presenting the player with ambiguities, no Bank/Program assignments in any DLS file image in (or referred to from) the XMF file may overlap or conflict with those of any other DLS file image in (or referred to from) the same XMF file.

Due to the custom of numbering programs from zero in DLS files, content creators using multiple DLS file images in a given XMF file should exercise care to avoid overlapping program numbers.

4. Playback System Specification

To maximize the reliability of XMF Type 0 and XMF Type 1 playback across different platforms and players, all players are required to support DLS and General MIDI. The player's expected behavior for a simple 'load-and-play' operation is defined.

4.1. Required MIDI Playback Capabilities

This sections details the MIDI Playback capabilities required for XMF Type 0 and XMF Type 1 files.

The General MIDI system is used to support common instruments, and the DLS system is used to support custom instruments and audio premixes including recorded performances. SMF content may be authored for GM 1 or GM 2. DLS content may be authored for DLS Level 1 or DLS Level 2, relying on DLS' fallback mechanism.

The implementation of any required part may be in software, hardware, or any combination thereof. There is no requirement for an Application Programming Interface (API) for any of the parts.

General MIDI player requirements are described in the Recommended Practice document **General MIDI System Level 1** and related documents. DLS player requirements are described in the Recommended Practice documents **DLS Level 1.0**, **DLS Level 2.0**, and **DLS Level 2.1**.

4.1.1. Required MIDI Capabilities for XMF Type 0 Files

To play a Type 0 XMF File, the playback system must include an SMF playback sequencer and supporting technology able to play an SMF Type 0 file image by streaming from a local file system via the URI `file:` protocol, or from a server via the URI `http:` protocol.

Streaming playback is intended as an optional feature to support quicker player start-up, not a required behavior for playing Type 1 XMF Files. The player may wait until an XMF Type 0 file is completely received before beginning playback, if desired.

Further characteristics are described below at heading 4.1.3.

4.1.2. Required MIDI Capabilities for XMF Type 1 Files

To play a Type 1 XMF File, the playback system must include an SMF playback sequencer able to play a locally queued SMF Type 0 or SMF Type 1 file image.

Streaming playback is not required.

Further characteristics are described below at heading 4.1.3.

4.1.3. Common MIDI Requirements for XMF Type 1 and Type 0

Further, to play a Type 1 or Type 0 XMF File, the playback system must have the following minimum MIDI instrument rendering capabilities:

- MIDI synth device(s) conforming to the General MIDI specifications. GM 1 is required; GM 2 is encouraged but optional. All GM Instrument data must be supplied or managed by the player, so that the XMF content can rely on the GM instruments without having to supply them directly in the XMF file. A DLS device may be used for GM rendering, at the player implementor's discretion.
- When SMF content is authored for GM2 but only a GM1 synth is available, it is recommended, but optional, for the XMF player to filter the MIDI stream appropriately, in order to prevent generation of obviously bad sounds.
- MIDI synth device(s) conforming to the DLS specifications, for custom DLS instruments supplied via the XMF file. DLS Level 1 is required; DLS Level 2 and 2.1 are encouraged but optional.
- MIDI connection between the SMF playback sequencer tracks and the synth(s).

4.2. Required Behavior

The only required behavior for an XMF Type 0 or XMF Type 1 file player is the procedure for loading and starting the file, which includes the Autostart and Preload features.

The player is not required to provide any automatic handling of, or specific behavior with regard to, any SMF and DLS file images other than those marked for Preload and Autostart. It is expected that content and software developers who wish to include multiple SMF and multiple DLS file images will have their own purposes in mind – such as archiving or interactive variation selections – and will use their own API-level commands to retrieve that content from their XMF files.

Loading an XMF Type 0 or Type 1 File and Starting Playback

Upon loading a Type 0 or Type 1 XMF file for playback, the playback system must take the following actions, in the indicated order.

1. Before loading the DLS device with any DLS file images from an XMF file, all previously loaded DLS instruments must be unloaded. This prevents incorrect instruments from being used inadvertently.
2. Load the DLS synth with all DLS files marked for Preload. (See Standard Meta-Data field number 12). For non-streaming SMFs, players may intelligently load only those instruments that the SMF calls for.
3. Load the SMF playback sequencer with all SMF file images marked for Preload or Autostart. (See section 2.4). Only one Autostart SMF is allowed.
4. Make MIDI connections between SMF tracks and synth devices, according to the initial synth assignment XMF

Patch Type Prefix Meta event in each track of the SMF. (See section 3.1.)

5. If an SMF file image was marked for Autostart in the XMF file (see section 2.4), have the playback sequencer start it. If no Autostart SMF was indicated in the XMF file the player should not start any SMF playback, but rather stand ready for the user or program to issue a Play command (via user interface or API) at any time.

For Type 0 XMF Files, the player may at its own election stream the SMF file image from its disk file or server, rather than delaying the start of playback until the entire SMF file image has been received.

6. The initial XMF Patch Type Prefix Meta events used to assign each SMF Track to a synth device should be processed only once at routing setup time, and not processed again during playback.

5. External File Type Identifiers

The preferred external identifiers for XMF Type 0 and Type 1 files are as follows:

	XMF Type 0	XMF Type 1
Filename Extension	.xmf0	.xmf1
MIME Type	audio/xmf0	audio/xmf1
Mac OS File Type	Xmf0	Xmf1

The use of filename extensions is recommended for all XMF Type 0 and Type 1 files irrespective of platform, in order to preserve external file type information should the file be transferred to a Windows computer.

On Mac OS computers the Mac OS File Types should be used.

For Internet-hosted XMF files the MIME Types should be used.

Examples

```
myStreamableXmfFile.xmf0
Unstreamable.xmf1
```

Date of issue: 11/ 29/2000 (rev 1/16/2001)

Originated by: MMA

Reference: TSBB Item #: 162 TSBB Volume #: 25 (revised)

Related Items: Specification for XMF Meta File Format (RP-030), SMF Meta Event for XMF Patch Type Prefix (RP-032), Bundling SMF and DLS data in an "RMID" file (RP-029)

CA/RP#: RP-031

Approved MMA: _____

Date: _____

Approved AMEI: _____

Date: _____