MMA Technical Standards Board/ AMEI MIDI Committee

Recommend Practice (RP-039)

XMF Meta File Format Updates 1.01

Abstract:

These 9 Updates to **XMF Meta File Format** (RP-030) and **Type 0 and Type 1 XMF Files** (RP-031) are for purposes of editorial clarification and technical enhancement, and increment the version number of each specification.

Background:

These changes are motivated by implementation experiences since the initial XMF Specifications were approved; background for each change is discussed separately in the proposal.

Contents

PO Box 3173 · La Habra CA 90632 · email: mma@midi.org

1. Updates to "Specification for XMF Meta File Format" (RP-030)

There are a total of 5 changes for the Specification for XMF Meta File Format, 2 technical and 3 editorial.

1.1 Add New ReferenceTypeID 6: XMF File URI and Node ID Number

1.1.1 Discussion

Currently the only way for an XMF file node to point to a file image stored in another XMF file is with ReferenceTypeID 5, which uses the URI of the XMF file and the node name (which must match the contents of a Node Name metadata field). In some applications it would be more efficient to identify the target node by Node ID Number metadata field, rather than by name. This new ReferenceTypeID provides a way to do this that works for both external XMF files and nodes in the same XMF file. ReferenceTypeID 6 implements this in a manner similar to ResourceTypeID 5, but with a Node ID Number instead of a Node Name. In other words, ReferenceTypeID 6 uses two parameters: URI of the XMF file in XString form, and the target node ID number in VLQ form.

1.1.2 Spec Text Changes

The text needs to be changed in two places:

- In section 2.2.1.2.1, insert new table entry for ReferenceTypeID 6:

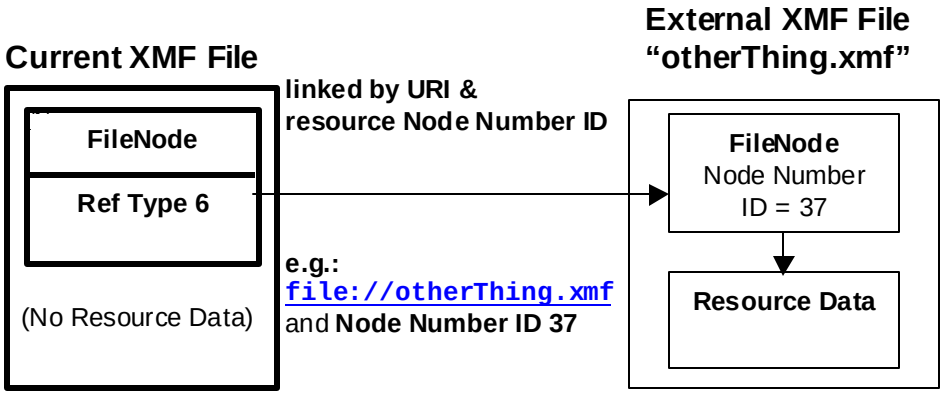
ID	Description	Description	Format
6	**XMF File URI and Node ID Number**	URI* of an external XMF file**, plus a VLQ integer specifying the target resource by Node ID Number, e.g. **`file://myOtherFile.xmf`** and Node ID Number **37** **Or** Empty URI (meaning look in the same XMF file), plus a VLQ integer specifying the target resource by node ID, e.g. **""** and Node ID Number **37**	URI: **XString** Node ID number: **VLQ**

- In section **2.2.1.2.1**, insert new description of ReferenceTypeID 6 after description of ReferenceTypeID 5:

<begin new text>

ReferenceTypeID 6: XMF File URI and Node ID

ReferenceTypeID 6 points to resources stored in (or referenced by) the same XMF file or external XMF files, identifying the file via URI (same schemes), but identifying the resource by Node Number ID, in **VLQ** form. For references to nodes in the same file, the URI will be empty (**XString** will have length **VLQ** of zero and no text characters). Use of the hash [**#**] convention in the URI is not permitted as that could conflict with the Node ID Number. For a type 6 reference to succeed, the target resource must include a Node ID Number field in its meta-data, whose contents must match the referring Node Number ID. **ReferenceTypeID** 6 has all the advantages of **ReferenceTypeID** 5, but in some applications may be more efficient – and as long as the numbers agree, the link will survive all changes to the target file's structure and other contents.



Typical Use Cases: This reference type has all the same resource sharing and aliasing benefits as ReferenceTypeID 5, but with better efficiency in some applications.

Note: In **Nodes** using **ReferenceTypeID** 6, the **NodeUnpackers** in the referenced Node override any **NodeUnpackers** in the referring node. Any **NodeUnpackers** indicated in the referring node should be ignored.

Note: The target node will not necessarily hold the target resource directly, as it has a **ContentReference** of its own. However, to avoid excessively long or circular seek chains, the target resource data must be found within 4 reference indirections, otherwise the XMF parser must fail and return a 'Too many XMF indirections' error.

<end new text>

1.1.3 Effect of Changes

Existing parsers conforming to version 1.00 of the meta file format spec will not support the new feature. Existing XMF files conforming to version 1.00 of the meta file format spec will be completely readable by parsers conforming to version 1.01.

1.2 Note Limitations of ReferenceTypeID 2

1.2.1 Discussion

For the same reasons as change 1.1, the following editorial note should be added, pointing out the issue with ReferenceTypeID 2:

1.2.2 Spec Text Changes

In section **2.2.1.2.1**, insert the following text at start of description of ReferenceTypeID 2:

"**Note: ReferenceTypeID** 2 should only be used with resource formats that include length delimiters. This is because **ReferenceTypeID** 2 does not provide any indication of the length of the target data. As a result, with non-length delimited resource formats there is no way to tell where the target resource ends. For non-inline storage of resource formats that are not length delimited, it is better to store the resource in a separate **Node** (perhaps detached from the Tree) and refer to that **Node** via **ReferenceTypeID** 3, **In-File Node**."

1.2.3 Effect of Changes

No technical effects – editorial note only.

1.3 Allow ReferenceTypeID 5 to Refer to Nodes in Same File

1.3.1 Discussion

Currently ReferenceTypeID 5 uses a URI to point to a file image stored in another XMF file, by Node Name. (Syntax is "`http://theOtherXmfFilename.xmf#theNodeName`".) This should be extended to allow the URI to refer to a file image stored in the same XMF file. The syntax is just to drop the scheme and filename parts of the URI string, e.g. just the hash and node name; for example "`#theNodeName`". As a result, the name of this ReferenceTypeID should be changed from 'External XMF Resource' to 'XMF File URI' to reflect the fact that the target node is no longer necessarily external. We should also clarify that Node Names intended to be used as targets for ReferenceTypeID 5 nodes must contain only URI-legal characters.

1.3.2 Spec Text Changes

The text needs to be changed in two places:

1. In section **2.2.1.2.1** replace table entry for ReferenceTypeID 5:

ID	Description	Description	Format
5	External XMF Resource	URI* of an external XMF file**, including specific resource name, e.g.: **`file://myOtherFile.xmf#myResource`**	XString

With this entry:

ID	Description	Description	Format
5	XMF File URI	URI* of a specific resource in an external XMF file**, e.g.: **`file://myOtherFile.xmf#myResource`** or in the same XMF file, e.g.: **`#resourceInSameXmfFile`**	XString

2. In section 2.2.1.2.1, at ReferenceTypeID 5: External XMF Resource:,

- Replace section heading text:

  "**ReferenceTypeID 5: External XMF Resource**"

  With this text: "**ReferenceTypeID 5: XMF File URI**"

- At the end of the first descriptive paragraph, insert this text:

  "To reference a node in the same XMF file, omit the scheme and filename parts of the URI string (example: '`#nodeInMySameXmfFile`')."

- At the end of the ReferenceTypeID description, insert this text:

  "**Note:** Node Names intended to be used as targets for ReferenceTypeID 5 nodes must contain only URI-legal characters."

1.3.3 Effect of Changes

Existing parsers conforming to version 1.00 of the meta file format spec will not support the new feature. Existing XMF files conforming to version 1.00 of the meta file format spec will be completely readable by parsers conforming to version 1.01.

1.4 Clarification of Byte Ordering for 16-bit Unicode Metadata

1.4.1 Discussion

The version 1.00 spec incorrectly states "It is not required to start the Unicode contents with a Byte Order Mark (`0xFEFF`), because the byte order of the entire XMF file may be determined from the first two bytes of the FileHeader structure". An editorial change is needed to clarify how to produce correct Unicode contents.

1.4.2 Spec Text Changes

In section **3.2.2.1. StringFormatTypeID Definitions**, replace this text:

> "Unicode Format -- Unicode contents for StringFormatTypeIDs 2 and 3 must conform to version 2.0 of the Unicode standard, in UTF-16 coding. It is not required to start the Unicode contents with a Byte Order Mark (`0xFEFF`), because the byte order of the entire XMF file may be determined from the first two bytes of the FileHeader structure (section 2.1)."

With this text:

> "Unicode Format -- Unicode contents for StringFormatTypeIDs 2 and 3 must conform to version 2.0 of the Unicode standard, in UTF-16 coding with big endian byte ordering. It is not required to start the Unicode contents with a Byte Order Mark (`0xFEFF`), as StringFormatTypeIDs 2 and 3 will always be in big endian byte order. It is not required to terminate the Unicode contents with a null string terminator as the Unicode contents are preceded by a VLQ. XMF Writers on little endian platforms must byte swap StringFormatTypeID 2 and 3 contents when writing XMF files. XMF Readers on little endian platforms must byte swap StringFormatTypeID 2 and 3 contents when reading XMF files."

1.4.3 Effect of Changes

There is a possibility that metadata contents for some incorrect existing XMF files will be unreadable by some XMF parsers, irrespective of version; clarifying the specification now reduces the likelihood of similar problems in the future.

1.5 Update Specification Version Number

1.5.1 Discussion

The foregoing changes necessitate an increment in the version number.

1.5.2 Spec Text Changes

The text needs to be changed in three places:

- On title page, and all page footers, replace this text: “Version 1.00”

With this text: “Version 1.01”

- 2. In section **2.1. FileHeader Structure**, replace all occurrences of this text: “Version 1.00”

With this text: “Version 1.01”

- In **Appendix: Notes on the XMF File Format**, in the detailed description of the Node structure (page 39), replace this text: “for XMF version 1.00, but”

With this text: “for XMF metafile format versions 1.00 and 1.01, but”

1.5.3 Effect of Changes

No technical effect.

2. Updates to “Type 0 & Type 1 XMF Files” (RP-031)

There are a total of 4 changes for the XMF Meta File Format Specification, 2 technical and 2 editorial.

2.1 Resolve Contradiction Regarding Preload Metadata

2.1.1 Discussion

In section **2.4. XMF Meta-Data Standard Fields for Type 0 and Type 1 Files** the table entry for the Preload metadata field incorrectly omits the fact that a Preload metadata item may be attached to certain Folder nodes, not just File nodes.

Presently the table states:

FieldID	Field Name and Notes	Valid for Node Types
12	Preload	File

However, this is contradicted by section **2.4.2. Meta-Data FieldID 12: Preload** which says "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."

Clearly "Folders" must be added to the table, but it is important to also avoid the impression that any arbitrary file can be marked for Preload.

2.1.2 Spec Text Changes

In section **2.4. XMF Meta-Data Standard Fields for Type 0 and Type 1 Files**, change the table entry for the Preload metadata item from this:

FieldID	Field Name and Notes	Valid for Node Types
12	Preload	File

To this:

FieldID	Field Name and Notes	Valid for Node Types
12	Preload	File or Folder (see section 2.4.2.)

2.1.3 Effect of Changes

Existing parsers conforming to revision 0 of the Type 0 & Type 1 spec may be incorrectly failing to apply inheritance for the Preload metadata field for Type 0 & Type 1 files containing Folder nodes with the Preload metadata item. Clarifying the spec now reduces the likelihood of similar problems in the future. Existing parsers correctly conforming to revision 0 of the Type 0 & Type 1 spec will properly handle all new Type 0 & Type 1 files conforming to revision 1.

2.2 Reflect New XMF Meta File Format Specification Version

2.2.1 Discussion

In section **2. File Format**, use of the **Specification for XMF Meta File Format** is required without mentioning which version should be used. It should be clarified that either v1.00 or v1.01 may be used.

2.2.2 Spec Text Changes

In section **2. File Format**, insert the following text just before section 2.1. About the XMF File format:

> “Note: Either v1.00 or v1.01 of the **Specification for XMF Meta File Format** may be used.”

2.2.3 Effect of Changes

Parsers conforming to revision 0 of the Type 0 & Type 1 spec will not be able to handle any of the new features in version 1.01 of the **Specification for XMF Meta File Format**, including ReferenceType 6. However these features only appear in revision 1 of the Type 0 & Type 1 spec, so revision 0 parsers are able to detect and reject revision 1 files. All revision 1 parsers will be able to handle all revision 0 files.

2.3 Reflect New ReferenceTypeID 6

2.3.1 Discussion

In section **2.5. VLQ Maximum Values** the table entry for the ReferenceTypeID field of the NodeContents structure states the maximum allowable value is 5, but with the introduction of ReferenceTypeID 6 in the **Specification for XMF Meta File Format v1.01**, this must be increased to 6.

2.3.2 Spec Text Changes

In section 2.5. VLQ Maximum Values, change the table entry for ReferenceTypeID from this:

VLQ	Max Value	Min Bits Unsigned Integer
ReferenceTypeID	5	3

To this:

VLQ	Max Value	Min Bits Unsigned Integer
ReferenceTypeID	6	3

2.3.3 Effect of Changes

No technical change.

2.4 Update Specification Version Number

2.4.1 Discussion

The foregoing changes necessitate an increment in the spec version number, which is reflected in the RevisionID field of the XMF File Type metadata item.

2.4.2 Spec Text Changes

Before the [Abstract], insert this text: "Revision 1 (RevisionID = 0x01)"

2.4.3 Effect of Changes

No technical effect.