

MXFWrapper

Specification Sheet

Version 1.4
June 2004



This information is subject to change without prior notice.

Uses the MXF SDK developed in co-operation with Institut für Rundfunktechnik GmbH, www.irt.de.

Institut für Rundfunktechnik The logo for Institut für Rundfunktechnik (IRT) consists of the letters 'IRT' in a bold, blue, sans-serif font. The letter 'I' is red, 'R' is green, and 'T' is blue. The text 'Institut für Rundfunktechnik' is written in a blue, sans-serif font and is underlined with a blue dashed line.

Table of Contents

Overview.....	5
ActiveX Interface Specification.....	7
Methods for both MXFWrapper versions (with and without DM).....	7
<i>wrap</i>	7
<i>reset</i>	7
<i>cancel</i>	8
<i>setProperty</i>	8
<i>addMuxEssence</i>	9
<i>generateSystemItem</i>	10
Additional Methods for the MXFWrapper without DM option.....	11
<i>addEssence</i>	11
<i>addMux</i>	12
<i>setIdentication</i>	13
Additional Methods for the MXFWrapper with DM option	14
<i>setMetadata</i>	14
<i>addEssence</i>	15
<i>addMux</i>	16
Events.....	16
<i>progressUpdate</i>	17
<i>done</i>	17
<i>error</i>	17
Properties	17
<i>dictionaryFile</i>	18
<i>streamable</i>	18
<i>masterEditUnitsPerPartition</i>	18
<i>closeAllPartitions</i>	19
<i>footerHasMetadata</i>	19
<i>metadataRepetitionPeriod</i>	20
<i>kagSize</i>	20
<i>writeRIP</i>	21
<i>footerHasIndex</i>	21
Annex I - Essence Properties.....	22
Annex II - Error Codes	23
References	24

Overview

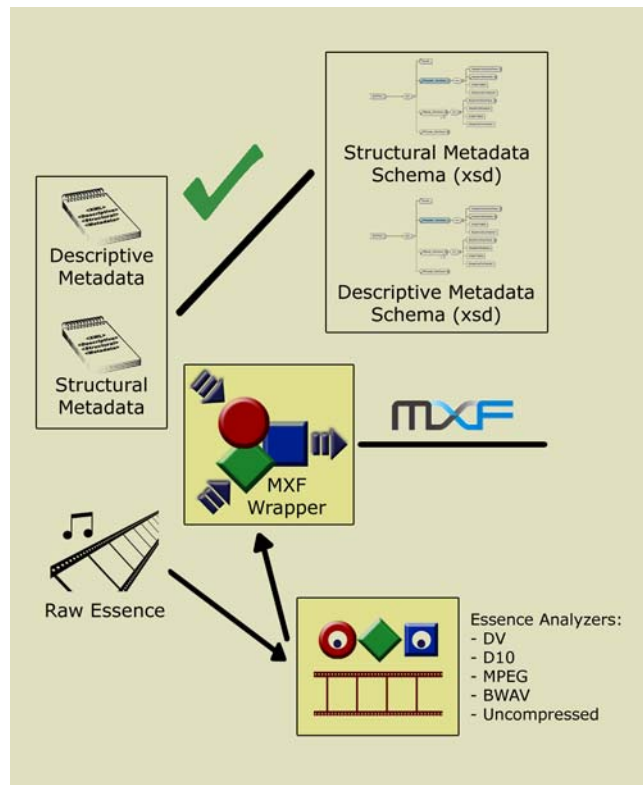
The **MXFWrapper** is a binary component for Microsoft Windows® that creates MXF [1] files, using Essence and/or Metadata XML documents as its input.

The **MXFWrapper** is available either as an ActiveX® component or as a C++ library. It uses a dictionary to interpret the Universal Labels in the MXF file. This dictionary can be a XML Schema document, as defined in [2], or a binary version of the Schema document allowing increase of performance at start-up. The same dictionary is used to interpret Descriptive or Structural Metadata passed in XML format to the **MXFWrapper**.

The **MXFWrapper** is distributed with two options:

- The first and the simplest uses the ActiveX interface to generate the Structural Metadata, adding Essence via method calls;
- The second, uses an input XML document that describes the whole Metadata, both Descriptive and Structural.

The programmatic method targets applications that require only the simplest Operational Patterns, from OP1x to OP2x and OP3a, including subsets of OP3b and OP3c. The XML-based method allows the creation of files of any complexity.



Currently, and due to the semantic richness of MXF Structural Metadata, Descriptive Metadata can only be added using the latter method.

The MXF standard is extremely flexible, making it possible to create compliant files in a variety of ways. The **MXFWrapper** allows users specify a number of properties that define the way files will be created, such as the repetition rate of Header Metadata, the number of Partitions, the streamability of the file, etc.

The **MXFWrapper** is Essence agnostic, relying in **Essence Analyzers** [3] to extract the part of the MXF Structural Metadata that can be automatically generated.

The next section details the ActiveX interface. The interface documentation for the C++ version of the Components is shipped in a separate document [6].

ActiveX Interface Specification

Methods for both MXFWrapper versions (with and without DM)

wrap

Method	
Generates the MXF file, according to the properties defined below. Returns instantly if asynchronous set to 1 otherwise method will block until MXF wrapping has been finalized.	
Return	
HRESULT	S_OK
Parameters	
BSTR outputFile	The path to the MXF file that will be generated.
SHORT asynchronous	If 1 then client will receive progress events and 0 for blocking the method to return.
[out] SHORT retValue	Result code. 1 if something failed and 0 (zero) if Ok.

reset

Method	
Resets all properties to its default value.	
Return	
HRESULT	S_OK

cancel

Method	
Aborts the wrapping process.	
Return	
HRESULT	S_OK

setProperty

Method	
Sets the properties for Identification Essence Analyzer.	
Return	
HRESULT	S_OK
Parameters	
ULONG id	The identifier of the Essence to which this property must be associated. This ID is obtained from the addEssence method.
BSTR name	See Annex 1 - Essence Properties Table.
BSTR value	See Annex 1 - Essence Properties Table .

addMuxEssence

Method	
Add an Essence stream to be multiplexed. The multiplexer is created with the addMux method.	
Return	
HRESULT	FALSE if essence type is not of type defined by the essenceTypeUL or if the essence contains invalid data. S_OK if success
Parameters	
BSTR muxID	The identifier of the Multiplexer to which this Essence will be added. This ID is obtained from the addMux method.
BSTR file	The path to the file containing the Essence.
BSTR essenceTypeUL	The Universal Label that defines the type of Essence, as defined in [4].
[out] ULONG id	An identifier of the Essence to be added to be used in other methods of this interface whenever you need to refer to this Essence again

generateSystemItem

Method	
Calling this method causes the MXFWrapper to insert a System Item according to SMPTE S386M	
Return	
HRESULT	S_OK
Parameters	
ULONG id	The identifier of the multiplexer obtained from addMux.
SHORT hours	The hours component of the initial timecode
SHORT minutes	The minutes component of the initial timecode
SHORT seconds	The seconds component of the initial timecode
SHORT frames	The frames component of the initial timecode

Additional Methods for the MXFWrapper without DM option

addEssence

Method	
Adds an Essence item.	
Return	
HRESULT	FALSE if essence type is not of type defined by the essenceTypeUL or if the essence contains invalid data. S_OK if success.
Parameters	
BSTR file	The path to the file containing the Essence.
BSTR essenceTypeUL	The Universal Label that defines the type of Essence, as defined in [4].
BSTR essenceLocator	If empty, the MXF file will have this Essence stored internally. If not empty, the Essence will not be wrapped with the MXF file but this locator will be inserted in the Structural Metadata.
[out] ULONG id	An identifier of the Essence to be added to be used in other methods of this interface whenever you need to refer to this Essence again

addMux

Method	
Create a Multiplexer inside the MXFWrapper that will multiplex in the same Essence Container several Essence streams. A feature also known in MXF as Essence Interleaving. To add Essence streams to the Multiplexer use the addMuxEssence method.	
Return	
HRESULT	S_OK
Parameters	
BSTR muxEssenceUL	The Universal Label that defines the type of Essence, as defined in [4].
BSTR muxEssenceLocator	If empty, the MXF file will have the multiplexed Essence streams (associated to this Multiplexer) stored internally in the same Essence Container. If not empty, the Essence will not be wrapped with the MXF file but this locator will be inserted in the Structural Metadata.
[out] ULONG id	An identifier of the create Multiplexer to be used in other methods of this interface whenever you need to refer to this Multiplexer again. E.g: the addMuxEssence method.

setIdentification

Method	
Sets the values of the Identification Metadata Set.	
Return	
HRESULT	S_OK
Parameters	
BSTR companyName	See [1], §A.2.
BSTR productName	See [1], §A.2.
SHORT productVersionMajor	See [1], §A.2.
SHORT productVersionMinor	See [1], §A.2.
SHORT productVersionPatch	See [1], §A.2.
SHORT productVersionBuild	See [1], §A.2.
SHORT productVersionRelease	See [1], §A.2.
BSTR versionString	See [1], §A.2.
BSTR productUID	See [1], §A.2.

Additional Methods for the MXFWrapper with DM option

setMetadata

Method	
<p>Sets the Metadata that is going to be used to generate the file. The format is defined in [2].</p> <p>If this method is called, the properties that define the way the MXF file is created will be ignored since this Metadata already contains the Structural Metadata.</p>	
Return	
HRESULT	S_OK
Parameters	
BSTR metadata	<p>The XML document containing the Structural and possibly Descriptive Metadata that will be used to generate the file.</p> <p>Note that this is a pointer to a buffer holding the XML data in text format, not a filename.</p>

addEssence

Method	
Adds an Essence item.	
Return	
HRESULT	FALSE if essence type is not of type defined by the essenceTypeUL or if the essence contains invalid data.
Parameters	
BSTR file	The path to the file containing the Essence.
BSTR essenceTypeUL	The Universal Label that defines the type of Essence, as defined in [4].
BSTR umid	String containing the UMID to be used when wrapping the essence for the SourcePackage. This allows to link to the XML document containing the descriptive metadata.
BSTR essenceLocator	If empty, the MXF file will have this Essence stored internally. If not empty, the Essence will not be wrapped with the MXF file but this locator will be inserted in the Structural Metadata.
[out] ULONG id	An identifier of the Essence to be added to be used in other methods of this interface whenever you need to refer to this Essence again

addMux

Method	
<p>Create a Multiplexer inside the MXFWrapper that will multiplex in the same Essence Container several Essence streams. A feature also known in MXF as Essence Interleaving. To add Essence streams to the Multiplexer use the addMuxEssence method.</p>	
Return	
HRESULT	S_OK
Parameters	
BSTR muxEssenceUL	The Universal Label that defines the type of Essence, as defined in [4].
BSTR umid	String containing the SourcePackage UMID to be used when wrapping the Essence. This allows to link to the XML document containing the descriptive metadata.
BSTR muxEssenceLocator	If empty, the MXF file will have the multiplexed Essence streams (associated to this Multiplexer) stored internally in the same Essence Container. If not empty, the Essence will not be wrapped with the MXF file but this locator will be inserted in the Structural Metadata.
[out] ULONG id	An identifier of the create Multiplexer to be used in other methods of this interface whenever you need to refer to this Multiplexer again. E.g: the addMuxEssence method.

Events

progressUpdate

Event	
Tells client application the percentage of completion while wrapping the MXF file. If asynchronous flag of the wrap function is set.	
Parameters	
SHORT percent	Percentage of wrapping process. Values range from 0 to 100.

done

Event
Event raised when the wrapping process has finished. If asynchronous flag of the wrap function is set.

error

event	
Event raised every time an exception occurs.	
Parameters	
SHORT errorCode	The exception error code, please refer to Annex II
BSTR message	The exception error message.

Properties

dictionaryFile

Property	
The path to the binary form of the MXF dictionary (e.g. mxf_file.bin). See [2].	
Type	BSTR
Default	""

Property	
If true, the partitions including Essence of different Essence items will be interleaved, allowing the decoding of the file while being received. This option only has meaning when wrapping more than one internal essence (therefore only in Operational Patterns higher than 1A). For details, see [5].	
Type	SHORT
Default	False

streamable

masterEditUnitsPerPartition

Property	
The number of Edit Units of the master Essence item per partition. The master Essence item is the first item added using addEssence or addEssenceClip . If 0 all Edit Units of one Essence item will be placed in the same partition.	
Type	ULONG
Default	0

closeAllPartitions

Property	
<p>If true, all Partitions will be closed. If false, the Structural Metadata in each partition will contain only the values of the Essence processed so far¹.</p> <p>While closing partitions all header metadata contained in the Partitions will be rewritten to allow closing of that partition. Please note that MXF mandates that a closed partition containing metadata must be identical in all closed partitions. For more information refer to [1].</p>	
Type	SHORT
Default	False

footerHasMetadata

Property	
<p>If true, the Header Metadata will be included in the Footer.</p>	
Type	SHORT
Default	True

¹ The latter is more efficient because closing all partitions requires pre-processing all Essence.

metadataRepetitionPeriod

Property	
The number of partitions between two repetitions of the Header Metadata. If 0, no Metadata will be inserted in Body Partitions.	
Type	SHORT
Default	0

kagSize

Property	
The KLV Alignment Grid size, as defined in [1], §5.4.1. Depending on the Essence Container type this value will be ignored for the partitions that require specific KAG sizes.	
Type	ULONG
Default	1

writeRIP

Property	
If true, the Random Index Pack will be included in the Footer.	
Type	SHORT
Default	True

footerHasIndex

Property	
If true, the footer partition will have Index Table information.	
Type	SHORT
Default	True

Annex I - Essence Properties

Essence Properties Table

Name	Value
"externallyStored"	if we want external essence, with the value "1" otherwise "0"
"CBE"	if we want fixed size picture item, with the value "1" otherwise "0"

Annex II - Error Codes

Error Codes	
Code	Description
0	Unknown Exception.
1 - 1023	Internal error (please report this error code to support@mog-solutions.com).
1024	General error with an error message.
1025	Error processing the metadata, see message for details.
1026	Error loading EssenceAnalyzer DLL. Please check your installation.
1027	Error when trying to parse Essence. Please make sure that the Essence format is the one specified in the Essence Type UniversalLabel.

References

- [1] SMPTE 377M, *Material Exchange Format (MXF) File Format Specification (Standard)*, 2003.
- [2] MOG Solutions, *XML Schema for MXF Metadata*, 2003.
- [3] MOG Solutions, *Essence Analyzers Specification Sheet*, 2003.
- [4] SMPTE EG43, *Material Exchange Format (MXF) Engineering Guideline (Informative)*, 2003.
- [5] SMPTE EG41, “*Material Exchange Format (MXF) File Format Specification*”, 2003.
- [6] MOG Solutions, “*MXFComponentSuite C++ API Reference*”.
- [7] SMPTE RP224, “*Registry of SMPTE Universal Labels*”