

SMPTE STANDARD**Material Exchange Format —
XML Encoding for Metadata
and File Structure Information**

Page 1 of 38 pages

Table of Contents	Page
Foreword	3
Intellectual Property	3
1 Scope	4
2 Conformance Notation	4
3 Normative References	4
4 Definition of Terms	6
5 General Considerations (Informative)	6
5.1 XML-Schema	6
5.2 Application Scenario (Information)	7
5.3 Relation to Architecture of the MXF Documentation	9
6 Conformance to MXF Specifications	9
7 XML Encoding Rules	9
7.1 General	9
7.2 Namespaces	9
7.2.1 Namespaces for class 13 and class 14 entries (informative)	13
7.3 General Encoding Rules	14
7.4 Data Types	14
7.4.1 XML Encoding	14
7.4.2 MXF-XML-Schema Extension	15
7.5 Dictionary Elements	15
7.5.1 XML Encoding	15
7.5.2 MXF-XML-Schema Extensions	16
7.6 Sets and Packs	19
7.6.1 XML Encoding	19
7.6.2 MXF-XML-Schema Extensions	21
7.7 References	25
7.7.1 XML Encoding	25
7.7.2 MXF-XML-Schema Extensions	27
7.8 Dark Metadata	29
7.8.1 General	29
7.8.2 XML Encoding	30
7.8.3 Reference Resolution	30
7.9 Partiton Multiplex and Index Table Information	31

Annex A MXF-XML-Schema Definition (Normative) 32

Annex B Steps to Extend the Encoding (Informative) 33

 B.1 General..... 33

 B.2 Adding Properties to Sets 33

 B.3 Adding Sets and Packs 33

 B.4 Adding Metadata Schemes 34

Annex C Examples (Informative)..... 35

 C.1 MXF-XML Documents Conformant with Annex A..... 35

 C.1.1 Metadata Only - Physical Model 35

 C.1.2 Metadata Only - Logical Model 35

 C.1.3 Partition Multiplex, Index Table and Metadata - Physical Model 35

 C.1.4 Metadata Only - Logical Model - DarkProperty..... 35

 C.1.5 Metadata Only - Logical Model - UnparsedDarkGroup 35

 C.1.6 Metadata Only - Logical Model - ParsedDarkSet 35

 C.2 MXF-XML Encoding Extensions Examples 35

 C.2.1 Schema Extension 35

 C.2.2 MXF-XML Document Conformant with Extended Schema 37

Annex D Bibliography (Informative)..... 38

Foreword

SMPTE (the Society of Motion Picture and Television Engineers) is an internationally-recognized standards developing organization. Headquartered and incorporated in the United States of America, SMPTE has members in over 80 countries on six continents. SMPTE's Engineering Documents, including Standards, Recommended Practices and Engineering Guidelines, are prepared by SMPTE's Technology Committees. Participation in these Committees is open to all with a bona fide interest in their work. SMPTE cooperates closely with other standards-developing organizations, including ISO, IEC and ITU.

SMPTE Engineering Documents are drafted in accordance with the rules given in its Standards Operations Manual.

SMPTE ST 434 was prepared by Technology Committee 31FS.

Intellectual Property

At the time of publication no notice had been received by SMPTE claiming patent rights essential to the implementation of this Standard. Attention is drawn, however, to the possibility that some of the elements of this document may be the subject of patent rights. SMPTE shall not be held responsible for identifying any or all such patent rights.

1 Scope

This standard specifies an encoding of the header metadata, partition multiplex structure and index tables of MXF files as XML documents. It specifies methods to support descriptive metadata plug-ins, dark metadata and private and future public extensions to the MXF specification.

This standard applies to the lossless representation of the metadata contained in MXF files as XML documents.

This standard applies to the interchange of the information of the header metadata, partition multiplex structure and index tables of MXF files as XML documents where the receiving application supports all XML namespaces from which there are elements in the XML document. The standard does not provide for the XML coding of essence.

2 Conformance Notation

Normative text is text that describes elements of the design that are indispensable or contains the conformance language keywords: "shall", "should", or "may". Informative text is text that is potentially helpful to the user, but not indispensable, and can be removed, changed, or added editorially without affecting interoperability. Informative text does not contain any conformance keywords.

All text in this document is, by default, normative, except: the Introduction, any section explicitly labeled as "Informative" or individual paragraphs that start with "Note:"

The keywords "shall" and "shall not" indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.

The keywords, "should" and "should not" indicate that, among several possibilities, one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.

The keywords "may" and "need not" indicate courses of action permissible within the limits of the document.

The keyword "reserved" indicates a provision that is not defined at this time, shall not be used, and may be defined in the future. The keyword "forbidden" indicates "reserved" and in addition indicates that the provision will never be defined in the future.

A conformant implementation according to this document is one that includes all mandatory provisions ("shall") and, if implemented, all recommended provisions ("should") as described. A conformant implementation need not implement optional provisions ("may") and need not implement them as described.

Unless otherwise specified, the order of precedence of the types of normative information in this document shall be as follows: Definitions inside the XML schema files shall be the authoritative definition; Normative prose shall be next; Tables shall be next; followed by formal languages; then figures; and then any other language forms.

3 Normative References

Note: All references in this document to other SMPTE documents use the current numbering style (e.g. SMPTE ST 380:2004) although, during a transitional phase, the document as published (printed or PDF) may bear an older designation (such as SMPTE 380M-2004). Documents with the same root number (e.g. 380) and publication year (e.g. 2004) are functionally identical.

The following standards contain provisions which, through reference in this text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision,

and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent edition of the standards indicated below.

SMPTE ST 335:2012, Metadata Element Dictionary Structure

SMPTE ST 336:2007, Data Encoding Protocol Using Key-Length-Value

SMPTE ST 377-1:2011, Material Exchange Format (MXF) — File Format Specification

Amendment 1:2012 to SMPTE ST 377-1:2011

Amendment 2:2012 to SMPTE ST 377-1:2011

SMPTE ST 377-4:2012, MXF Multichannel Audio Labeling Framework

SMPTE ST 380:2004, Material Exchange Format (MXF) — Descriptive Metadata Scheme-1

SMPTE ST 381-1:2005, Material Exchange Format (MXF) — Mapping MPEG Streams into the MXF Generic Container

SMPTE ST 381-2:2011, Material Exchange Format (MXF) — Mapping MPEG Streams into the MXF Constrained Generic Container

SMPTE ST 382:2007, Material Exchange Format — Mapping AES3 and Broadcast Wave Audio into the MXF Generic Container

Amendment 1:2012 to SMPTE ST 382:2007

Amendment 2:2013 to SMPTE ST 382:2007

SMPTE ST 385:2012, Material Exchange Format (MXF) — Mapping SDTI-CP Essence and Metadata into the MXF Generic Container

SMPTE ST 410:2008, Material Exchange Format — Generic Stream Partition

SMPTE ST 422:2014, Material Exchange Format — Mapping JPEG 2000 Codestreams into the MXF Generic Container

SMPTE ST 429-5:2009, D-Cinema Packaging — Timed Text Track File

SMPTE ST 429-6:2006, D-Cinema Packaging — MXF Track File Essence Encryption

SMPTE ST 429-10:2008, D-Cinema Packaging — Stereoscopic Picture Track File

SMPTE ST 436-1:2013, MXF Mappings for VBI Lines and Ancillary Data Packets

SMPTE ST 2029:2009, Uniform Resource Names for SMPTE Resources

SMPTE ST 2037:2009, Mapping VC-1 into the MXF Generic Container

SMPTE RP 2057:2011, Text-Based Metadata Carriage in MXF

SMPTE ST 2075:2013, Mapping EBU TECH 3264 (STL) into the MXF Generic Stream Container

Extensible Markup Language (XML) 1.0, W3C Recommendation, 2008

Namespaces in XML 1.0, W3C Recommendation, 2009

XML Schema Part 1: Structures, W3C Recommendation, 2004

XML Schema Part 2: Datatypes, W3C Recommendation, 2004

4 Definition of Terms

For the purposes of this document, the following terms and definitions apply.

4.1

MXF-File

A file compliant with the MXF-Specification-Suite.

4.2

MXF-Specification-Suite

SMPTE ST 377-1, its amendments and corrigenda and the set of SMPTE Engineering Documents that define metadata sets that are intended for interchange within the framework of SMPTE ST 377-1 header metadata.

4.3

MXF-to-XML-Encoder

A process that receives an MXF-File at its input and produces an MXF-XML-Document at its output.

4.4

MXF-XML-Document

An XML document that satisfies the provisions of this specification. MXF-XML-Documents can either represent the MXF-File's header metadata or the MXF-File's partition multiplex structure including header metadata and index tables, but not essence container data.

4.5

MXF-XML-Schema

The set of XML Schema files incorporated in this specification that defines the format and syntax of MXF-XML-Documents or a version of this set of XML Schema files that has been modified in accordance with the provisions of this specification.

4.6

XML-to-MXF-Encoder

A process that receives an MXF-XML-Document at its input and produces an MXF-File at its output.

5 General Considerations (Informative)

5.1 XML-Schema (Informative)

An XML encoding can be specified by defining the format and syntax of XML instance documents, and by defining further constraints on their content. This specification uses a MXF-XML-Schema written in XML Schema to define the format and syntax of MXF-XML-Documents. Additional constraints on the contents of the MXF-XML-Documents result from the provisions of the documents of the MXF-Specification-Suite.

This specification also contains provisions how to extend the MXF-XML-Schema to support additions to the MXF-Specification-Suite, user-defined extensions of metadata sets defined in documents of the MXF-Specification-Suite and user-defined metadata schemes.

The MXF-XML-Schema does not need to be an element of an MXF-XML implementation. While the MXF-XML-Schema can be used to validate the format and syntax of MXF-XML-Document, it cannot be used to validate the contents of MXF-XML-Documents. In particular, the MXF-XML-Schema does not represent all of

the rules and constraints of the MXF-Specification-Suite and, for example, does not validate if the MXF operational pattern properties of the MXF Partition packs and of the MXF Preface set do contain the same value. According to SMPTE ST 377-1 this is required.

5.2 Application Scenario (Informative)

The MXF-XML application scenario is represented in Figure 1.

The lower section of the figure shows the MXF domain. In this domain MXF files are interchanged between MXF aware applications. The content of the MXF files is multiplexed into partitions according to the rules of SMPTE ST 377-1. They contain MXF header metadata and can contain essence. The data structures and properties of the files conform to the provisions of the relevant documents of the MXF-Specification-Suite. The MXF aware applications can use different metadata schemes. Examples for such schemes are MXF structural metadata, MXF structural metadata in combination with DMS-1 or another descriptive scheme or MXF structural header metadata with proprietary extensions. The independent extensibility supported by the design of MXF ensures that each of the MXF aware applications can successfully decode the information that is supported by its scheme, while being able to ignore and pass on the information that lies outside its supported scheme.

The structure of an MXF file and the MXF header metadata contained within it can be represented in XML through the use of an MXF-to-XML-Encoder that produces MXF-XML-Documents. MXF-XML-Documents need to be compliant with provisions of the relevant documents of the MXF-Specification-Suite as well as with the provisions of this specification. MXF-XML-Documents do not contain essence container data.

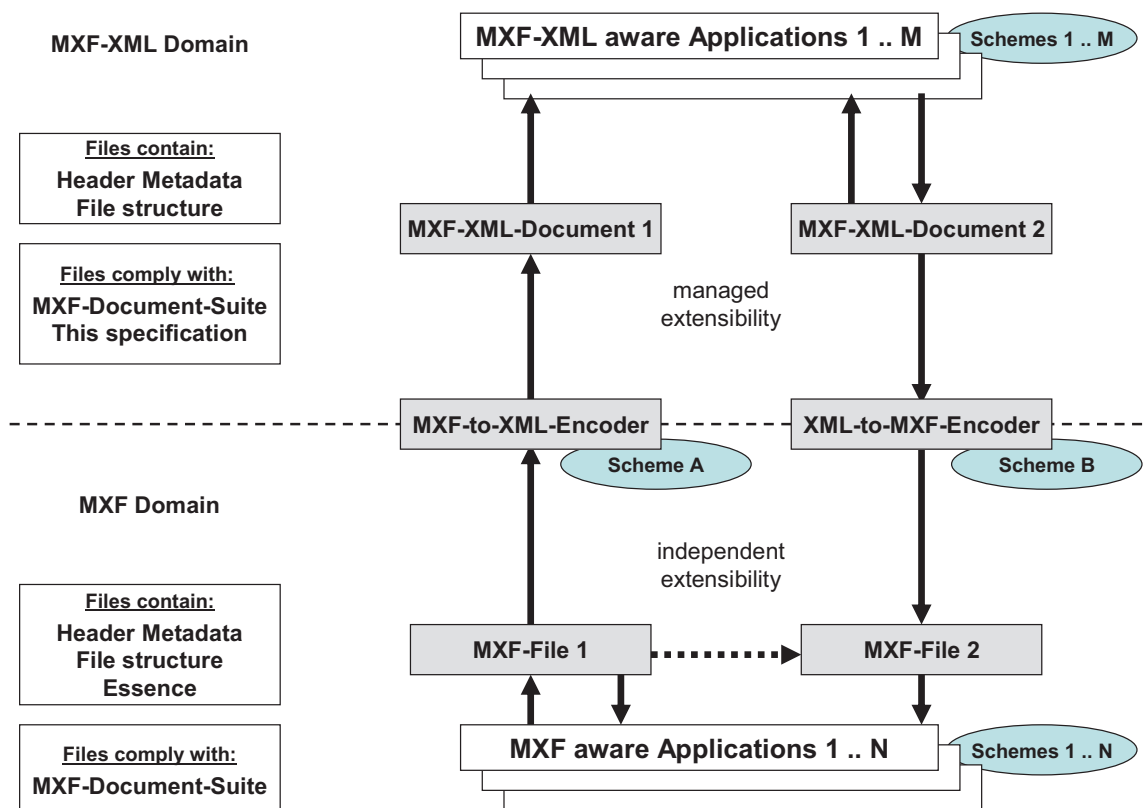


Figure 1 – Application Scenario

MXF aware applications can decode the KLV stream, process the MXF partition multiplex information and are able to decode MXF header metadata contained in the MXF file that is supported by their scheme. They might also be able to generate and/or update MXF files.

The MXF-to-XML-Encoder decodes the partition multiplex information as well as MXF header metadata and index table information contained in the MXF file that is supported by its scheme, and encodes all or a subset of it into MXF-XML. All MXF header metadata or index table information that is not supported by the scheme of the MXF-to-XML-Encoder can be encoded lossless into XML using the mechanisms of the MXF-XML specification to represent dark metadata.

MXF-XML aware applications read existing MXF-XML-Documents, update existing one or produce entirely new MXF-XML-Documents.

XML-to-MXF-Encoders can take MXF-XML-Documents as input and produce and/or update MXF files.

The MXF-XML-Schema in Annex A and its extensions that satisfy the rules defined in Section 5 do not represent a complete validation tool for MXF-XML-Documents. However, the validation of an MXF-XML-Documents by the MXF-XML-Schema used by an MXF-XML aware application constitutes a necessary condition that an application that supports all namespaces from which there are elements in the MXF-XML-Documents can successfully read the MXF-XML-Documents and that an XML-to-MXF-Encoder can produce an MXF file that contains a lossless representation of the MXF header metadata contained in the MXF-XML-Documents.

MXF-XML does not aim to be an interchange format that supports independent extensibility. This problem has already been solved by the encoding mechanisms of SMPTE ST 377-1. In environments that require independent extensible interchange, it is therefore recommended to rely on the provisions of the MXF-Specification-Suite.

Within one MXF-XML installation domain, extensibility can be supported by managing the schemes used by the MXF-XML aware applications. Information that is supported or needed only by one of the MXF-XML aware applications can be encoded into MXF-XML-Documents using the dark metadata mechanisms of this specification. All MXF-XML aware applications will be able to ignore this information (that is irrelevant to them) and can forward it losslessly. MXF-XML aware applications can try to parse the KLV data carried in *ParsedDarkSet* XML elements, *UnparsedDarkGroup* XML elements and *DarkProperty* XML elements to detect and decode usable metadata.

Information that is required by more than one application is to be represented at the XML level in decoded form (as opposed to dark). This information also needs to be supported by the schemes of the relevant applications of the installation domain. Figure 1 emphasizes this management requirement by introducing the term “managed extensibility” in the MXF-XML part of the application scenario.

Wherever possible and desirable, systems are to be designed such that all known MXF extensions in use within the system are contained in schemes available to each MXF-to-XML Encoder and to each XML-to-MXF Encoder.

In Figure 1, MXF-to-XML-Encoders and XML-to-MXF-Encoders are drawn to lie in the MXF domain as well as in the MXF-XML domain. They are both XML aware and MXF-XML aware.

While MXF-XML-Documents do not contain essence container data, they can be used by MXF-XML aware applications to access essence container data stored in an MXF file. In particular, if the MXF-XML document contains the partition multiplex information and index table information, an MXF-XML aware application does not need to decode any of the header metadata or multiplex information from the MXF file in order to access specific essence container data elements.

MXF-XML-Documents can also be used as elements of a representation that contains information of an equivalent MXF file, including the essence. If the equivalent MXF contains index table information, the MXF-XML-Document needs to contain the partition multiplex information and index table information. The essence container data for all essence containers is extracted into external files and network locators are added to the descriptors of the File Packages to link each of the files containing the extracted essence container data to the package that was associated with the essence container. If the original file does not contain index table information (or if it is not required to preserve index table information that can be present in the original file), the partition multiplex information can be discarded and it is sufficient if the MXF-XML-Document contains only the header metadata.

In combination with an MXF-to-XML encoder and an XML-to-MXF encoder, MXF-XML aware applications can also be used to update the header metadata of an MXF file while preserving the essence containers and index tables of the original file.

In Figure 1, the functionality of the previous two application examples is indicated by the dotted arrow pointing from MXF File 1 to MXF File 2.

5.3 Relation to Architecture of the MXF Documentation

The specification of the XML encoding follows the modularity of the architecture of the MXF-Specification-Suite. Namely, it has a core and supports plug-ins.

The core defines the XML representation for the MXF structural metadata sets defined in SMPTE ST 377-1, partition multiplex structure and index tables. The plug-in mechanism allows the extension of the XML encoding by adding new properties to existing metadata sets, new metadata sets to existing metadata schemes and new metadata schemes. Some of these plug-ins are already included in this specification.

In the MXF-Specification-Suite, a new scheme is defined by adding one or more new documents to the suite. Examples are descriptive metadata schemes and essence mapping documents. In the MXF-XML-Schema there shall be one file and one unique namespace that contain the definitions of the metadata sets for each such document.

Annex A consists of a description of the directory structure of the MXF-XML-Schema documents and of the MXF-XML-Schema documents which constitute a part of this specification.

6 Conformance to MXF Specifications

The property values and property hierarchy of the data encoded in MXF-XML-Documents shall conform to the provisions of SMPTE ST 377-1 and the provisions of the applicable documents of the SMPTE-Documents-Suite.

7 XML Encoding Rules

7.1 General

This standard complies to XML Schema Part 1: Structures and XML Schema Part 2: Datatypes.

7.2 Namespaces

All groups defined in a particular SMPTE Engineering Document and the MXF partition multiplex structure and index tables information shall lie in namespaces that are identified by an URI of the form

`http://www.smpte-ra.org/schemas/434/<434-ref-revision>/<shortname>/<ref-number>/<ref-revision>`

All properties and data types defined a particular SMPTE Engineering Document shall lie in namespaces that are identified by an URI of the form

`http://www.smpte-ra.org/schemas/434/<434-ref-revision>/<shortname>/<ref-number>`

The <434-ref-revision> shall be a string of digits identifying the year and version of this standard. Its format shall follow the same rules as the <ref-version> string, defined below.

The <shortname> shall be one of:

- “types” – if the namespace identified by this URI holds data types, appropriately defined in an SMPTE Engineering Document, and expressed as defined in Section 7.3;
- “property” – if the namespace identified by this URI holds metadata dictionary elements defined in an SMPTE Engineering Document, and expressed as defined in Section 7.4;
- “groups” – if the namespace identified by this URI holds groups, appropriately defined in an SMPTE Engineering Document, and expressed as defined in Section 7.5;
- “storage” - if the namespace identified by this URI holds information expressed as defined in Section 7.8.

The <ref-number> shall be a string representing the official unique number assigned by SMPTE to the SMPTE Engineering Document where the data types, properties or groups are defined.

The <ref-revision> shall be a string of digits identifying the year and version of the SMPTE Engineering Document. Its format shall be of the form yyyy or yyyy.n. The characters yyyy shall be replaced by the four digit year during which the SMPTE Engineering Document was published or amended. The term “.n” shall be absent for the first publication or amendment within year yyyy. For all subsequent revisions or amendments of the SMPTE Engineering Document within the same year the term “.n” shall be present and n shall equal the count of publication events (publication, revision and amendment) of the Specification in year yyyy minus 1.

Note: XML Namespaces are used to assure globally unique tag names and to link MXF-XML-Documents to their defining MXF-XML-Schema. Amendments or new revisions of documents of the MXF-Specification-Suite can change provisions of the original versions such as adding new groups or adding new properties to already existing groups. For this reason, a new XML Namespace is assigned that identifies that an MXF-XML-Document conforms to the revised or amended document of the MXF-Specification-Suite. According to SMPTE ST 377-1, all MXF files that are of the same major version are backwards compatible. As long as the major version number of the MXF file does not change, the XML Namespace associated with an older version of the documents of the MXF-Specification-Suite can be safely used to represent the information that conforms to that version (sets or properties that have been added with the amendment or revision will be represented as dark metadata). This can be used to manage extensibility.

The namespace URI, defined according to the above stated rules, and recommended default namespace prefixes used for the MXF-XML-Schema in Annex A are as shown in Table 1.

The definition of private types, properties and groups (such as class 13 or 14 entries in the registry controlled by SMPTE ST 335) shall be accompanied with the definition of appropriate namespace names (URI). Informative examples for this are provided in Section 7.1.1. The namespace names and corresponding namespace prefixes shall be used in the schema documents that extend this SMPTE ST 434 schema.

The private addition of properties to groups, the private addition of groups to metadata schemes and the private addition of properties or groups to the partition multiplex structure and index table information defined in SMPTE engineering documents shall be accompanied with the definition of appropriate namespaces. The definition of the URI for these namespaces lies outside the scope of this specification.

The URI of the namespaces of all groups that are intended to be part of MXF header metadata shall be added as `<xs:any namespace="<namespace URI>" processContents="strict"/>` declarations to the contents of the

second `xs:choice` model group in the definition of the *MetadataSets* XML element and to contents of the `xs:choice` model group in the definition of the *FloatingMetadataGroups* XML element.

Notes:

1 The *MetadataSets* and the *FloatingMetadataGroups* XML elements are defined in the MXF-XML-Schema file `Annex-B/file/metadata.xsd`.

2 Annex C.2 provides examples for extensions of the MXF-XML encoding and their impacts on namespace assignment.

Table 1 – XML Namespace URI Definitions

Specification and Identified Elements	Namespace URI	Namespace Prefix used in the remainder of this specification (informative)
Groups (sets and packs)		
All groups defined in SMPTE RP 2057	http://www.smp-te-ra.org/schemas/434/2014/groups/2057/2011	rp2057Groups
All groups defined in SMPTE ST 377-1. Dark metadata sets and their properties (XML attributes and XML elements) as defined in this specification.	http://www.smp-te-ra.org/schemas/434/2014/groups/377-1/2012	st377-1Groups
All groups defined in SMPTE ST 377-4	http://www.smp-te-ra.org/schemas/434/2014/groups/377-4/2012	st377-4Groups
All groups defined in SMPTE ST 380	http://www.smp-te-ra.org/schemas/434/2014/groups/380/2004	st380Groups
All groups defined in SMPTE ST 381-1	http://www.smp-te-ra.org/schemas/434/2014/goups/381-1/2005	st381-1Groups
All groups defined in SMPTE ST 381-2	http://www.smp-te-ra.org/schemas/434/2014/goups/381-2/2011	st381-2Groups
All groups defined in SMPTE ST 382	http://www.smp-te-ra.org/schemas/434/2014/goups/382/2012	st382Groups
All groups defined in SMPTE ST 385	http://www.smp-te-ra.org/schemas/434/2014/groups/385/2012	st385Groups
All groups defined in SMPTE ST 422	http://www.smp-te-ra.org/schemas/434/2014/groups/422/2006	st422Groups
All groups defined in SMPTE ST 429-5	http://www.smp-te-ra.org/schemas/434/2014/groups/429-5/2009	st429-5Groups
All groups defined in SMPTE ST 429-6	http://www.smp-te-ra.org/schemas/434/2014/groups/429-6/2006	st429-6Groups
All groups defined in SMPTE ST 429-10	http://www.smp-te-ra.org/schemas/434/2014/groups/429-10/2008	st429-10Groups