

Television METADATA —

Part 2: Data encoding protocol using key-length-value

The European Standard EN 62261-2:2006 has the status of a
British Standard

ICS 33.160.01; 35.040

National foreword

This British Standard was published by BSI. It is the UK implementation of EN 62261-2:2006. It is identical with IEC 62261-2:2005.

The UK participation in its preparation was entrusted to Technical Committee EPL/100, Audio, video and multimedia systems and equipment.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 May 2007

© BSI 2007

ISBN 978 0 580 50819 6

Amendments issued since publication

Amd. No.	Date	Comments

Television METADATA
Part 2: Data encoding protocol using key-length-value
(IEC 62261-2:2005)

Métadonnées des applications télévision
Partie 2: Protocole de codage des
données par méthode de longueur de clés
(CEI 62261-2:2005)

Fernseh-Metadaten
Teil 2: Protokoll zur Datencodierung mit
Schlüssellängenwert
(IEC 62261-2:2005)

This European Standard was approved by CENELEC on 2006-12-01. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the Central Secretariat or to any CENELEC member.

This European Standard exists in two official versions (English and German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the Central Secretariat has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

Central Secretariat: rue de Stassart 35, B - 1050 Brussels

Foreword

The text of the International Standard IEC 62261-2:2005, prepared by Technical Area 6: Higher data rate storage media, data structure and equipment, of IEC TC 100, Audio, video and multimedia systems and equipment, was submitted to the formal vote and was approved by CENELEC as EN 62261-2 on 2006-12-01 without any modification.

The following dates were fixed:

- latest date by which the EN has to be implemented
at national level by publication of an identical
national standard or by endorsement (dop) 2007-12-01
- latest date by which the national standards conflicting
with the EN have to be withdrawn (dow) 2009-12-01

Annex ZA has been added by CENELEC.

Endorsement notice

The text of the International Standard IEC 62261-2:2005 was approved by CENELEC as a European Standard without any modification.

CONTENTS

1	Scope.....	5
2	Normative references	5
3	KLV protocol.....	5
3.1	General	5
3.2	ANSI/SMPTE 298M universal label key	6
3.3	Encoding of the KLV length field	9
3.4	Encoding of data values	10
3.5	Empty metadata items	11
4	KLV coding of individual data items	11
4.1	General	11
4.2	Identification of value data representations	11
5	KLV group coding	12
5.1	General	12
5.2	Universal sets	13
5.3	Global sets	13
5.4	Local sets.....	15
5.5	Variable-length packs.....	18
5.6	Fixed-length packs	21
6	Labels	23
	Annex A (normative) Glossary of terms.....	24
	Annex B (informative) Example usage of Universal Label Key.....	26
	Annex C (informative) Example of the KLV encoding of a single metadata item	27
	Annex D (informative) Example of a universal set	28
	Annex E (informative) Example of a global set	29
	Annex F (informative) Example of a local set	30
	Annex G (informative) Example of a variable-length pack	31
	Annex H (informative) Example of a fixed-length pack	32
	Annex I (informative) Example of a label.....	33
	Annex ZA (normative) Normative references to international publications with their corresponding European publications	35
	Bibliography.....	34
	Figure 1 – KLV encoding.....	6
	Figure 2 – KLV coded universal set data structure	14
	Figure 3 – KLV coded global set data structure.....	15
	Figure 4 – KLV coded local set structure.....	17
	Figure 5 – Informative illustration of local set label to global key linking.....	19
	Figure 6 – KLV coded variable-length pack structure	20
	Figure 7 – KLV coded fixed-length pack structure	22
	Figure 8 – UL key for labels	23
	Figure B.1 – Example of universal label fields for metadata encoding in the SMPTE namespace	26

Table 1 – KLV fields for encoding of data.....	6
Table 2 – Field descriptions for the universal label key for the KLV encoding of data	7
Table 3 – UL designators for octets 5 through 7	8
Table 4 – Field descriptions for the UL key for the KLV encoding of universal sets	14
Table 5 – Field descriptions for the UL key for global set encoding	16
Table 6 – Coding of registry designator (octet 6) for global set syntax	17
Table 7 – Field descriptions for the UL key for local set encoding	17
Table 8 – Coding of registry designator (octet 6) for local set syntax	18
Table 9 – Field descriptions for the UL key for variable-length pack encoding	20
Table 10 – Coding of registry designator (octet 6) for variable-length pack syntax	20
Table 11 – Field descriptions for the UL key for fixed-length pack encoding.....	22
Table 12 – Field descriptors for the UL key for labels.....	23
Table B.1 – Expanded example of SMPTE universal label fields for metadata encoding.....	26
Table C.1 – Informative example of KLV individual value encoding of metadata.....	27
Table D.1 – Informative example of KLV universal set encoding of metadata (octets separated by spaces for readability)	28
Table E.1 – Informative example of KLV global set encoding of metadata (octets separated by spaces for readability)	29
Table F.1 – Informative example of KLV local set encoding of metadata (octets separated by spaces for readability)	30
Table G.1 – Informative example of KLV variable-length pack encoding of metadata (octets separated by spaces for readability).....	31
Table H.1 – Informative example of KLV fixed-length pack encoding of metadata (octets separated for readability)	32
Table I.1 – Example of a label.....	33

TELEVISION METADATA –

Part 2: Data encoding protocol using key-length-value

1 Scope

This part of IEC 62261 defines an octet-level data encoding protocol for representing data items and data groups. This protocol defines a data structure which is independent of the application or transportation method used.

The standard defines a key-length-value (KLV) triplet as a data interchange protocol for data items where the key identifies the data, the length specifies the length of the data, and the value is the data itself. The KLV protocol provides a common interchange for all compliant applications irrespective of the method of implementation or transport.

The standard also provides methods for combining associated KLV triplets in data sets where the set of KLV triplets is itself coded with KLV data coding protocol. Such sets can be coded in either full form (universal sets) or in one of four increasingly bit-efficient forms (global sets, local sets, variable-length packs, and fixed-length packs). The standard provides a definition of each of these data constructs. The encoding octet range (length of the payload) specified in this standard may generate unusually large volumes of data. Consequently, a specific application of KLV encoding is capable of only a limited operating data range and those details shall be defined in a relevant application document.

Of necessity, keys and other reference data have to be globally unique if clashes are to be avoided. The IEC will therefore, from time to time, designate other bodies to act as its registration authority and agent in this respect; it is important to note that, as a result of this, registrations will always contain the designator of the registration authority acting at the time of registration and not that of the IEC (unless the IEC was acting directly as its own registration authority). A mixture of registration authority designators is therefore to be expected.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8825-1:2002, (ITU-T X.690), *Information Technology – ASN.1 Encoding Rules – Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER), and Distinguished Encoding Rules (DER)*

ANSI/SMPTE 298M:1997, *Television – Universal Labels for Unique Identification of Digital Data*

3 KLV protocol

3.1 General

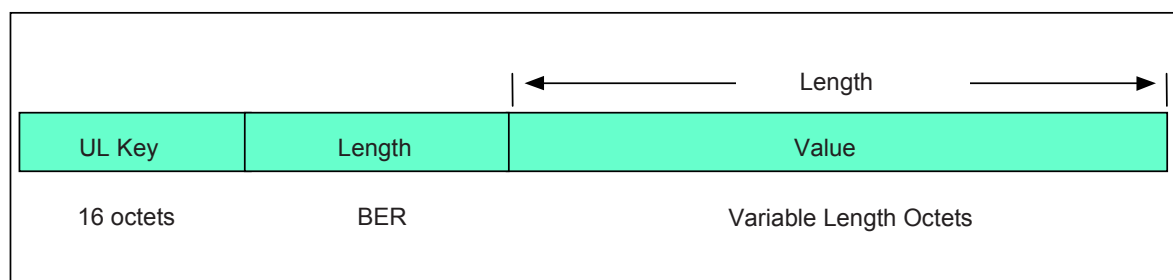
Table 1 and Figure 1 present an introductory view of the KLV protocol for encoding data. The data encoded may be a single data item or a data group. The coding of data items is described in Clause 4 while the coding of data groups is described in Clause 5 of this standard.

The KLV coding protocol is composed of a universal label (UL) identification key (UL key), followed by a numeric length (value length), followed by the data value.

The composition of the UL key is described in 3.2 of this standard. The length of the full UL key shall be 16 octets. The length field is described in 3.3 of this standard. The value is described in 3.4 of this standard. The value is a sequence of octets of the data type as specified in a relevant standard and is not further specified by the KLV protocol. The length of the value field is variable and any limitations are defined in a relevant defining standard.

Table 1 – KLV fields for encoding of data

Field	Description	Length	Content/Format
UL key	Universal label for identification of the value	16 octets	Subclause 3.2
Length	Length of the value field	Defined in a relevant dictionary, essence, application standard, but variable length	Subclause 3.3
Value	Value associated with the UL key	Variable	Subclause 3.4



IEC 1161/05

Figure 1 – KLV encoding

The bit-order (lsb or msb first) for KLV encoding shall be that of the transport used to carry the information.

3.2 ANSI/SMPTE 298M universal label key

3.2.1 General

KLV coding protocol shall use a 16-word universal label (UL) generated according to ANSI/SMPTE 298M as the UL key to identify the data in the value field but with the appropriate registration authority's 4-octet prefix following the OID tag and length byte.

Each word in the ANSI/SMPTE 298M UL is coded using the basic encoding rules (BER) for the encoding of an object identifier value specified in ISO/IEC 8825-1, 8.19. Each word of the UL key shall be limited to the range 0x00 to 0x7F and shall be represented by a single octet. The UL key shall have left-to-right significance with the first octet as the most significant. The leftmost octet of value 0x00 in the UL key shall define the termination of the label and all octets of lower significance shall also be set to 0x00. Octets of value 0x00 shall have no significance to the meaning of the UL key.

The full UL key consists of a 16-octet field including an object ID (OID) and the UL size (0x0E indicating a total UL key size of 16 octets) followed by a UL code and a series of sub-identifiers which shall define the UL designators. The first two UL designators shall have reserved values for the KLV coding protocol according to this standard.

NOTE The ANSI/SMPTE UL itself adopts KLV coding with the object ID as the key, the UL size as the length, and the UL designators as the value.

When applying the ANSI/SMPTE UL to the coding of data according to this standard, these fields shall be categorized as the UL header, UL designators, and item descriptor as shown in Table 2.

Annex B shows an informative example of a metadata UL key in tabular and figure formats.

Note that decoders which recognize the UL key but do not want to, or cannot, decode the associated value, may ignore the item and shall continue the decoding process of subsequent items using the length value to skip the value of the undecoded item. If decoders only store or forward the item, they shall forward the item unaltered.

3.2.2 UL designators

Table 3 defines octet values for the designators to be used in octets 5 through 7 of the UL designators. Standards and recommended practices which define a UL key with the value of octet 5 (registry category designator) in the range 0x01 to 0x04 shall register the full UL key or keys used with the IEC's designated registration authority in the registry identified by octets 6 and 7 (registry designator and structure designator).

Table 2 – Field descriptions for the universal label key for the KLV encoding of data

No.	Field	Description	Length	Content/format
1	UL header			
	OID	Object identifier	1 octet	Always 0x06
2	UL size	16-octet size of the UL	1 octet	Always 0x0E
3	UL designators			
	UL code	Concatenated subidentifiers ISO, ORG	1 octet	Always 0x0B
4	Registration authority designator	Registration authority subidentifier	1 octet	Dependant on Registration Authority (for example, 0x34 for SMPTE)
5	Registry category designation	Registry category designator identifying the category of registry described (for example, dictionaries)	1 octet	See Table 3
6	Registry designator	Registry designator identifying the specific registry in a category (for example, metadata dictionary)	1 octet	See Table 3
7	Structure designator	Designator of the structure variant within the given registry	1 octet	Subclause 3.2.3
8	Version number	Version of the given registry which first defines the item specified by the item designator	1 octet	Incrementing number
9-16	Item descriptor	Unique identification of the particular item within the context of the UL designator	8 octets	See relevant standard and version

Table 3 – UL designators for octets 5 through 7

Registry category Octet 5	Registry Octet 6	Structure Octet 7	Described in
01 - Dictionaries	01 – Metadata dictionaries 02 – Essence dictionaries 03 – Control dictionaries 04 – Types dictionaries	0x01-0x07F – Structure designator 0x01-0x07F – Structure designator 0x01-0x07F – Structure designator 0x01-0x07F – Structure designator	Clause 4 Other standard/practice Other standard/practice Other standard/practice Other standard/practice
02 – Groups (sets and packs)	01 – Universal sets 02 – Global sets (default) 03 – Local sets (default) 04 – Variable-length packs (default) 05 – Fixed-length packs	0x01-0x07F – Structure designator 0x01-0x07F – Structure designator 0x01-0x07F – Structure designator 0x01-0x07F – Structure designator 0x01-0x07F – Structure designator	Clause 5 Subclause 5.2 Subclause 5.3 Subclause 5.4 Subclause 5.5 Subclause 5.6
03 – Wrappers and containers	01 – Simple wrappers 02 – Complex wrappers	0x01-0x07F – Structure designator 0x01-0x07F – Structure designator	Other standard/practice Other standard/practice
04 – Labels	01 – Labels dictionary	0x01-0x07F – Structure designator	Clause 6 Other standard/practice

3.2.2.1 Dictionaries

Standards and recommended practices which define the value of word 5 of the UL key as 0x01 are dictionary standards and shall be used to define single data items with the KLV data construct. The coding of individual items is defined in Clause 4.

3.2.2.2 Sets and packs

Standards and recommended practices which define the value of word 5 of the UL key as 0x02 are set and pack standards and shall be used to define groups of KLV coded data items. The coding of data groups is defined in Clause 5.

3.2.2.3 Wrappers and containers

Standards and recommended practices which define the value of word 5 of the UL key as 0x03 are wrapper and container standards and use the UL key to identify the wrapper or container and its contents. Wrappers and containers differ from sets and packs in that they do not necessarily employ an overall KLV data construct for the entire contents of the wrapper or container. It is recommended that individual parts of a wrapper or container encode data using the KLV coding protocol, but these parts may be bound together by other techniques. In some cases, a wrapper or container may employ an overall KLV construct in certain applications (such as a streaming interface) but employ another technique in other applications (such as a storage container). In these cases, the wrapper or container is not redefined as a set or a pack but retains the definition as a container or wrapper for consistency of identification.

Simple wrappers and containers are defined as embedding all the data into a single framework with no external references. Complex containers and wrappers are defined by frameworks where individual data items may be included in a file by reference rather than embedding. Complex containers and wrappers can be more efficient and are suited to local environments where references can be easily resolved. The definition of wrappers and containers is outside the scope of this standard and can be found in other documents.

3.2.3 Structure designator

Octet 7 shall contain the structure designator for the given registry. Structure designators are allocated to distinguish between incompatible versions of the same registry. They may be thought of as a major version number.

NOTE Different structure designators may be assigned when a registry becomes so cluttered that it has outlived its usefulness, or when changes to the contents, structure, or class relationships of the registry are necessary which prevent backward compatibility.

Different structure designators may also be assigned to distinguish between different syntax rules for construction and interpretation of values. Different structure designators may also be assigned when more than one standard or practice is used to define the contents of a specific registry. The precise discipline for allocation of structure designators is described for each specific registry in the clauses below.

3.2.4 Version number

Octet 8 shall contain the version number of the given registry which first defines the item specified by the item designator.

New items may be added to registries after initial approval of the controlling standard or practice. Each time a set of item definitions is added, the current version number of the particular registry is incremented. Each entry in a registry includes the version number in which the item was first defined. It is this number which is carried in octet 8.

Parsers may use the version number as an additional guide and consistency check in the process of parsing a UL key.

3.2.5 Item designator

Octets 9 through 16 of the UL key comprise the item designator. The item designator field is fixed 8 octets in length. Item designators are from 1 to 8 bytes long, and are padded on the right with zero octets to fill the 8-byte field. ASN.1 object identifier coding is used to provide for allocation of varying quantities of sub-identifiers in a hierarchical manner.

The precise meaning and construction of the item designator depends upon the specific registry and structure variant, and is described further in the clause below.

3.3 Encoding of the KLV length field

In the KLV coding protocol, the value of the length field shall be encoded using the basic encoding rules (BER) for either the short-form or long-form encoding of length octets specified in ISO/IEC 8825-1, 8.1.3. This method of value length encoding is self-contained and allows for efficient parsing of KLV encoded data. When the KLV coding protocol is applied to groups of KLV coded units, the length field for individual units may adopt a different method as defined by the standard for the coding of that group (see Clause 5).

NOTE 1 Bits are numbered bit 8 to bit 1 with bit 8 as the MSB.

NOTE 2 Receivers/decoders must decode the length field correctly whether the long or short form is used.

3.3.1 BER short-form length encoding

The following normative subclause (including the example) is quoted from ISO/IEC 8825-1:

- 8.1.3.4: In the short form, the length octets shall consist of a single octet in which bit 8 is zero and bits 7 to 1 encode the number of octets in the contents [value] octets (which may be zero), as an unsigned binary integer with bit 7 as the most significant bit.

Example: L = 38 can be encoded as 00100110₂.

The short form for length encoding shall be used whenever the data value length is less than 128 octets.

3.3.2 BER long-form length encoding

The following normative subclause (including the example) is quoted from ISO/IEC 8825-1:

8.1.3.5: In the long form, the length octets shall consist of an initial octet and one or more subsequent octets. The initial octet shall be encoded as follows:

- a) bit 8 shall be one;
- b) bits 7 to 1 shall encode the number of subsequent octets in the length octets, as an unsigned binary integer with bit 7 as the most significant bit;
- c) the value 1111111_2 shall not be used.

NOTE 1 This restriction is introduced for possible future extensions.

Bits 8 to 1 of the first subsequent octet, followed by bits 8 to 1 of the second subsequent octet, followed in turn by bits 8 to 1 of each further octet up to and including the last subsequent octet shall be the encoding of an unsigned binary integer equal to the number of octets in the value field, with bit 8 of the first subsequent octet as the most significant bit.

NOTE 2 This is sometimes known as big-endian byte order.

Example: $L = 201$ can be encoded as: Octet 1 = 10000001_2 Octet 2 = 11001001_2 [b8 ... b1]

NOTE 3 While there are no restrictions in this standard on the maximum number of octets in the data value length field, the presence of large data value lengths can be determined from the first octet in the BER long-form length encoding.

Where appropriate, individual application standards and recommended practices may define the maximum octet length of the length field or may place limitations on the value range of the length field in order to simplify decoder requirements.

Implementations shall make every effort to apply a valid value to the length field. However, in certain operations, it may prove impractical to establish the length of the value field. Such a case is an incoming data stream which is assigned a key and a length field at the start point. In this case, the value of the length cannot be established until the termination of the stream and at that point, it may prove impossible to return to the length field to enter the value. In such cases, the length field shall be set to $[0x80]$ which shall indicate a non-deterministic length of the value field. Any application document which allows the length of the value field to be undefined must define an alternative method of locating the end of the value field.

NOTE The length value $[0x80]$ is used because it is normally meaningless as a BER long-form value as it indicates zero subsequent octets.

3.4 Encoding of data values

Data values may be either individual data items or data groups. In either case, the data is an octet string whose length is specified by the length value. The last octet of the value shall be the terminating octet of the data sequence.