INTERNATIONAL LASER
DISPLAY ASSOCIATION

Technical Committee

The ILDA Digital Network

Stream Specification

Standard Identifier: IDN-Stream

Revision 001, July 2015

Table of Contents

1 Introduction

This technical standard is part of the International Laser Display Association Digital Network (ILDA Digital Network, IDN) protocol suite and describes the encoding of laser show artwork into digital data streams. This can be from single laser projector data sent across a network connection up to entire laser shows including multimedia content, stored in data files for playing back in a target environment. The standards for the network connection itself, session maintenance, time control or the file structure however are defined in other parts of the IDN protocol suite.

A long time ago people used to wire their projectors as they thought it was best, resulting in almost as many hardware variations and configurations as there were people producing laser light shows. Then, in 1989 (CPC-37, abandoned) and in 1996 (ISP-DB25, ISP-DMX, ISP-ADAT) ILDA set standards for laser projector connections and artwork interchange. While the original wiring standard lost importance due to its size and cost, the success of the ISP-DB25 standard is remarkable. It's used with almost every laser projector.

Time has changed and demands have changed. Where there used to be single projectors, now multiple projectors are operated simultaneously. ISP-DB25 cables are heavy, hard to maintain and the DB25 connector is large. Setups must be efficient and special care for analog cabling (ground loops etc.) is costly. ADAT can only store one projector per tape (Multi-Projector shows require multiple synchronized tapes running simultaneously), uses magnetic video tape (magnetic fields, tape jam) and is limited to three colors. Over time and due to the abandoned technology, it lost most of it's importance.

People again did what they thought was best and built converter hardware (USB or networking) to be used with player software. These boxes still use ISP-DB25 as the common projector interface, along with the cabling. But this additional hardware needs to go somewhere, causing new trouble with setups. Further, protocols and the encoding of the data streams are proprietary and thus not compatible causing cost and inconvenience to users. With artwork interchange, people use hard disk recorders or multi channel wave files, still struggling with color mismatches.

ISP is maxed out. There is a lack of “meta information” - information about each projector provided by the projector itself. While ISP is still good for analog signals, “plug and play” is not working since just about every projector uses a different set of colors. With ISP, the projector has no way of reporting about its wavelengths resulting in manual setup procedures and mismatches. Further, more advanced projection systems need additional custom communication for monitoring and control, usually a network connection. Together with the converter boxes, this requires another box (a switch) or another patch cable.

The goal of this standard is to set a new direction in laser projection data handling. It is to add meta information and timing to content and thereby making artwork describe itself instead of being bound to specific environments.

This standard doesn't necessarily describe how the stream is to be handled. Depending on the environment, network, session configuration, link type or storage in files, the stream may have to meet specific requirements.

This standard specifies a format that can be used in communication and storage. It wants to unify digital connections by moving the projector interface from an electrical level to a protocol level and wants to offer a new universal exchange format for movie-like artwork. It again ensures the compatibility and interchangeability of hardware, software and artware in order that the results are predictable and that the artwork is faithfully reproduced from system to system like ISP did when it was defined.

1.1 Nomenclature and Structure

Throughout this document, the word "SHALL" is used in capitals to stress required conformance. The word "SHOULD" in capitals indicates suggested conformance.

1.2 Octets / Bytes / Endianness

Generally, the term "byte" is avoided as it is ambiguous. Alternatively, the term "octet" is used as it unambiguously specifies a size of eight bits. For multi byte/octet data words, big endian byte order (network byte order) is used. This specifies that the most significant octet is stored (or transmitted) first and the least significant octet is stored (or transmitted) last.

1.3 Version

When reported to a producer, a consumer implementing this revision of the IDN-Stream protocol SHALL report a version number of 1.0. This protocol version is introduced for explanatory reasons and may not have to be used. Due to the distributed approach of the IDN protocol suite, a producer may check for the existence of single features instead. The protocol version may not be related to the revision number of the standard. Changes of the major version number SHALL indicate significant enhancements or changes. Changes of the minor version number SHALL indicate changes that do not affect compatibility.

1.4 Help Improving

This standard has been elaborated and put together very carefully. However, it is complex and may miss explanations or contain ambiguity. Please tell ILDA in case you, as an implementer, come across such passages. This way ILDA can clarify with the next revision and implementations stay compatible.

2 IDN Channel Messages

IDN communication is packed into messages. These messages consist of a header section and a data section. The header section of all messages starts with a 16 bit size field to delimit messages from each other when present in a stream and a 16 bit content identifier which is used to interpret the data section.

2.1 Channel Message Header

Channel messages use a large set of content identifiers (0x8000-0xFFFF) and thus interpret the content identifier specifically by breaking it up into sub-fields. Further, channel messages add fields to the basic message header.

Octet	0	1	2	3
0	Total Size		CNL	Chunk Type
4	Timestamp			
8...	Channel Configuration (optional)			

Total Size

This is the total amount of octets in the message (header section plus data section) and SHALL be in range 0x0008 to 0xFF00. The lower limit comes from the smallest channel message header (no configuration, no data section) and the upper limit is mandatory for IDN messages because space may be needed in case messages are wrapped in protocols that need to add further headers while still keeping 16 bit size fields. This field allows consumers to allocate buffers, check boundaries or skip over messages.

Channel configuration and routing information (CNL)

For channel messages, the most significant octet of the basic message content identifier contains channel configuration and routing information. The most significant bit of this octet (which corresponds to the most significant bit of the content identifier) is always '1', assigning an identifier range from 0x8000 to 0xFFFF to channel messages.

Bit	0	1	2	3	4	5	6	7
	1	CCLF	Channel ID					

Channel Configuration and Last Fragment bit (CCLF)

For single message data chunks and messages carrying the first fragment of a large data chunk, this bit indicates the presence of channel configuration data. If set to '1', the message header is followed by a configuration header (which may be followed by service configuration data). If set to '0', the message header is directly followed by the data section.

Since messages carrying fragments other than the first one can't have channel configuration headers, this bit, if set to '1' indicates the message with the last fragment of the data chunk.

Channel ID

This is an arbitrary identifier for the channel, defined by the producer and used by the consumer to identify the channel. Channels get opened (and connected to a service), receive data and get closed. Up to 64 channels can be open simultaneously per device or realm in a file.

The implementation might limit the amount of simultaneously open channels but SHALL support the whole range of channel identifiers. This means that producers are free to choose any identifiers for their channels.

Chunk Type

Type identifier for the data section of the message.

Types 0x00 to 0xBF are used for single message data chunks and for the first fragment of large data chunks which are split across multiple messages. Types with the two most significant bits set, 0xC0 to 0xFF, are used for the remaining fragments of large data chunks. Messages carrying these types must be treated differently as their timestamp field is shared with the fragment number. Moving fragmentation into the application layer is a design decision and motivated by buffer management and data interpretation.

Types 0xA0 to 0xBF (single message / first fragment) and 0xE0 to 0xFF (other fragments) are reserved for dynamic type assignments through session configuration and SHALL be disabled by default. These types can be used for manufacturer specific implementations.

Type	
0x00	Void (no data section)
0x01	Laser Projector Wave Samples
0x02	Laser Projector Frame Samples (entire chunk)
0x03	Laser Projector Frame Samples (first fragment)
0x10	Octet Segment
0x11	Octet String
0x18	Dimmer Levels
0xC0	Laser Projector Frame Samples (sequel fragment)

Messages with a chunk type of 'Void' do not carry a data section. These messages however can have a configuration header and can be used to open, configure or close a channel in case management is implemented separate from data streaming. Depending on the environment or session configuration, 'Void' chunks can be used in keep alive messages to prevent the automatic close of idle channels.

Timestamp

All messages SHALL have a valid timestamp in microseconds. The first message received by a session can have an arbitrary timestamp. Thereafter, the timestamp has to increase monotonically. Identical timestamps are valid but disadvantageous as the consumer may use the timestamp for clock synchronisation algorithms. Wraps must be taken care of on consumer side.

In case of realtime streaming, the simplest way for assigning timestamps is to take the system time when sending the message. This allows the consumer to adapt to the producer timing. When using file storage, streaming with fixed latencies or ahead of time streaming with buffering and clock manipulation, the timestamp can be used for the precise placement of output events.

Chunk types 0xC0 to 0xFF must be excluded when using the timestamp for consumer side calculations. These chunk types indicate fragments of large data chunks. In this case, the timestamp is shared with the fragment number and must be equal to the timestamp used for the first fragment plus the (0-based) number of the fragment.

2.2 Channel Configuration Header

To manage a channel (open, close, configure), a channel configuration header must be present in the header section. For messages containing an entire data chunk or messages containing the first fragment of a large data chunk, the configuration header is appended to the channel message header in case the CCLF bit is set.

Separating the channel configuration from the actual data is an optimisation and reduces overhead and processing. The configuration, however, must be present when opening a channel since data interpretation and routing must be known. Further, specific requirements may have to be met depending on the environment. For example, on unreliable connections, the configuration SHALL be present every 200ms or 250ms for recovery and plug and play. On the other hand, configuration processing may consume significant resources. For this reason, changes SHALL not be done faster than 100ms per channel. Consumers may use binary comparisons before processing new configurations. In files, the channel configuration may have to be present at synchronisation entry points or recovery markers.

Octet	0	1	2	3
0	SCWC	CFL	Service ID	Service Mode

Service Configuration Word Count (SCWC)

The amount of 32 bit service configuration words that follow the header. In case the Routing flag is not set, this field SHALL be set to 0 by producers and ignored by consumers.

The presence of service configuration data is optional but may be required by service modes. When present (SCWC not equal '0'), the configuration data is passed transparently to the assigned service.

Channel and service configuration Flags (CFL)

This field contains control bits for the channel routing and the assigned service.

Bit	0	1	2	3	4	5	6	7
	00		SDM		00		Close	Routing

Service Data Match (SDM)

Service data match for configurable service modes. In case a service mode is configurable, the service stores these bits along with its new configuration. In case the service mode is not

configurable or the Routing flag is not set, this field SHALL be set to '00' by producers and ignored by consumers.

When data chunks are processed, the bits are compared with the match bits found in the data chunk header. In case the service configuration has changed and has led to a changed data structure or interpretation, these bits SHALL be changed too to signal data incompatibility. This is a cross check feature for unreliable connections or file recovery in case the message containing the configuration or the message that closed the channel is lost.

Close

This bit, when set, signals that the channel is to be closed after the message has been processed. It is absolutely valid to open a channel, pass data and close the channel with a single message.

Routing

This bit, when set, signals that the channel is to be opened (if not already opened) and routed to the specified service before the message data is processed. When set, the Service ID and Service Mode fields must be valid in order to connect the channel to a service. Further, in case the service mode is configurable, the SCWC and SDM fields may have to be set to pass the configuration. It is absolutely valid to open a channel, pass data and close the channel with a single message. Opening a channel without passing data SHALL place the channel into an idle state until the first data chunk arrives.

Combining channel open, configuration and the first data chunk in a single message supports 'instant on' needed by realtime applications that have to keep low latencies between pressing a button and generating device output.

Service ID

The identifier of the service that the channel is to be connected to. This is important for files where multiple data streams containing same chunk types must be assigned to different tracks or devices that offer the same type of service more than once. In case the Routing flag is not set, this field SHALL be set to 0 by producers and ignored by consumers.

With a set Routing flag, this field can be set to 0x00 which means that the channel is to be connected to the default service handling the passed service mode.

Service Mode

The mode, the service is to be operated in. In case the Routing flag is not set, this field SHALL be set to 0 by producers and ignored by consumers.

Service mode identifiers 0xC0 - 0xFF are reserved for dynamic mode assignments through session configuration and SHALL be disabled by default. These modes can be used for manufacturer specific implementations.

Mode	
0x00	Void
0x01	Laser Projector Graphic (Continuous)
0x02	Laser Projector Graphic (Discrete)
0x03	Laser Projector Effects (Continuous)
0x04	Laser Projector Effects (Discrete)
0x05	DMX512 (Continuous)
0x06	DMX512 (Discrete)

A service mode of 'Void' is not valid for channel routings and SHALL be avoided. However, consumers SHALL implement this mode for channel testing purposes and ignore passed data chunks.

Continuous versus discrete: Consumers have to manage buffering and there are two different ways of doing this. While continuous streams provide gapless data that can't be repeated, the data delivered by discrete streams can be stored and repeated if needed until new data arrives. This makes continuous streams more difficult to handle with regard to buffering, jitter and underruns.

Service modes may overlap in function but must be unique for different devices in order to be able to assign a service in case the channel is to connect to the default service (Service ID set to 0x00).

3 Laser Projector Service

Laser projectors are built in many different styles. Some are monochromatic or colored and offer graphics only, while others have additional color lines for a wider gamut and a complement of lumia, scan-through effects and beam actuators. The IDN laser projector service is the representation of all projectors.

The existing ISP and IDTF standards give more information on how hardware is to look like and image exchange files are structured. This standard focuses on the unification and description of the digital data stream.

ISP-DB25, ISP-TAPE

These standards define how graphic signal sources and projectors with graphic scanners are to be connected and artwork is to be recorded. The IDN substitute is the continuous graphic mode.

ISP-DMX, ISP-EFX, ISP-TAPE

These standards define how laser effects are to be handled and recorded. The IDN substitute is the effects mode. This can be either continuous or discrete depending on the application. Discrete mode is good as a default since the nature of the data is frame oriented. Continuous mode may improve the latency for some applications.

IDTF

This standard defines how laser image files are to be exchanged between systems. In case the file contains a movie, an IDN playback system can use the discrete graphic mode to encode the frames.

3.1 Laser Graphic

Graphic scanners are used in laser projectors to deflect the beam horizontally and vertically. This is used to draw images on surfaces and to project atmospheric beam effects. Due to inertia and driver characteristics, the commanded position follows a waveform. At the same time, the laser is modulated to draw a stroke. With regard to signal frequencies, this modulation is the more demanding part since colors may change while the stroke is drawn.

The basic encoding for graphics are samples of uniformly spaced times like used when digitizing audio signals. The samples can be computed on the fly, be created when reading points from a file or be obtained by sampling waveforms of existing signal sources. Other graphic encodings like polynomial compression or vectors may follow.

A major goal of this standard is to stay compatible with the analog ISP-DB25 standard. In order to be able to smoothly reproduce sampled analog waveforms, laser projector services SHALL be able to at least play back sample frequencies of 100 kHz. This also defines the minimum capabilities to expect from systems with regard to persistent storage in files. Further, both, continuous and discrete modes for graphics and effects SHALL be implemented for the service to guarantee basic compatibility.

3.2 Continuous Graphic Mode

This service mode transmits an uninterrupted sequence of samples representing a waveform. Since this is an exclusive operation, no other channel can use this mode on the same service at the same time.

The first sample of the waveform after opening a channel in this mode SHOULD be at the origin with all coordinates set to 0 and all color lines off. The frequency spectrum of the waveform is in the responsibility of the producer and SHALL match the consumers capabilities.

In this mode, the shutter is initially closed. This means that the shutter of a laser projector service SHALL not open until data is received. The shutter is then either implicitly open or controlled by the sample stream. It SHALL close when the channel is closed.

Accepted data chunks:

- Laser Projector Wave Samples

3.3 Discrete Graphic Mode

This mode transmits a sequence of frames. These frames are drawn one after another as they are scheduled. Frames that started to draw must finish before the next frame is processed. Newer frames replace older ones and in case of an overrun (more frames received than processed), older frames are dropped. In case frames arrive slower than being processed, the last frame is either repeated or the scanning system is to wait for the next frame depending on the single scan flag in the data chunk header.

Each frame has a start position and an end position that may be different from the start position. The projector is responsible for moving the scanners from the end position to the start position of the next frame. Depending on the implementation, the scanning system may be shared by multiple discrete mode channels, each delivering frames to be drawn in different frequencies and speeds. In this case, the implementation is responsible for proper buffering and all necessary movements that are needed to connect the individual frames.

In this mode, the shutter is initially open. This means that the shutter of a laser projector service SHALL open with the first channel operating in this mode and stay open as long as there is at least one channel open in this mode. Empty frames SHALL not close the shutter.

The producer SHALL not make assumptions on durations or frame repeat rates since timing, buffering, movements between frames and the amount of simultaneous open channels are load and implementation dependent. In case tighter timing control is desired, the producer can use single scan frames or switch to continuous graphic mode.

Accepted data chunks:

- Laser Projector Frame Samples

3.4 Graphic Mode Configuration

Graphic scanning systems vary depending on preferences, application, technology and requirements. Even most basic X/Y/R/G/B projectors differ in wavelengths of their primary colors. Advanced projectors have additional color lines to cover a higher gamut or a second pair of scanners for stereoscopic displays. Listing all possible combinations would lead to a confusing and very likely incomplete set of variations.

The approach of IDN is a dictionary that is used to create a decoder for the sample structure passed in the data sections of the messages. The dictionary is an array of tags. These tags configure general decoder properties, define functions or describe an octet in the sample. No general assumption on the order of the tags SHALL be made but a sequence of tags may be required by certain functions. The order of octet descriptor tags must match the order of octets in the sample array though.

Dissenting from the ISP standard, IDN does not require the shutter status to occur in the data stream. Shutter output can be derived from channel open and close events. Mandatory tags are X and Y for the draw position and at least one draw color. Further tags like modifiers or descriptors for additional color lines are optional.

Consumers SHALL implement all tags specified in this version of the standard but may choose to modify or ignore the stream in case of incompatibilities. Smart decoders may, as long as all mandatory tags are present, generate an output even in case of unknown tags of a defined combination of category and subcategory. They SHALL abort though in case of tags of an unspecified combination of category and subcategory since format and function can't be known.

3.4.1 Generic Tag Structure

All tags are 16 bit words. The codes are structured along the four nibbles of the word (a nibble is a unit of 4 bits).

Nibble	0	1	2	3
	Category	Subcategory	Identifier	Parameter

Category (CAT) / Subcategory (SUB)

These two nibbles are used to group the tag identifiers. They can be used to implement a fast tablewalk through two tables, using both nibbles as indexes for consecutive lookups to access the function for the identifier. A single table lookup can be used as well but may consume more memory.

Category	Struct		Octet
0	Yes	Basic decoder modifiers with data	No
1	No	Basic decoder modifiers with parameter	No
4	No	Basic sample octet descriptors	Yes
5	No	Draw color and attributes	Yes

Tags of struct categories can be followed by 16 bit data words. These tags use their parameter to pass the number of data words that belong to the tag and the next tag is found after this data area.

Tags of modifier categories are for general decoder configuration. These tags do not have an associated octet in the sample but are used to pass options.

Tags of descriptor categories describe single data octets in the sample in ascending order. The first tag describing an octet in the sample describes the first octet, the second tag describes the second octet and so on.

Categories 12 to 15 are reserved for dynamic tag assignments through session configuration and SHALL be disabled by default. These tags can be used for manufacturer specific implementations.

Identifier (ID) / Parameter (PRM)

Each decoder function has an identifier. Depending on the category, the parameter can be used to pass arguments or the number of 16 bit data words following the tag.

3.4.2 Category 0, Subcategory 0 Tags

Standard modifiers with data.

CAT/SUB	ID	PRM	
0000.0000	0000	xxxx	Void

Void

This tag can be used to align the tag array to a 32 bit boundary. Since all headers are 32 bit aligned and the service configuration is appended to the channel configuration in the header section it is required for the tag array to be 32 bit aligned. In case the parameter is different from 0, parsing SHALL skip over the amount of 16 bit words passed in the parameter. This tag can be used for testing purposes.

3.4.3 Category 1, Subcategory 0 Tags

Standard modifiers with parameter.

CAT/SUB	ID	PRM	
0001.0000	0000	xxxx	Break

Break

This tag starts the next group of signals and can be used in applications where multiple independent sets of scanners and associated lasers of one physical projector must be operated on a sample-synchronous base. Standard applications use only one scan head and may either abort or ignore all following tags in case of reading the break tag. Multi-Head application that are to be operated sample-synchronous set default values, assign the next scan head and start over with parsing the next tag. The parameter is currently unused and SHALL be set to 0.

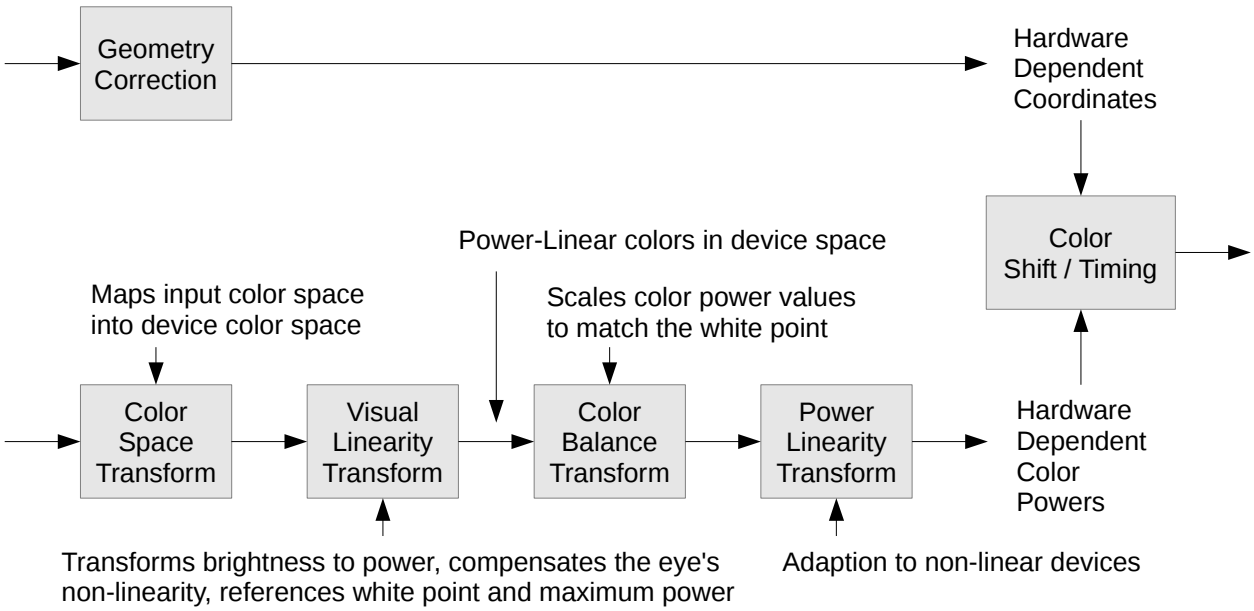
3.4.4 Category 1, Subcategory 1 Tags

Coordinate and color space modifiers.

CAT/SUB	PRM	PRM	
0001.0001	xxxx	xxxx	Space Modifier

If preset, this tag tells the decoder about the coordinate and color space referenced by the content. In case the stream is stored in a file, playback software can retrieve capabilities from the projector and modify values for unsupported transformations. For compatibility reasons, in case this tag is not present, the default behaviour is projector specific.

Transformation Pipeline



Tag Parameters

The coding of this tag varies from the standard scheme. The identifier field is used to extend the parameter to 8 bit.

Bit	0	1	2	3	4	5	6	7
	GLIN		CLIN		CBAL		CTIM	

A projector specific behaviour assumes that producer and consumer have negotiated a behaviour and stream data is encoded to meet this setting. Likewise, disabled transformation assumes native projector behaviour. Both options are device dependent or may be used for test setups. The use of these settings is discouraged for generic applications and file transfer as device dependency may have a significant impact on quality.

Options may be redundant. For example, in case the used laser hardware already produces power-linear output, 'Projector specific', 'No transform' and 'Power linear' options all refer to the same native setting. Likewise in case a transform can't be disabled. Color timing compensation for example may be implemented in a way that it can't be disabled. In this case, again, 'Projector specific', 'No transform' and 'Correlated in time' options all refer to the same setting.

Graphic Space Linearity (GLIN)

GLIN	
00	Projector specific
01	Geometrically corrected and linear, aspect ratio 1:1
10	Reserved
11	No transformation

In case this modifier is used, the stream contains linear coordinates and assumes an appropriate setup. For proper display, the consumer SHALL compensate for nonlinearities and transform the stream coordinates into its native coordinates by applying a geometric correction.

An aspect ratio of 1:1 implies square pixels since the size of the graphic space is even on all coordinates. Consequently, aspect ratios other than 1:1 are encoded by limiting the values on the corresponding axis. Since the origin SHALL stay in the center, limiting would apply to positive and negative coordinates symmetrically.

Color Space Linearity (CLIN)

CLIN	
00	Projector specific
01	Power linear (half value SHALL be half power)
10	Visually linear (half value SHALL be half brightness)
11	No transformation

In case this modifier is used, the stream contains linear color values (either in brightness or in power) and assumes an appropriate setup. For proper display, the consumer SHALL compensate for nonlinearities and transform the stream color values into its native color values.

Color Balance (CBAL)

CBAL	
00	Projector specific
01	White balanced
10	Reserved
11	No transformation

In case this modifier is used, the stream contains color values referring to a specific color balance and assumes an appropriate setup. For proper display, the consumer SHALL scale the stream color values to achieve the specified color balance.

Color Timing (CTIM)

CBAL	
00	Projector specific
01	Coordinates and colors correlated in time
10	Reserved
11	No transformation

In case this modifier is used, the stream contains coordinates and color values that correlate in time and assumes an appropriate setup. For proper display, the consumer SHALL delay the signal paths (shift color lines against coordinates and color lines against each other) according to its hardware.

3.4.5 Category 4, Subcategory 0 Tags

Standard sample descriptors

CAT/SUB	ID	PRM	
0100.0000	0000	0000	NOP
0100.0000	0001	0000	Precision

No OPeration (NOP)

The NOP tag ignores its associated octet in the sample and can be used for testing purposes.

Precision

Every descriptor tag has exactly one associated octet in the sample. Descriptor tags that allow for higher precision interpret their octets left aligned and can be directly followed by precision tags to add octets of precision to their value.

3.4.6 Category 4, Subcategory 1 Tags

Draw control descriptors

CAT/SUB	ID	PRM	
0100.0001	0000	xxxx	Hint

Hint

In case present, the hint octet in the sample gives additional information to the decoder. The parameter of the hint tag is used to select the format of the associated octet.

PRM \ Bit	0	1	2	3	4	5	6	7
0000	CSCL		ISCL		0	0	0	0
0001	CSCL		ISCL		0	0	0	SHT

Format 0 sets the shutter open and can be used in discrete mode and continuous mode when the sample stream doesn't contain shutter information.

Format 1 moves control over the shutter into the sample stream and is useful in continuous mode when the sender needs direct control over the shutter. Format 1 may not be valid in discrete mode.

Color scale (CSCL)

These two bits define a divider by which colors are scaled. This feature affects all color values simultaneously and can be used to move the value into an optimal numeric range. A single octet of color information is sufficient for most applications. With higher powers, dark colors may get coarse though but using two octets may just increase overhead. With this divider, all color values are shifted right by 2 times the scale bits which equates to a division by 4 raised to the power of the scale. A color scale of 0 leaves all colors untouched, a scale of 1 divides by 4, a scale of 2 divides by 16 and a scale of 3 divides by 64. This extends the numeric range from 8 bits to 14 bits.

Intensity scale (ISCL)

These two bits define a divider by which the intensity is scaled. The handling of intensity has been included for the support of legacy ISP systems. The signal is used for blanking and single wavelength lasers. IDN systems would just send colors. It is currently unclear in which way a separate intensity signal can make sense with IDN streams. Therefore, intensity handling is separate and SHALL be only used to control legacy systems. The usage of the scale corresponds to the color scale.

Shutter (SHT)

Status of the optical shutter in case the control is moved to the sample stream. The shutter is intended to extinguish all light from the graphic scanners. A value of '1' SHALL open the shutter and corresponds to signal level of +5V on the ISP-DB25 connector. A value of '0' SHALL close the shutter such that no light is emitted and corresponds to signal level of 0V on the ISP-DB25 connector.

3.4.7 Category 4, Subcategory 2 Tags

Draw coordinate descriptors

CAT/SUB	ID	PRM	
0100.0010	0000	xxxx	X
0100.0010	0001	xxxx	Y
0100.0010	0010	xxxx	Z

X

Controls the horizontal beam position. Octet values are interpreted as left aligned signed numbers in range [-128 .. -1, 0, 1 .. 127]. This tag can be followed by precision tags to extend the numeric range to 16 bit or higher. The most negative number corresponds to a signal level of -10V on the ISP-DB25 connector and SHALL deflect to the left (front projection), the most positive number corresponds to a signal level of +10V on the ISP-DB25 connector and SHALL deflect to the right.

A tag parameter value of 0 SHALL address the standard scanner which corresponds to the signal 'X' in the ISP-DB25 standard. In case multiple scanners are used with the same laser, further X coordinates can be addressed with the parameter.

For stereoscopic applications, a tag parameter of 0 SHALL address the scanner used for left eye graphics which corresponds to the signal 'X' in the ISP-DB25 standard and a tag

parameter of 1 SHALL address the scanner used for right eye graphics which corresponds to the signal 'X-prime' in the ISP-DB25 standard.

Y

Controls the vertical beam position. Octet values are interpreted as left aligned signed numbers in range [-128 .. -1, 0, 1 .. 127]. This tag can be followed by precision tags to extend the numeric range to 16 bit or higher. The most negative number corresponds to a signal level of -10V on the ISP-DB25 connector and SHALL deflect to the bottom, the most positive number corresponds to a signal level of +10V on the ISP-DB25 connector and SHALL deflect to the top.

A tag parameter value of 0 SHALL address the standard scanner which corresponds to the signal 'Y' in the ISP-DB25 standard. In case multiple scanners are used with the same laser, further Y coordinates can be addressed with the parameter.

Z

Indicates the depth for volumetric or stereoscopic applications. Octet values are interpreted as left aligned signed numbers in range [-128 .. -1, 0, 1 .. 127]. This tag can be followed by precision tags to extend the numeric range to 16 bit or higher. The most negative number corresponds to a signal level of -10V on the ISP-DB25 connector and SHALL represent rear/ far from the viewer, the most positive number corresponds to a signal level of +10V on the ISP-DB25 connector and SHALL represent front / toward the viewer. A value of 0 SHALL represent neutral distance such as the plane of the screen.

A tag parameter value of 0 SHALL address the standard scanner which corresponds to the signal 'Z' in the ISP-DB25 standard. In case multiple scanners are used with the same laser, further Z coordinates can be addressed with the parameter.

3.4.8 Category 5, Subcategory 0 - 3 Tags

Color command descriptors

CAT/SUB	ID	PRM	
0101.00ww	wwww	wwww	Color

Controls the intensity of a single color line with the passed 10 bit wavelength and corresponds to the signals 'R', 'G', 'B', 'Deep Blue', 'Yellow' and 'Cyan' of the ISP-DB25 standard. Octet values are interpreted as left aligned unsigned numbers in range [0 .. 255]. This tag can be followed by precision tags to extend the numeric range to 16 bit or higher. A

value of 0 corresponds to a signal level of 0V on the ISP-DB25 connector and SHALL produce no output, the most positive number corresponds to a signal level of +5V on the ISP-DB25 connector and SHALL produce maximum output. To improve the resolution of single octet based color spaces, the range can be shifted using the CSCL bits.

The coding of this tag varies from the standard scheme. Since the wavelength of the color is passed, 10 parameter bits are needed. This is accomplished by spanning 4 subcategories and using the identifier field. In case the wavelength is set to a value of 0, this tag must be directly preceded with a wavelength prefix to read the wavelength from the sample stream.

Since this standard is to define the streaming and storage of content, it focuses on representing this content as good as possible. For colors done by lasers, this means working with single color lines of a specific wavelength and their intensities. Producers may ask the projector for installed colors but in case of mismatches between wavelength requested by the content and available wavelength, the projector SHALL solve the conflict by finding a closest match or transforming the content color space into its device color space. This operation may be combined with processing of color balance and linearisation.

3.4.9 Category 5, Subcategory 12 Tags

Attribute command descriptors

CAT/SUB	ID	PRM	
0101.1100	0000	0000	Wavelength Prefix
0101.1100	0001	0000	Intensity
0101.1100	0010	0000	Beam-Brush

Wavelength Prefix

This tag can be used to dynamically parameterize the decoder with wavelength values from the sample stream. It is used as a prefix of a color command tag that has it's wavelength parameter set to 0. The associated octet is interpreted as an unsigned number, left aligned in a 10 bit word to represent wavelengths from 0 to 1023 nm in steps of 4 nm. If needed, this tag can be followed by precision tags to extend the numeric range to 16 bit, resulting in a unsigned 10:6 fixed-point number.

Intensity

This tag is included for the support of legacy ISP-DB25 systems. Besides the use as a separate signal for blanking, monochromatic projectors can be wired such that they just

operate on the intensity signal. It is currently unclear in which way a separate intensity signal can make sense with IDN streams since IDN systems would send color tags describing the intensities of color lines. Therefore, intensity handling is separate and SHALL be only used to control legacy systems.

This tag corresponds to the signal 'Intensity' of the ISP-DB25 standard. Octet values are interpreted as left aligned unsigned numbers in range [0 .. 255]. This tag can be followed by precision tags to extend the numeric range to 16 bit or higher. A value of 0 corresponds to a signal level of 0V on the ISP-DB25 connector and indicates zero intensity and that the beam is fully blanked. The most positive number corresponds to a signal level of +5V on the ISP-DB25 connector and indicates full intensity and that the beam is not blanked. To improve the resolution for single octet representations, the range can be moved using the ISCL bits.

Beam-Brush

This tag controls the beam diameter and corresponds to the signal 'Beam-Brush' of the ISP-DB25 standard. Octet values are interpreted as left aligned unsigned numbers in range [0 .. 255]. This tag can be followed by precision tags to extend the numeric range to 16 bit or higher. A value of 0 corresponds to a signal level of 0V on the ISP-DB25 connector and SHALL produce the smallest beam diameter, the most positive number corresponds to a signal level of +5V on the ISP-DB25 connector and SHALL produce the maximum beam diameter.

3.4.10 ISP-DB25 Backwards Compatibility Tags

Defining a digital complement of the analog ISP-DB25 standard implies that all signals can be converted back and forth between their analog and their digital representations. Although, by moving to a digital stream, meta information can be passed that are not available on the ISP-DB25 connector, specifically the wavelengths of the colors and the usage of the user defined signals. To accomplish the conversion, this standard defines a default usage of the ISP-DB25 connector that SHALL be used in case the user didn't modify settings.

0x4101	Optional, moves shutter control to sample stream
0x5C10	Optional, intensity/blanking
0x527E	Red, 638 nm
0x5214	Green, 532 nm
0x51CC	Blue, 460 nm
0x51BD	Optional(U1), used as deep blue, 445 nm

0x5241	Optional(U2), used as yellow, 577 nm
0x51E8	Optional(U3), used as cyan, 488 nm
0x4200	X
0x4010	16 bit precision
0x4210	Y
0x4010	16 bit precision
0x4201	Optional(U4), used as X-prime
0x4010	Optional(U4), 16 bit precision

This set of tags defines the default mapping of the entire ISP-DB25 connector into a digital data stream to be used with IDN laser projector services.

For devices sampling the waveforms on the ISP-DB25 connector, the default setting SHALL be a sampling frequency of 100kHz with 100 samples per IDN channel message with the above tags and signal assignments resulting in one message per millisecond of content.

3.4.11 Tags for IDTF with X/Y/R/G/B Projectors

The discrete graphic mode of the laser projector service can be used to handle images in IDTF files. These files contain draw coordinates (X, Y, Z) and colors (R, G, B). Blanking is to be handled such that the intensities of all color lines are set to 0. IDTF specifies 16 bit signed integer coordinates. This is accomplished by adding a precision octet to the coordinates in IDN. The orientation is the same in both standards. Meta information for colors is missing in IDTF. To accomplish the conversion, this standard defines a set of default wavelengths that SHALL be used in case the user didn't modify settings.

0x4200	X
0x4010	16 bit precision
0x4210	Y
0x4010	16 bit precision
0x527E	Red, 638 nm
0x5214	Green, 532 nm
0x51CC	Blue, 460 nm

In addition, a linearity tag that indicates visual linearity SHOULD be added since the IDTF standard defines colors to be visually linear. Although, transformation filters may be required since the laser projector may not be able to transform visual linearity into it's native color space.

3.5 Laser Effects

Effects have been an important part of laser shows since their beginnings. There are many visually stunning patterns and textures that can be produced with laser effects which are not achievable in other mediums. Examples of common effects are lumia and diffraction gratings.

Laser projector effect modes refer to the ISP-DMX and ISP-EFX standards. IDN just adds an interface layer. ISP-DMX channel assignments are retained but the DMX512 protocol and hardware got optional. Depending on the implementation, the IDN layer for laser effects can either address an intermediate DMX512 output interface to connect to standard DMX512 hardware or can address a software demultiplexer which dispatches the data to device specific hardware interfaces.

With regard to the IDN protocol, service modes for laser projector effects and service modes for DMX512 may share the same implementation. This is because of the close relationship and due to the required compatibility. As an example, in object oriented implementations, effect modes and DMX512 modes could be derived from the same ancestor classes.

3.6 Continuous Effects Mode

Please refer to the DMX512 service in continuous mode. The separate service mode identifier has been introduced to distinguish laser projector effect traffic from ordinary DMX512 traffic.

Please note that this mode is used for transparent copies of the DMX512 data stream which means that all DMX512 octets are present. This includes the start code in the first octet of the DMX512 packet. For compatibility, a start code of 0 SHALL address the laser effects. Referring the ISP-DMX standard, Channel 1 of the laser effects is located on dimmer level 1, in the second octet of the DMX512 packet and so on.

3.7 Discrete Effects Mode

Please refer to the DMX512 service in discrete mode. The separate service mode identifier has been introduced to distinguish laser projector effect traffic from ordinary DMX512 traffic.

4 DMX512 Service

The DMX512 standard consists of an interface specification and the representation of data. The IDN DMX512 service detaches both and moves the data into IDN channel messages. This allows for transparent bridges, recording, playback and generation of data without using actual DMX512 hardware. All characteristics of the DMX512 standard with regard to the data representation are kept.

Although it is a useful feature to provide a separate, unassigned DMX512 output for use with local fog machines, wind, screens and the like, the main reason for including the DMX512 service into the IDN standard is the close relationship to the laser projector service in effects mode. Both SHALL be compatible. This means that in case an interface converts DMX512 signals into an IDN data stream it SHALL be able to either address a laser projector service in effects mode or a DMX512 service.

4.1 Continuous Mode

This service mode transmits a continuous stream of delimited octet sequences. The stream is a transparent copy of the UART stream transmitted by DMX512 interfaces. Each sequence starts with the DMX512 start code octet and stops at the delimiter which is derived from the DMX512 break character.

This mode does not store or repeat DMX512 packets and the message timing SHALL comply with the DMX512 standard. This means that the timestamp interval of consecutive channel messages SHALL be derived from the amount of octets, DMX512 baud rate, break characters and mark times.

Because of the tracking of the octet sequence, streaming is exclusive on a per channel base. This means that the octet sequences on the channel form one uninterrupted stream of octets.

Sophisticated implementations may, since the nature of DMX512 data is packet oriented, be able to either merge multiple continuous mode streams or merge data from discrete mode streams into the continuous mode stream. This however is not required for basic implementations.

Accepted data chunks:

- Octet Segment

4.2 Continuous Mode Configuration

There is no configuration for the DMX512 service in continuous mode defined in this version of the standard. Consumers SHALL ignore the channel in case service configuration data is passed with the channel configuration.

4.3 Discrete Mode

This service mode transmits dimmer level sets that contain complete or partial data sections of DMX512 start code 0 packets. These level sets are processed one after another as they are scheduled. Only one set can be active per IDN channel at a time and subsequent sets replace older ones. Depending on the implementation, multiple discrete mode channels may be supported simultaneously. All levels are merged into a single background DMX512 start code 0 packet buffer which is repetitively sent to the DMX512 output hardware.

Dimmer level sets SHALL not overlap since results may be unpredictable. Overlapping could happen by using multiple channels and/or level subsets.

Sophisticated implementation may implement a mask of processed levels when setting up the start code 0 level buffer. This mask can then be used to merge discrete mode buffers into continuous mode streams.

Pure discrete mode operation may allow octet strings to be inserted between the repetitive start code 0 packets. The first octet of these strings is interpreted as the DMX512 start code of the packet. This start code SHALL not be 0 since this would be in conflict with the repetitive packets.

Accepted data chunks:

- Dimmer Levels
- Octet String (optional)

4.4 Discrete Mode Configuration

Basic DMX512 level set processing does not need a service mode configuration. In this default case, the first octet in the data chunk corresponds to the first level and the last octet in the data chunk corresponds to the last level in the DMX512 packet.

Advanced applications however may for example want to distribute the transmission of DMX512 levels across multiple IDN channels, each transmitting a subset of levels that are mapped to specific DMX512 devices.

Consumers SHALL be aware of the configuration and since discrete mode configuration is optional, either have it implemented or ignore the channel in case it can't be handled.

The service mode configuration is a list of single octet tags that may be followed by data octets. These tags configure decoder properties or describe octets in the data chunk. The codes are structured along the two nibbles of the octet (a nibble is a unit of 4 bits).

Bit	0	1	2	3	4	5	6	7
	Identifier				Parameter			

Tag Identifier (ID) / Parameter (PRM)

Each decoder function has an identifier. Depending on the function, the parameter can be used as an argument or can be used to specify the amount of function data octets that follow the tag octet.

Identifier		Parameter
0	Void	Data octet count
4	Dimmer Level Subset	Data octet count

Currently, the service mode configuration can affect dimmer levels data chunks only. There are no tags that affect the processing of octet string data chunks.

Function identifiers 12 to 15 are reserved for dynamic tag assignment through session configuration and SHALL be disabled by default. These tags can be used for manufacturer specific implementations.

Void

This tag can be used to align the tag list to a 32 bit boundary required by the header section of the IDN channel message. The parameter is used to pass the amount of data octets that SHALL be skipped when processing the tag.

Dimmer Level Subset

A dimmer levels data chunk may either contain a contiguous set of levels for which a base level in the DMX512 packet can be specified or contain multiple consecutive subsets for which a level count and a base level in the DMX512 packet can be specified.

The most basic application is the contiguous set of levels with no service mode configuration. This maps levels 1:1 into the DMX512 packet. Second, a dimmer level subset tag (PRM=2) can be used to move this contiguous set to a different start point in the packet. Third, multiple

dimmer level subset tags (PRM=3) can be used to split up the contiguous set into multiple subsets that can be mapped to individual start points in the packet.

The tag parameter is used to set the mapping function and to specify the amount of data octets that follow the tag octet. Parameter values other than the ones listed here are not valid.

PRM \ Octet	0	1	2	3
2	Base			
3	Base		Count	

Only one PRM=2 tag is allowed to occur in the configuration since this maps the whole level set to the new start point.

Multiple PRM=3 tags are allowed to occur in the configuration. The subsets SHALL be mapped such that no overlapping occurs. The first tag describes the first subset and all subsets are stored consecutively in dimmer levels data chunks. The amount of dimmer level octets in the chunk SHALL match the total amount of octets mapped through subset tags.

Mixing PRM=2 and PRM=3 tags is strongly discouraged. Semantically, multiple PRM=3 tags could be followed by one PRM=2 tag.

Base

The DMX512 level number in the start code 0 output packet which the first octet in the subset is mapped to. The number SHALL be in range 1 to 512 and Base + Count SHALL be less than or equal 513. This number is 1-based to accord to the DMX512 standard. A special base value of 0xFFFF SHALL be used to ignore the subset. This may be useful for testing purposes.

Count

The number of octets in the subset. The number SHALL be in range 1 to 255 and Base + Count SHALL be less than or equal 513. Multiple subset tags can be used in case more than 255 octets are to be mapped.

5 Data Chunks

Data chunks carry channel data directed to services and are contained in the data section of channel messages. In case of fragmentation, the chunks can be split and spread across data sections of multiple messages.

5.1 Laser Projector Wave Samples

This data chunk contains samples for a finite time interval of a waveform targeting laser graphic scanners. In order to be able to encode a gapless waveform of arbitrary length, multiple subsequent chunks have to be encoded.

The amount of octets per sample must match with the channel configuration. A sample descriptor tag must be present for each octet of the sample and the order of descriptor tags and octets of the sample must match.

The array SHOULD contain a reasonable amount of samples but SHALL contain at least 20 samples in order to allow consumers to compensate the chunk processing overhead. The size of the sample array SHOULD be chosen such, that it adapts to the medium. For IP on Ethernet for example it is desirable to use the maximum segment size and to avoid IP fragmentation.

The data chunk starts with a header and is followed by an array of samples.

Octet	0	1	2	3
0	Flags	Duration		
4...	Samples			

Flags

Processing flags for the chunk. The upper four bits of this octet are used for service mode related purposes (please refer also to section 5.6) while the lower four bits of this octet are used for processing options of the chunk.

Bit	0	1	2	3	4	5	6	7
	00		SCM		0000			

Duration

Duration of the interval in microseconds. Duration and message timestamp SHALL be used for validation. This means that the timestamp of the current message plus the duration must be equal to the timestamp of the next wave sample message on a channel. Wraps must be taken care of on consumer side.

To most accurately match sample frequencies, the sample count and the duration can both be adjusted. The frequency equals the count divided by the duration.

5.2 Laser Projector Frame Samples

This data chunk contains samples for a frame targeting laser graphic scanners.

The amount of octets per sample must match with the channel configuration. A sample descriptor tag must be present for each octet of the sample and the order of descriptor tags and octets of the sample must match.

The first sample is interpreted as the start point of the shape and SHALL be invisible. It can be used to move the draw cursor. The remaining samples define vertex coordinates and draw attributes of the segments that are to be drawn. The last sample contains the end point of the shape. A consumer calculated hidden movement SHALL be inserted in case a frame with differing start point and end point is to be repeated or in case the end point of the previous frame is different from the start point of the next frame. Likewise, in case start point and end point are same, a hidden bend SHOULD be inserted in case the first segment and the last segment are not continuous.

An empty sample array indicates an empty frame and can be used to void the frame buffer in case of using repetitive frames. If not empty, the array SHALL contain at least 2 samples (the start point and the end point).

The data chunk starts with a header and is followed by an array of samples.

Octet	0	1	2	3
0	Flags	Duration		
4...	Samples			

Flags

Processing flags for the chunk. The upper four bits of this octet are used for service mode related purposes (please refer also to section 5.6) while the lower four bits of this octet are used for processing options of the chunk.

Bit	0	1	2	3	4	5	6	7
	00		SCM					Once

Once

In case this bit is set, the frame SHALL be drawn exactly once. When done with the frame and no further frame is scheduled, the graphic scanners SHALL be moved to home position (0, 0) and SHALL wait for the next frame. This can be used to precisely control frame playback and avoid doubly scanned frames.

Duration

Duration of the frame in microseconds. The first sample SHALL not be taken into account when calculating the sample timing because scanner movements take place on the segments connecting the samples.

5.3 Octet Segment

This data chunk contains a delimited sequence of octets or part of it in case the sequence is split into multiple segments. The sequence can contain binary values, settings, commands or ASCII text to be transmitted to suitable services.

The splitting of the sequence into multiple segments with each segment being packed into a separate IDN channel message is useful for reducing the latency in case of realtime streaming when IDN is used to tunnel slower UART-based protocols.

The chunk starts with a header which contains sequencing and reassembly information and is followed by an array of octets.

Octet	0	1	2	3
0	Flags	Sequence	Offset	
4...	Octets			

Flags

Processing flags for the chunk. The upper four bits of this octet are used for service mode related purposes (please refer also to section 5.6) while the lower four bits of this octet are used for processing options of the chunk.

Bit	0	1	2	3	4	5	6	7
	00		SCM					DLIM

Delimiter (DLIM)

The delimiter marks the end of the sequence. The segment that contains the last octet of the sequence SHALL be marked by setting this bit. Offset count and evaluation SHALL restart at 0 and the sequence counter SHALL be counted up thereafter.

Sequence

The sequence number. Producers SHALL count this number up for each subsequent sequence and consumers SHALL use the number for sequence tracking and lost sequence detection. In case a sequence is split into multiple segments, all segments share the same sequence number.

Offset

The zero-based offset of the first octet in the array with respect to the beginning of the sequence. The offset marks the position at which the sequence has been split in case it spans multiple segments. Consequently, the offset field stays 0 for sequences that are not split.

5.4 Octet String

This data chunk contains a sequence of octets (the string). This sequence can contain binary values, settings, commands or ASCII text to be transmitted to suitable services. The octet string is delimited by the end of the data chunk. Opposed to IDN octet segments, octet strings can not be split into multiple parts. Each string contains a discrete set of octets.

The chunk starts with a header, which is defined for future expansion and is followed by an arbitrary amount of octets.

Octet	0	1	2	3
0	Flags	0	0	0
4...	Octets			

Flags

Processing flags for the chunk. The upper four bits of this octet are used for service mode related purposes (please refer also to section 5.6) while the lower four bits of this octet are used for processing options of the chunk.

Bit	0	1	2	3	4	5	6	7
	00		SCM		0000			

5.5 Dimmer Levels

This data chunk contains dimmer level octets to be used with discrete DMX512 or similar service modes. Depending on the service mode configuration, the chunk may contain a contiguous set of levels that can be mapped to an arbitrary base level or may contain multiple level subsets. Default configuration SHALL treat the set contiguous and map the first octet to the first level.

The chunk starts with a header, which is defined for future expansion and is followed by an arbitrary amount of level octets.

Octet	0	1	2	3
0	Flags	0	0	0
4...	Dimmer Level Octets			

Flags

Processing flags for the chunk. The upper four bits of this octet are used for service mode related purposes (please refer also to section 5.6) while the lower four bits of this octet are used for processing options of the chunk.

Bit	0	1	2	3	4	5	6	7
	00		SCM		0000			

5.6 Generic Fields

This chapter introduces fields that are used in multiple structures.

Service configuration match (SCM)

Producers set these bits according to the value of the current channel configuration match bits SDM (Service Data Match, please refer also to section 2.2) and consumers compare the bits with the match bits received and stored with the latest channel configuration.

This is a cross check feature for unreliable connections or file recovery in case the message containing the configuration or the message that closed the channel is lost. In case of a mismatch, the consumer SHALL discard it's current service configuration, void the channel and wait for a new configuration.

6 Revision History

6.1 Revision 001, July 2015

The initial publication

Contributors (Revision 001)

Matthias Frank, University of Bonn
Horacio Pugliese, LaserNet
Andrew Berry, X-Laser
Tim Walsh, Laser Spectacles, Inc.
Dirk Apitz, DexLogic

