Juli 2020

DIN SPEC 15800

ICS 35.240.99; 97.200.10

Veranstaltungstechnik – General Device Type Format (GDTF); Text Englisch

Entertainment Technology – General Device Type Format (GDTF); Text in English

Technologies du spectacle – General Device Type Format (GDTF); Texte en anglais

Gesamtumfang 73 Seiten

DIN-Normenausschuss Veranstaltungstechnik, Bild und Film (NVBF)

www.din.de
www.beuth.de

3173040

Contents

Foreword

This DIN SPEC has been developed according to the PAS procedure. The development of a DIN SPEC according to the PAS procedure is carried out in DIN SPEC (PAS)-consortiums and does not require the participation of all stakeholders.

This document has been developed and adopted by the initiator(s) and authors named below:

— DeerSoft GmbH; Moritz Staffel;

— Lightpower GmbH; René Berhorst;

— MA Lighting Technology GmbH; Gerhard Krude, Daniel Müller;

— ROBE lighting s.r.o.; Josef Valchář, Petr Vanek;

— Vectorworks Incorporated; Jeremy Powell, Biplap Sarkar;

— Verband für Medien- und Veranstaltungstechnik (VPLT); Randell Greenlee.

At present, there are no standards covering this topic in the body of German Standards.

DIN SPEC (PAS)s are not part of the body of German Standards.

A draft of this DIN SPEC (PAS) has not been published.

Despite great efforts to ensure the accuracy, reliability and precision of technical and non-technical information, the DIN SPEC (PAS)-consortium cannot give any explicit or implicit assurance or warranty in respect of the accuracy of the document. Users of this document are hereby made aware that the consortium cannot be held liable for any damage or loss. The application of this DIN SPEC (PAS) does not release users from the responsibility for their own actions and is applied at their own risk.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. DIN shall not be held responsible for identifying any or all such patent rights.

Provision of this document free of charge as a PDF via the Beuth WebShop has been financed in advance.

For current information on this standard, please go to DIN's website (www.din.de) and search for the document number in question.

Introduction

Nowadays lighting fixtures (luminaires and other controllable devices) have become more and more complex. Additionally, the development of these devices has become faster than ever. New devices are designed with very complex structures and multiple instances, they have more complex color-mixing systems and mode dependencies. To give the user access to the enormous flexibility of the existing devices a way to provide the accurate Fixture Type data is needed to control and pre-visualize the particular devices as good as possible and as quickly as needed. GDTF is that measure. There are many different lighting consoles and software manufacturers on the market and all of them are using different ways and different file formats to get the fixture control information into their systems. As the development of new high-end fixtures takes place at an amazing speed, this creates a 'lack' of available control data on the side of the console and pre-visualization software manufacturers. Also, fixture manufacturers are often approached by their clients directly to support them with accurate fixture types. As there are so many different consoles and visualizers on the market this process requires vast knowledge of many different systems. Fixture manufacturers would need to understand how every console or visualizer works, and how to provide the required data. Moreover, a way of format description is needed that not only allows to provide all of the required control information, but also structures it already in a hierarchical way that follows the structure of the device to be described. The lighting designer who would like to use these devices has to deal with such obstacles. They often receive the device control data of a specific new fixture later than expected. Also, the data can be incomplete, because it was not created with the latest information needed from the manufacturer of the fixture. This very clearly demonstrates that our industry is missing a standardized way of defining the description of intelligent and complex devices.

This document defines a data format. After the DIN SPEC has been published, the format will continue to be developed further, but it is important to make an initial version publicly available. Topics for which no specifications can be made at this time, but for which it is foreseeable that this will be necessary, are therefore already specified in this DIN SPEC, but with the note that no specifications can be made at this time.

1 Scope

This document specifies the "General Device Type Format" (GDTF).

This document provides a unified way of listing and describing the hierarchical and logical structure and controls of any type of controllable device (e.g. luminaires, fog machines, etc.) in the lighting and entertainment industry. It will be used as a foundation for the exchange of device data between lighting consoles, CAD and 3D-pre-visualization applications. The purpose of an existing GDTF-file is to reflect the real-world physical components of the devices and to provide control based on this information. It contains and is derived from the 3D geometry (real world or virtual) of the device.

This document is only applicable for lighting systems and equipment used in the entertainment industry.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements 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.

ANSI E1.54-2015, *PLASA Standard for Color Communication in Entertainment Lighting*

ANSI/IES TM-30, *IES Method for Evaluating Light Source Color Rendition*

IEC 61966-2-1:1999, *Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB*

ISO 22028-2:2013, *Photography and graphic technology — Extended colour encodings for digital image storage, manipulation and interchange — Part 2: Reference output medium metric RGB colour image encoding (ROMM RGB)*

RFC 4122, *A Universally Unique IDentifier (UUID) URN Namespace*[1]

3 Terms and definitions

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

DIN and DKE maintain terminological databases for use in standardization at the following addresses:

— DIN-TERMinologieportal: available at https://www.din.de/go/din-term

— DKE-IEV: available at http://www.dke.de/DKE-IEV

3.1
GDTF
descriptive name of the specification and the acronym for General Device Type Format

3.2
Fixture Type Attribute
singular mutual exclusive control function

Note 1 to entry: Definitions of common attributes can be found in Annex A.

1 Available at: https://www.ietf.org/rfc/rfc4122.txt.

3.3
Activation Group
Attributes which need to be activated together to gain control over one logical function

Note 1 to entry: As example Pan & Tilt are paired to gain control over Position.

3.4
Feature
groups the Fixture Type Attributes into a structured way for easier access and search

3.5
Feature Groups
groups the logical control elements called Feature into a structured way for easier access and finding

3.6
DMXBreak
term used when a fixture needs more than one DMX start address

4 File Format Definition

To describe the device type, an uncompressed zip file with the extension "*.gdtf" is used. The archive shall contain a description XML file and resource files. Some of the resource files are located in a folder structure. There are two folders defined: "./wheels" and "./models". The folder "./models" has two subfolders for a better structural overview called "./models/3ds" and "./models/svg". The description.xml file contains the description of the device type and all DMX modes as well as all firmware revisions of the device.

./description.xml

./thumbnail.png

./thumbnail.svg

./wheels/gobo1.png

./wheels/gobo2.png

./models/3ds/base.3ds

./models/3ds/yoke.3ds

./models/svg/base.svg

./models/svg/yoke.svg

The ZIP archive name is specified as follows:
`ManufacturerName@FixtureTypeName@OptionalComment`

EXAMPLE `generic@led@comment`

UTF-8 has to be used to encode the XML file. Each XML file internally consists of XML nodes. Each XML node could have XML attributes and XML children. Each XML attribute has a value. If a XML attribute is not specified, the default value of this XML attribute will be used. If the XML attribute value is specified as a string, the format of the string will depend on the XML attribute type. All XML attribute types are specified in Table 1.

Table 1 — XML Attribute Value Types

Value Type	Format	Description
Uint	Integer	Unsigned integer
Int	Integer	Signed integer
Hex	Integer	Number in hexadecimal notation; Default value: 0
Float	float	Floating point numeric; Separator: "."
String	Literal	Text
Name	restricted Literal	Unique object names; The allowed characters are listed in Annex C Default value: object type with an index in parent.
Date	yyyy-mm-ddThh:mm:ss	Date and time corresponding to UTC +00:00 (Coordinated Universal Time): yyyy – year, mm – month, dd – day, hh – hours (24 format), mm – minutes, ss – seconds. Example: "2016-06-21T11:22:48"
Node	Name.Name.Name...	Link to an element: "Name" is the value of the attribute "Name" of a defined XML node. The starting point defines each attribute separately.
ColorCIE	floatx, floaty, floatY	CIE color representation xyY 1931
Matrix	{float,float,float,float} {float,float,float,float} {float,float,float,float} {float,float,float,float}	The transformation matrix consists 4 x 4 floats. Stored in a row-major order. For example, each row of the matrix is stored as a 4-component vector. The mathematical definition of the matrix is in a column-major order. For example, the matrix rotation is stored in the first three columns, and the translation is stored in the 4th column. The metric system consists of the Right-handed Cartesian Coordinates *XYZ*: *X* – from left (−*X*) to right (+*X*), *Y* – from the outside of the monitor (−*Y*) to the inside of the monitor (+*Y*), *Z* – from bottom (−*Z*) to top (+*Z*). 0,0,0 – center base.
Rotation	{float, float, float} {float, float, float} {float, float, float}	The Rotation matrix consists of 3*3 floats. Stored as **row-major** matrix, i.e. each row of the matrix is stored as a 3-component vector. Mathematical definition of the matrix is **column-major**, i.e. the matrix rotation is stored in the three columns. Metric system, right-handed Cartesian coordinates *XYZ*: *X* – from left (−*X*) to right (+*X*), *Y* – from the outside of the monitor (-*Y*) to the inside of the monitor (+*Y*), *Z* – from the bottom (−*Z*) to the top (+*Z*).
Enum	Literal	Possible values are predefined.
DMXAddress	Int, Alternative format: Universe.Address	Absolute DMX address (size 4 bytes); Alternative format: Universe – integer universe number, starting with 1; Address: address within universe from 1 to 512. Format: integer

Value Type	Format	Description
DMXValue	Uint/n for ByteMirroring values Uint/ns for ByteShifting values	Special type to define DMX value where n is the byte count. The byte count can be individually specified without depending on the resolution of the DMX Channel. By default byte mirroring is used for the conversion. So 255/1 in a 16 bit channel will result in 65535. You can use the byte shifting operator to use byte shifting for the conversion. So 255/1s in a 16 bit channel will result in 65280.
GUID	XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX	Unique ID corresponding to RFC 4122: X–1 digit in hexadecimal notation. Example: "308EA87D-7164-42DE-8106-A6D273F57A51".
Resource	String	File name of the resource file without extension and without subfolder.
Pixel	Pixel	Integer value representing one Pixel inside a MediaFile. Pixel count starts with zero in the top left corner.

The first XML node is always the XML description node:

```
<?xml version="1.0" encoding="UTF-8"?>
```

The second XML node is the GDTF node. The attribute of this node is the DataVersion (see Table 2):

```
<GDTF DataVersion="1.1">
```

The example above shows the XML node for the GDTF version 1.1.

Table 2 — GDTF Node Attributes

XML Attribute Name	Value Type	Description
DataVersion	*Uint.Unit*	The DataVersion attribute defines the minimal version of compatibility. The Version format is “Major.Minor”, where major and minor is Uint with size 1 byte.

5 Fixture Type Node

The FixtureType node is the starting point of the description of the fixture type within the XML file. The defined Fixture Type Node attributes of the fixture type are specified in Table 3.

Table 3 — Fixture Type Node Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	Name of the fixture type.
ShortName	*String*	Shortened name of the fixture type.
LongName	*String*	Detailed name of the fixture type.
Manufacturer	*String*	Manufacturer of the fixture type.

XML Attribute Name	Value Type	Description
Description	*String*	Description of the fixture type.
FixtureTypeID	*GUID*	Unique number of the fixture type.
Thumbnail	*Resource*	Optional; File name without extension containing description of the thumbnail. Use the following as a resource file: — png file to provide the rasterized picture. Maximum resolution of picture: 1 024 x 1 024; — svg file to provide the vector graphic. These resource files are located in the root directory of the zip file. Default value: empty.
RefFT	*GUID*	GUID of the referenced fixture type
CanHaveChildren	*Enum*	Describes if it is possible to mount other devices to this device. Value: “Yes”, “No”. Default value: “Yes”

Fixture type node children are specified in Table 4.

Table 4 — Fixture Type Node Children

Child Node	Mandatory	Description
AttributeDefinitions	Yes	Defines all Fixture Type Attributes that are used in the fixture type.
Wheels	No	Defines the physical or virtual color wheels, gobo wheels, media server content and others.
PhysicalDescriptions	No	Contains additional physical descriptions.
Models	No	Contains models of physically separated parts of the device.
Geometries	Yes	Describes physically separated parts of the device.
DMXModes	Yes	Contains descriptions of the DMX modes.
Revisions	No	Describe the history of the fixture type.
FTPresets	No	Is used to transfer user-defined and fixture type specific presets to other show files.
Protocols	No	Specifies supported protocols.

One or more sections could be empty or missing, but the order of sections is mandatory as specified in *Table 4*.

6 Attribute Definitions

This section defines the attribute definitions for the Fixture Type Attributes.

NOTE 1 More information on the definitions of attributes can be found in Annex A “Attribute Definitions”.

NOTE 2 All currently defined Fixture Type Attributes can be found in Annex B “Attribute Listing”.

NOTE 3 All currently defined Activation Groups can be found in Annex B "Attribute Listing".

NOTE 4 All currently defined Feature Groups can be found in Annex B "Attribute Listing".

The current attribute definition node does not have any XML attributes (XML node `AttributeDefinitions`). Children of the attribute definition are specified in *Table 5*.

Table 5 — Attribute Definition Children

XML node	Mandatory	Description
ActivationGroups	No	Defines which attributes are to be activated together. For example, Pan and Tilt are in the same activation group.
FeatureGroups	Yes	Describes the logical grouping of attributes. For example, Gobo 1 and Gobo 2 are grouped in the feature Gobo of the feature group Gobo.
Attributes	Yes	List of Fixture Type Attributes that are used. Predefindes fixtury type attributes can be found in Annex A.

6.1 Activation Groups

6.1.1 General

This section defines groups of Fixture Type Attributes that are intended to be used together.

Example: Usually Pan and Tilt are Fixture Type Attributes that shall be activated together to be able to store and recreate any position.

The current activation groups node does not have any XML attributes (XML node `ActivationGroups`). As children, it can have a list of *activation group*.

6.1.2 Activation Group

This section defines the activation group Attributes (XML node `ActivationGroup`). Currently defined XML attributes of the activation group are specified in Table 6.

Table 6 — Activation Group Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the activation group.

The activation group does not have any children.

6.2 Feature Groups

6.2.1 General

This section defines the logical grouping of Fixture Type Attributes (XML node `FeatureGroups`).

NOTE 1 A feature group can contain more than one logical control unit.

A feature group Position shall contain PanTilt and XYZ as separate Feature.

NOTE 2 Usually Pan and Tilt create a logical unit to enable position control, so it is necessary to group them in a Feature PanTilt.

As children, the Feature Groups has a list of a *feature group*.

6.2.2 Feature Group

6.2.2.1 General

This section defines the feature group (XML node `FeatureGroup`). The currently defined XML attributes of the feature group are specified in Table 7.

Table 7 — Feature Group Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the feature group.
Pretty	*String*	The pretty name of the feature group.

As children, the Feature Group has a list of a *feature*.

6.2.2.2 Feature

This section defines the feature (XML node `Feature`). The currently defined XML attributes of the feature are specified in Table 8.

Table 8 — Feature Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the feature.

The feature does not have any children.

6.3 Attributes

6.3.1 General

This section defines the Fixture Type Attributes (XML node `Attributes`). As children, the Attributes has a list of a *attribute*.

6.3.2 Attribute

This section defines the Fixture Type Attribute (XML node `Attribute`). The currently defined XML attributes of the attribute Node are specified in Table 9.

Table 9 — XML Attributes of the Attribute Node

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the attribute.
Pretty	*String*	The pretty name of the attribute.
ActivationGroup	*Node*	Optional link to the activation group. The starting point is the *activation groups* node.
Feature	*Node*	Link to the corresponding feature. The starting point is the *feature groups* node.

XML Attribute Name	Value Type	Description
MainAttribute	*Node*	Optional link to the main attribute. The starting point is the *attribute* node.
PhysicalUnit	*Enum*	The currently defined unit values are: “None”, “Percent”, “Length” (m), “Mass” (kg), “Time” (s), “Temperature” (K), “LuminousIntensity”(cd), “Angle” (degree), “Force” (N), “Frequency” (Hz), “Current” (A), “Voltage” (V), “Power” (W), “Energy” (J), “Area” (m^2), “Volume” (m^3), “Speed” (m/s), “Acceleration” (m/s^2), “AngularSpeed” (degree/s), “AngularAccc” ($degree/s^2$), “WaveLength” (nm), “ColorComponent”. Default: “None”.
Color	*ColorCIE*	Optional: Defines the color for the attribute.

The attribute Node does not have any children.

7 Wheel Collect

7.1 General

This section defines all physical or virtual wheels of the device. It currently does not have any XML attributes (XML node `Wheels`). As children, Wheel Collect has a list of a *wheel*.

NOTE Physical or virtual wheels represent the changes to the light beam within the device. Typically color, gobo, prism, animation, content and others are described by wheels.

7.2 Wheel

7.2.1 General

Each wheel describes a single physical or virtual wheel of the fixture type. If the real device has wheels you can change, then all wheel configurations have to be described. Wheel has the following XML node: `Wheel`. The currently defined XML attributes of the wheel are specified in Table 10.

Table 10 — Wheel Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the wheel

As children, Wheel have a list of a *wheel slot*.

7.2.2 Wheel Slot

7.2.2.1 General

The wheel slot represents the slot on the wheel (XML node `Slot`). The currently defined XML attributes of the wheel slot are specified in Table 11.

Table 11 — Wheel Slot Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the wheel slot
Color	*ColorCIE*	Color of the wheel slot, Default value: {0.3127, 0.3290, 100.0} (white) For Y give relative value compared to overall output defined in property Luminous Flux of related Beam Geometry (transmissive case).
Filter	*Node*	Optional link to filter in the physical description; Do not define color if filter is used; Starting point: Filter Collect
MediaFileName	*Resource*	Optional; PNG file name without extension containing image for specific gobos etc. — Maximum resolution of picture: 1 024x1 024; — Recommended resolution of gobo: 256x256; — Recommended resolution of animation wheel: 256x256 These resource files are located in a folder called `./wheels` in the zip archive. Default value: empty.

NOTE More information on the definitions of images used in wheel slots to visualize gobos, animation wheels or color wheels can be found in Annex E "Wheel Slot Image Definition".

The link between a slot and a *channel set* is done via the wheel slot index. The wheel slot index of a slot is derived from the order of a wheel's slots. The wheel slot index is normalized to 1.

If the wheel slot has a prism, it has to have one or several children called *prism facet*. If the wheel slot has an AnimationWheel, it has to have one child called *Animation Wheel*.

7.2.2.2 Prism Facet

This section can only be defined for the prism wheel slot and has a description of the prism facet (XML node `Facet`). The currently defined XML attributes of the prism facet are specified in Table 12.

Table 12 — Wheel Slot Attributes

XML Attribute Name	Value Type	Description
Color	*ColorCIE*	Color of prism facet, Default value: {0.3127, 0.3290, 100.0} (white)
Rotation	*Rotation*	Specify the rotation, translation and scaling for the facet.

The prism facet cannot have any children.

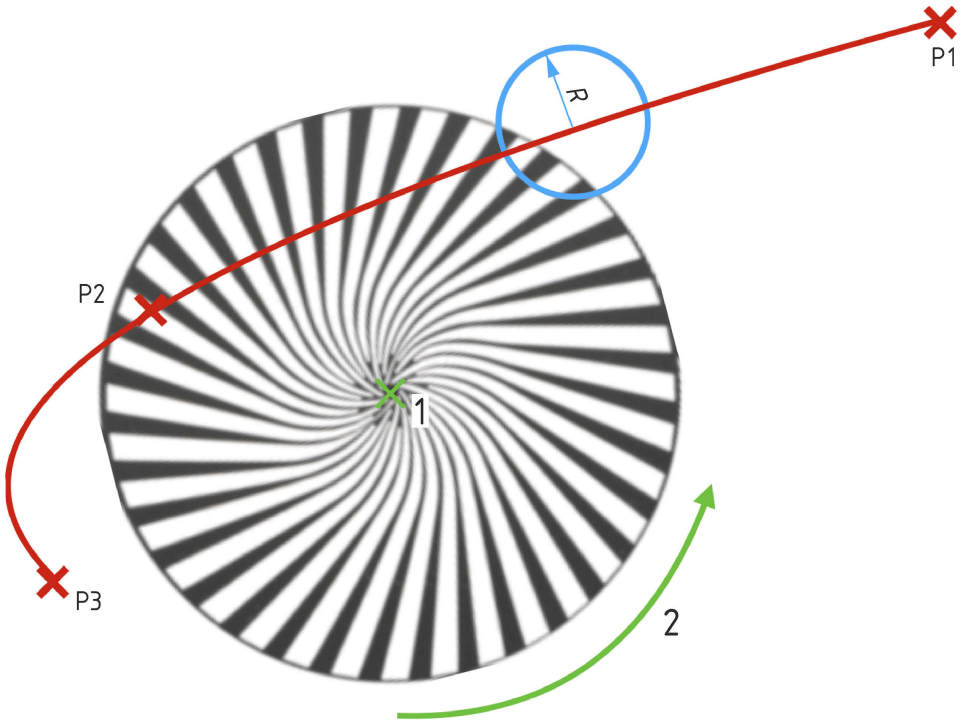
7.2.2.3 Animation System

This section can only be defined for the animation system disk and it describes the animation system behavior (XML node `AnimationSystem`). The currently defined XML attributes of the AnimationSystem are specified in *Table 13*. An example of AnimationSystem is given in Figure 1.

Table 13 — AnimationSystem Attributes

XML Attribute Name	Value Type	Description
P1	*Array of Pixel*	First Point of the Spline describing the path of animation system in the beam in relation to the middle of the Media File; Array of two floats; Seperator of values is ","; First Pixel is X-axis and second is Y-axis.
P2	*Array of Pixel*	Second Point of the Spline describing the path of animation system in the beam in relation to the middle of the Media File; Array of two floats; Seperator of values is ","; First Pixel is X-axis and second is Y-axis.
P3	*Array of Pixel*	Third Point of the Spline describing the path of animation system in the beam in relation to the middle of the Media File; Array of two floats; Seperator of values is ","; First Pixel is X-axis and second is Y-axis.
Radius	*Pixel*	Radius of the circle that defines the section of the Animation system which will be shown in the beam.

The AnimationSystem cannot have any children.



Key

1	Rotation center	R	Radius
2	Rotation orientation	P1, P2, P3	Points

Figure 1 — Example of AnimationSystem

8 Physical Descriptions

8.1 General

This section describes the physical constitution of the device. It currently does not have any XML Attributes (XML node `PhysicalDescriptions`). Children of Physical Description are specified in *Table 14*.

Table 14 — Physical Description Children

XML node	Mandatory	Description
Emitters	No	Describes device emitters
Filters	No	Describes device filters
ColorSpace	No	Describes device color space
DMXProfiles	No	Describes nonlinear correlation between DMX input and physical output of a channel.
CRIs	No	Describes color rendering according to ANSI/IES TM-30 (99 color samples).
Connectors	No	Describes physical connectors of the device.
Properties	No	Describes physical properties of the device.

8.2 Emitter Collect

8.2.1 General

This section contains the description of the emitters. Emitter Collect defines additive mixing of light sources, such as LEDs and tungsten lamps with permanently fitted filters. It currently does not have any XML Attributes (XML node `Emitters`). As children, the Emitter Collect has a list of a *emitter*.

8.2.2 Emitter

This section defines the description of the emitter (XML node `Emitter`). The currently defined XML attributes of the emitter are specified in Table 15.

Table 15 — Emitter Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	Unique Name of the emitter
Color	*ColorCIE*	Approximate absolute color point if applicable. Omit for non-visible emitters (e.g., UV). For Y give relative value compared to overall output defined in property Luminous Flux of related Beam Geometry (transmissive case).
DominantWaveLength	*Float*	Required if color is omitted, otherwise it is optional. Dominant wavelength of the LED.
DiodePart	*String*	Optional; Manufacturer's part number of the diode.

As children, the Emitter Collect has a list of a *measurement*.

8.3 Filter Collect

8.3.1 General

This section contains the description of the filters. The Filter Collect defines subtractive mixing of light sources by filters, such as subtractive mixing flags and media used in physical or virtual Color Wheels. It currently does not have any XML Attributes (XML node `Filters`). As children, the Filter Collect has a list of a *filter*.

8.3.2 Filter

This section defines the description of the filter (XML node `Filter`). The currently defined XML attributes of the filter are specified in Table 16.

Table 16 — Filter Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	Unique Name of the filter.
Color	*ColorCIE*	Approximate absolute color point when this filter is the only item fully inserted into the beam and the fixture is at maximum intensity. For Y give relative value compared to overall output defined in property Luminous Flux of related Beam Geometry (transmissive case).

As children, the Filter Collect has a list of a *measurement*.

8.4 Measurement

8.4.1 General

The measurement defines the relation between the requested output by a control channel and the physically achieved intensity. XML node for measurement is `Measurement`. The currently defined XML attributes of the measurement are specified in Table 17.

Table 17 — Measurement Attributes

XML Attribute Name	Value Type	Description
Physical	*Float*	For additive color mixing: uniquely given emitter intensity DMX percentage. Value range between > 0 and ≤ 100. For subtractive color mixing: uniquely given flag insertion DMX percentage. Value range between 0 and 100.
LuminousIntensity	*Float*	Used for additive color mixing: overall candela value for the enclosed set of measurements.
Transmission	*Float*	Used for subtractive color mixing: total amount of lighting energy passed at this insertion percentage.
InterpolationTo	*Enum*	Interpolation scheme from the previous value. The currently defined values are: "Linear", "Step", "Log"; Default: Linear

The order of the measurements corresponds to their ascending physical values.

Additional definition for additive color mixing: It is assumed that the physical value 0 exists and has zero output.

Additional definition for subtractive color mixing: The flag is removed with physical value 0 and it does not affect the beam. Physical value 100 is maximally inserted and affects the beam.

NOTE Some fixtures can vary in color response. These fixtures define multiple measurement points and corresponding interpolations.

As children, the Measurement Collect has an optional list of a *measurement point*.

8.4.2 Measurement Point

The measurement point defines the energy of a specific wavelength of a spectrum. The XML node for measurement point is `MeasurementPoint`. The defined XML attributes of the measurement points are specified in Table 18.

It is recommended, but not required, that measurement points are evenly spaced.

Regions with minimal light energy can be omitted, but the decisive range of spectrum must be included. Recommended measurement spacing is 1 nm. Measurement spacing should not exceed 4 nm.

Table 18 — Measurement Point Attributes

XML Attribute Name	Value Type	Description
WaveLength	*Float*	Center wavelength of measurement (nm).
Energy	*Float*	Lighting energy ($W/m^2/nm$).

The measurement point does not have any children.

8.5 Color Space

This section defines the color space that is used for color mixing with indirect RGB, Hue/Sat, xyY or CMY control input. (XML node `ColorSpace`). The currently defined XML attributes of the color space are specified in Table 19.

Table 19 — Color Space Attributes

XML Attribute Name	Value Type	Description
Mode	*Enum*	Definition of the Color Space that used for the indirect color mixing. The defined values are "Custom", "sRGB", "ProPhoto" and "ANSI". Default Value: "sRGB"
Red	*ColorCIE*	Optional; CIE xyY of the Red Primary; this is used only if the ColorSpace is "Custom".
Green	*ColorCIE*	Optional; CIE xyY of the Green Primary; this is used only if the ColorSpace is "Custom".
Blue	*ColorCIE*	Optional; CIE xyY of the Blue Primary; this is used only if the ColorSpace is "Custom".
WhitePoint	*ColorCIE*	Optional; CIE xyY of the White Point; this is used only if the ColorSpace is "Custom".

The predefined modes for the color space XML Attributes are specified in Table 20.

Table 20 — Predefined Modes for Color Space Attribute Mode

Mode	sRGB	ProPhoto	ANSI
Description	Adobe sRGB, HDTV IEC 61966-2-1:1999	Kodak ProPhoto ROMM RGB ISO 22028-2:2013	ANSI E1.54-2015
Red	0.6400, 0.3300, 0.2126	0.7347, 0.2653	0.7347, 0.2653
Green	0.3000, 0.6000, 0.7152	0.1596, 0.8404	0.1596, 0.8404
Blue	0.1500, 0.0600, 0.0722	0.0366, 0.0001	0.0366, 0.001
WhitePoint	0.3127, 0.3290, 1.0000	0.3457, 0.3585	0.4254, 0.4044

The color space does not have any children.

8.6 DMX Profile Collect

8.6.1 General

This section defines DMX profile descriptions. Currently it does not have any XML attributes (XML node `DMXProfiles`). As children, DMX profile collect has a list of a *DMX profile*.

8.6.2 DMX Profile

This section defines the DMX profile description (XML node `DMXProfile`).

8.7 Color Rendering Index Collect

8.7.1 General

This section contains TM-30 Fidelity Index (Rf) for 99 color samples. Currently it does not have any XML attributes (XML node `CRIs`). As children, Color Rendering Index Collect has a list of *color rendering index groups*.

8.7.2 Color Rendering Index Group

8.7.2.1 General

This section contains Color Rendering Indexes (CRI) for a single color temperature (XML node `CRIGroup`). The currently defined XML attributes of the CRI group are specified in Table 21.

Table 21 — CRI Group Attributes

XML Attribute Name	Value Type	Description
ColorTemperature	*Float*	Color temperature; Default value: 6 000; Unit: Kelvin

8.7.2.2 Color Rendering Index

This section defines the CRI for one of the 99 color samples (XML node `CRI`). The currently defined XML attributes of the Color Rendering Index are specified in Table 22.

Table 22 — CRI Attributes

XML Attribute Name	Value Type	Description
CES	*Enum*	Color sample. The defined values are “CES01”, “CES02”, … “CES99”. Default Value “CES01"
ColorRenderingIndex	*UInt*	The color rendering index for this sample. Size: 1 byte; Default value: 100

The color rendering index does not have any children.

8.8 Connector Collect

8.8.1 General

This section defines the physical connectors. It currently does not have any XML attributes (XML node `Connectors`). As children, the Connector Collect has a list of a *connector*.

8.8.2 Connector

This section defines the connector (XML node `Connector`). The currently defined XML attributes of the connector are specified in Table 23.

Table 23 — Connector Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	Unique Name of the connector.
Type	*Name*	The type of the connector. Find a list of predefined types in *Annex D*.
DMXBreak	*Uint*	Optional: Defines which DMX Break this connector belongs to.
Gender	*Int*	Connectors where the addition of the Gender value equals 0, can be connected; Default value: 0; Male Connectors are −1, Female are +1, Universal are 0.
Length	*Float*	Defines the length of the connector’s wire in meters. "0" means that there is no cable and the connector is built into the housing. Default value "0"

The connector does not have any children.

8.9 Properties Collect

8.9.1 General

This section defines the general properties of the device type (XML node `Properties`). The Properties Collect currently does not have any XML attributes. The currently defined children nodes of properties collect are specified in Table 24.

Table 24 — Properties Collect

XML node	Amount	Description
OperatingTemperature	0 or 1	Temperature range in which the device can be operated.
Weight	0 or 1	Weight of the device including all accessories.
PowerConsumption	Any	Power information for a given connector.
LegHeight	0 or 1	Height of the legs.

8.9.2 OperatingTemperature

This section defines the ambient operating temperature range (XML node `OperatingTemperature`). The currently defined XML attributes of the OperatingTemperature are specified in Table 25.

Table 25 — Operating Temperature Attributes

XML Attribute Name	Value Type	Description
Low	*Float*	Lowest temperature the device can be operated. Unit: °C. Default value: 0
High	*Float*	Highest temperature the device can be operated. Unit: °C. Default value: 40

The OperatingTemperature currently does not have any children.

8.9.3 Weight

This section defines the overall weight of the device (XML node `Weight`). The currently defined XML attributes of the weight are specified in Table 26.

Table 26 — Weight Attributes

XML Attribute Name	Value Type	Description
Value	*Float*	Weight of the device including all accessories. Unit: kilogram. Default value: 0

The weight currently does not have any children.

8.9.4 PowerConsumption

This section defines the maximum power consumption per connector (XML node `PowerConsumption`). The currently defined XML attributes of the PowerConsumption are specified in Table 27.

Table 27 — Power Consumption Attributes

XML Attribute Name	Value Type	Description
Value	*Float*	Defines the power consumption of the connector at full load. Unit: VA. Default value: 0
PowerFactor	*Float*	Defines the cosine of phase of voltage relative to current. Unit: None. Default value: 1
Connector	*Node*	Name of the linked Connector
VoltageLow	*Float*	Defines the lowest possible operating voltage. Unit: Volt. Default value: 90
VoltageHigh	*Float*	Defines the highest possible operating voltage. Unit: Volt. Default value: 240
FrequencyLow	*Float*	Defines the ,owest possible operating frequency. Unit: Hertz. Default value: 50
FrequencyHigh	*Float*	Defines the highest possible operating frequency. Unit: Hertz. Default value: 60

The PowerConsumption currently does not have any children.

8.9.5 LegHeight

This section defines the height of the legs (XML node `LegHeight`). The currently defined XML attributes of the LegHeight are specified in Table 28.

Table 28 — Leg Height Attributes

XML Attribute Name	Value Type	Description
Value	*Float*	Defines height of the legs – distance between the floor and the bottom base plate. Unit: meter. Default value: 0

The LegHeight currently does not have any children.

9 Model Collect

9.1 General

Each device is divided into smaller parts: body, yoke, head and so on. These are called geometries. Each geometry has a separate model description and a physical description. Model collect contains model descriptions of the fixture parts. The model collect currently does not have any XML attributes (XML node `Models`). As children, Model Collect has a list of *model*.

9.2 Model

This section defines the type and dimensions of the model (XML node `Model`). The currently defined XML attributes of the model are specified in Table 29.

Table 29 — Model Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the model
Length	*Float*	Unit: meter; Default value: 0
Width	*Float*	Unit: meter; Default value: 0
Height	*Float*	Unit: meter; Default value: 0
PrimitiveType	*Enum*	Type of 3D model; The currently defined values are: "Undefined", "Cube", "Cylinder", "Sphere", "Base", "Yoke", "Head", "Scanner", "Conventional", "Pigtail", "Base1_1", "Scanner1_1", "Conventional1_1"; Default value: "Undefined"
File	*Resource*	Optional; File name without extension and without subfolder containing description of the model. Use the following as a resource file: — 3DS file to provide 3D model. — STEP file to provide 3D model as a parametric model; — SVG file to provide the 2D symbol. It is possible to add several files with the same name but different formats. Preferable format for the 3D model is 3ds. The resource files are located in subfolders of a folder called `./models`. The names of the subfolders correspond to the file format of the resource files (3ds, step, svg). The path for 3ds files would be `./models/3ds`.

The model currently does not have any children.

All models of a device combined should not exceed a maximum vertices count of 1 200.

The device shall be drawn in a hanging position displaying the front view; the pan axis is *Z* aligned and the tilt axis is X aligned.

The metric system consists of the Right-handed Cartesian Coordinates *XYZ*: X – from left (−*X*) to right (+*X*), *Y* – from the outside of the monitor (−*Y*) to the inside of the monitor (+*Y*), *Z* – from bottom (−*Z*) to top (+*Z*). 0,0,0 – center base plate (see Figure 2).

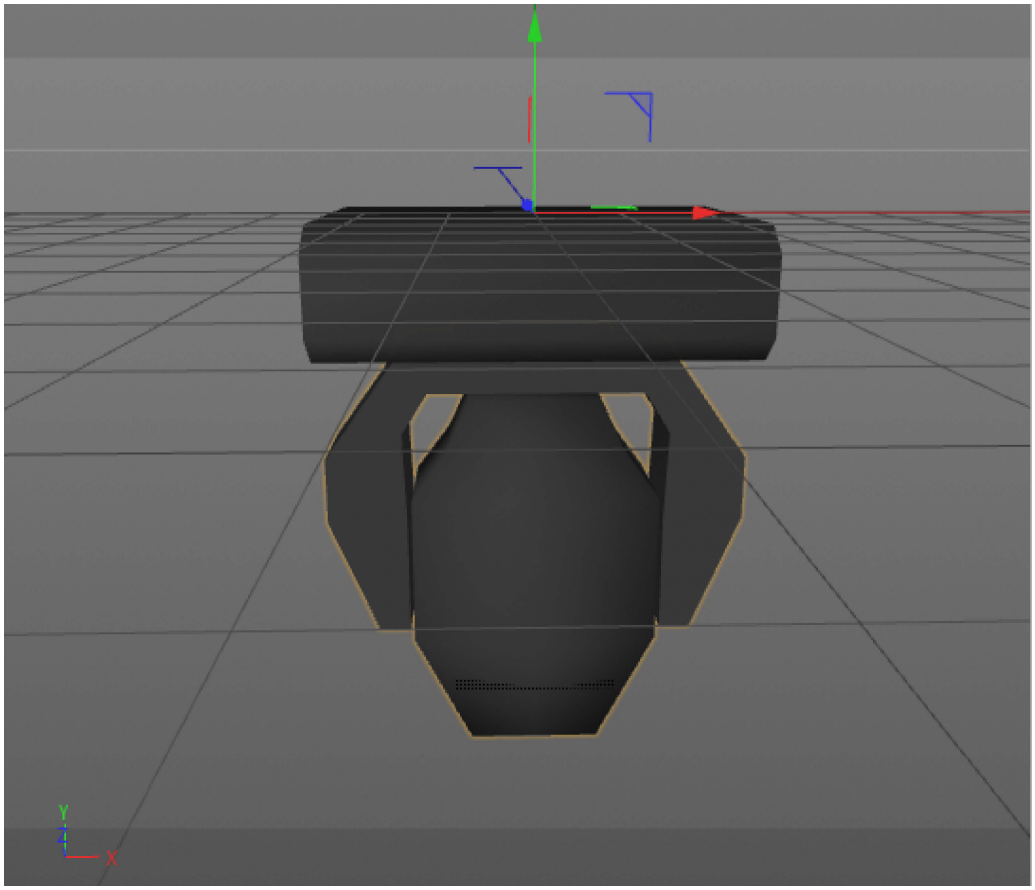


Figure 2 — Device in a hanging position – front view

The mesh of each fixture part shall be drawn around its own suspension point. The zero point of a device does not necessarily have to contain the offset related to the yoke, but it must be centered on its axis of rotation. The offset is defined by the geometry and has to be related to its parent geometry and not to the base.

NOTE In general, the offsets are mostly negative, because the device is displayed in a hanging position.

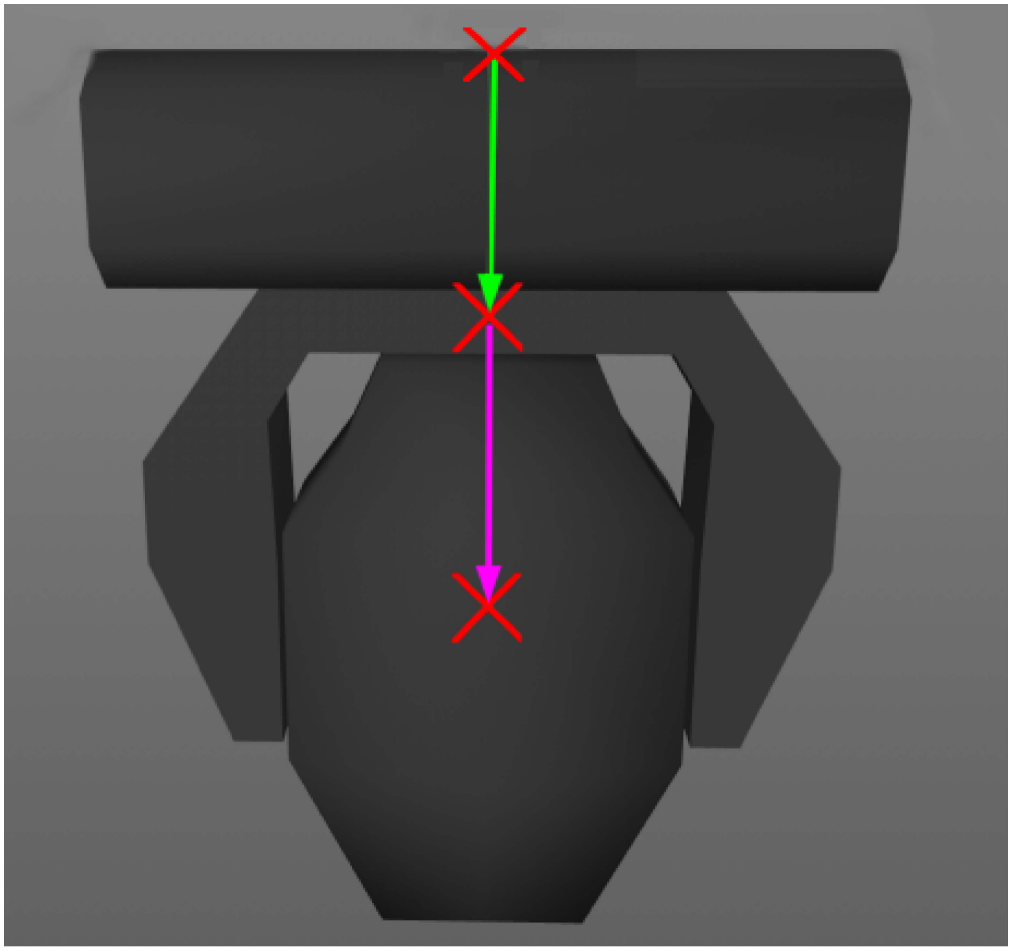


Figure 3 — Offsets of the parts

In Figure 3 the green arrow displays the offset of the yoke related to the base. The magenta arrow displays the offset of the head related to the yoke. The offsets are to be defined by the position matrix of the according geometry (Table 31 to Table 42). It is important that the axis of rotation of each device part is exactly positioned (see Figure 4).

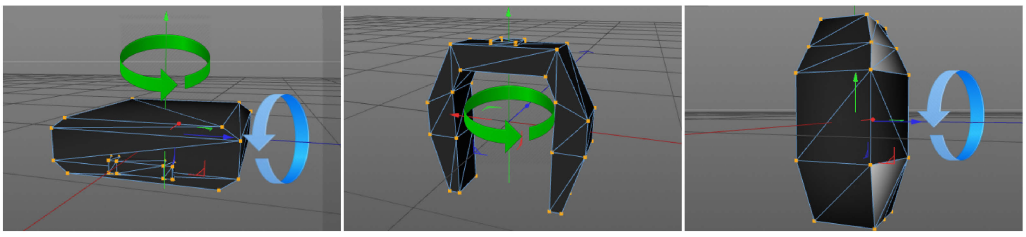


Figure 4 — Positions of rotation axis

The dimension XML attributes of model (see Table 29) are always used, no matter the scaling and ratio of the 3ds file. The mesh is explicitly scaled to this dimension. The length defines the dimension of the model on the *X* axis, the width on the *Y* axis and the height on the *Z* axis.

10 Geometry Collect

10.1 General

The physical description of the device parts is defined in the geometry collect. Geometry collect can contain a separate geometry or a tree of geometries. The geometry collect currently does not have any XML attributes (XML node `Geometries`). The currently defined children nodes of geometry collect are specified in Table 30.

Table 30 — Geometry Children Types

XML node	Amount	Description
Geometry	Any	General Geometry.
Axis	Any	Geometry with axis.
FilterBeam	Any	Geometry with a beam filter.
FilterColor	Any	Geometry with color filter.
FilterGobo	Any	Geometry with gobo.
FilterShaper	Any	Geometry with shaper.
Beam	Any	Geometry that describes a light output to project.
MediaServerLayer	Any	Geometry that describes a media representation layer of a media device.
MediaServerCamera	Any	Geometry that describes a camera or output layer of a media device.
MediaServerMaster	Any	Geometry that describes a master control layer of a media device.
Display	Any	Geometry that describes a surface to display visual media.
GeometryReference	Any	Reference to already described geometries.

NOTE Position the geometry in its "Default" position. This is defined by the Default Value from the DMX Channel that controls the position of that geometry.

10.2 General Geometry

It is a basic geometry type without specification (XML node `Geometry`). The currently defined XML attributes of the geometry are specified in Table 31.

Table 31 — Geometry Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of geometry. Recommendation for conventional is “Body”. Recommendation for a geometry that is representing the base housing of a moving head is “Base”.
Model	*Name*	Link to the corresponding model.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The geometry has the same children types as the geometry collect (see Table 30).

10.3 Geometry Type Axis

This type of geometry defines device parts using a rotation axis (XML node `Axis`). The currently defined XML attributes of the axis are specified in Table 32.

Table 32 — Axis Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry. Recommendation for an axis-geometry is “Yoke”. Recommendation for an axis-geometry representing the lamp housing of a moving head is “Head”. NOTE The Head of a moving head is usually mounted to the Yoke.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The axis has the same children types as the geometry collect (see Table 30).

10.4 Geometry Type Beam Filter

This type of geometry defines device parts with a beam filter (XML node `FilterBeam`). The currently defined XML attributes of the beam filter are specified in Table 33.

Table 33 — Beam Filter Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry. Recommendation for beam filter limiting the diffusion of light is “BarnDoor”. Recommendation for beam filter adjusting the diameter of the beam is “Iris”. NOTE BarnDoor and Iris are usually mounted to conventional.
Model	*Name*	Link to the corresponding model.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The beam has the same children types as the geometry collect (see Table 30).

10.5 Geometry Type Color Filter

This type of geometry is used to describe device parts which have a color filter (XML node `FilterColor`). The currently defined XML attributes of the color filter are specified in Table 34.

Table 34 — Color Filter Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of geometry. Recommendation for filter of a color or mechanical color changer is “FilterColor”. NOTE FilterColor is usually mounted to conventional.
Model	*Name*	Link to the corresponding model.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The color has the same children types as the geometry collect (see Table 30).

10.6 Geometry Type Gobo Filter

This type of geometry is used to describe device parts which have gobo wheels (XML node `FilterGobo`). The currently defined XML attributes of the gobo filter are specified in Table 35.

Table 35 — Gobo Filter Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry. Recommendation for filter of a gobo or mechanical gobo changer is “FilterGobo”. NOTE FilterGobo is usually mounted to conventional.
Model	*Name*	Link to the corresponding model.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The color has the same children types as the geometry collect (see Table 30).

10.7 Geometry Type Shaper Filter

This type of geometry is used to describe device parts which have a shaper (XML node `FilterShaper`). The currently defined XML attributes of the shaper filter are specified in Table 36.

Table 36 — Shaper Filter Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry; Recommendation for filter used to form the beam to a framed, triangular, or trapezoid shape, is “Shaper”. NOTE Shaper is usually mounted to conventional.
Model	*Name*	Link to the corresponding model.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The color has the same children types as the geometry collect (see Table 30).

10.8 Geometry Type Beam

This type of geometry is used to describe device parts which have a light source (XML node `Beam`). The currently defined XML attributes of the Beam are specified in Table 37.

Table 37 — Beam Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry. Recommendation for a light source of a conventional or moving head or a projector is “Beam”. NOTE 1 Beam is usually mounted to Head or Body. Recommendation for a self-emitting single light source is “Pixel”. NOTE 2 Pixel is usually mounted to Head or Body. Recommendation for a number of Pixel that are controlled at the same time is “Array”. NOTE 3 Array is usually mounted to Head or Body. Recommendation for a light source of a moving mirror is “Mirror”. NOTE 4 Mirror is usually mounted to Yoke.
Model	*Name*	Link to the corresponding model.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix
LampType	*Enum*	Defines type of the light source; The currently defined types are: Discharge, Tungsten, Halogen, LED; Default value “Discharge”
PowerConsumption	*Float*	Power consumption; Default value: 1 000; Unit: Watt
LuminousFlux	*Float*	Intensity of all the represented light emitters; Default value: 10 000; Unit: lumen
ColorTemperature	*Float*	Color temperature; Default value: 6 000; Unit: kelvin
BeamAngle	*Float*	Beam angle; Default value: 25,0; Unit: degree
FieldAngle	*Float*	Field angle; Default value: 25,0; Unit: degree
ThrowRatio	*Float*	Throw Ratio of the lens for BeamType Rectangle; Default value: 1; Unit: None
RectangleRatio	*Float*	Ratio from Width to Height of the Rectangle Type Beam; Default value: 1.7777; Unit: None
BeamRadius	*Float*	Beam radius on starting point. Default value: 0,05; Unit: meter.
BeamType	*Enum*	Beam Type; Specified values: “Wash”, “Spot”, “None”, “Rectangle“. Default value “Wash”
ColorRenderingIndex	*Uint*	The CRI according to ANSI/IES TM-30 is a quantitative measure of the ability of the light source showing the object color naturally as it does as daylight reference. Size 1 byte. Default value 100.

The beam has the same children types as the geometry collect (see *Table 30*).

Use the Geometry Type "Beam" to describe the position of the fixture’s light output (usually the position of the lens) and not the position of the light source inside the device. The origin of the Geometry Type "Beam" is the origin of the rendered beam. The origin of the Geometry Type "Beam" should not be covered by any faces of other geometries in order to not block the rendered beam.

The `BeamType` describes how the Beam will be rendered.

— “Wash” – A conical beam with soft edges.

— “Spot” – A conical beam with hard edges.

— “Rectangle” – A pyramid-shaped beam with hard edges.

— “None” – No beam will be drawn, only the geometry itself will emit light.

The beam geometry emits its light into negative *Z* direction (and *Y*-up).

10.9 Geometry Type Media Server Layer

This type of geometry is used to describe the layer of a media device that is used for representation of media files (XML node `MediaServerLayer`). The currently defined XML attributes of the media server layer are specified in Table 38.

Table 38 — Media Server Layer Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry.
Model	*Name*	Link to the corresponding model that will be used to display the alignment in media server space.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The media server layer has the same children types as the geometry collect (see *Table 30*).

10.10 Geometry Type Media Server Camera

This type of geometry is used to describe the camera or output of a media device (XML node `MediaServerCamera`). The currently defined XML attributes of the media server camera are specified in Table 39.

Table 39 — Media Server Camera Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry.
Model	*Name*	Link to the corresponding model that will be used to display the alignment in media server space.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The media server camera has the same children types as the geometry collect (see Table 30).

The media server camera-view faces in the positive *Y*-direction (and *Z*-up).

10.11 Geometry Type Media Server Master

This type of geometry is used to describe the master control of one or several media devices (XML node `MediaServerMaster`). The currently defined XML attributes of the media server master are specified in Table 40.

Table 40 — Media Server Master Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry.
Model	*Name*	Link to the corresponding model.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix

The media server master has the same children types as the geometry collect (see Table 30).

10.12 Geometry Type Display

This type of geometry is used to describe a self-emitting surface which is used to display visual media (XML node `Display`). The currently defined XML attributes of the display are specified in Table 41.

Table 41 — Display Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the geometry.
Model	*Name*	Link to the corresponding model.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix
Texture	*Resource*	Name of the mapped texture in Model file that will be swapped out for the media resource.

The display has the same children types as the geometry collect (see Table 30).

10.13 Geometry Type Reference

10.13.1 General

The Geometry Type Reference is used to describe multiple instances of the same geometry. Example: LED panel with multiple pixels. (XML node `GeometryReference`). The currently defined XML attributes of reference are specified in Table 42.

NOTE Geometry Reference also allows easier definition of the DMX Channels for these geometries.

Table 42 — Geometry Reference Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of geometry.
Position	*Matrix*	Relative position of geometry; Default value: Identity Matrix
Geometry	*Name*	Name of the referenced geometry. Only top level geometries are allowed to be referenced.
Model	*Name*	Optional; Link to the corresponding model. The model only replaces the model of the parent of the referenced geometry. The models of the children of the referenced geometry are not affected. The starting point is Models Collect. If model is not set, the model is taken from the referenced geometry.

As children, the Geometry Type Reference has a list of a *break*. The count of the children depends on the number of different breaks in the DMX channels of the referenced geometry. If the referenced geometry, for example, has DMX channels with DMX break 2 and 4, the geometry reference has to have 2 children. The first child with DMX offset for DMX break 2 and the second child for DMX break 4. If one or more of the DMX channels of the referenced geometry have the special value “Overwrite” as a DMX break, the DMX break for those channels and the DMX offsets need to be defined.

10.13.2 Break

This XML node specifies the DMX offset for the DMX channel of the referenced geometry (XML node `Break`). The currently defined XML attributes of the break are specified in Table 43.

Table 43 — Break Attributes

XML Attribute Name	Value Type	Description
DMXOffset	*DMXAddress*	DMX offset; Default value:1 (Means no offset for the corresponding DMX Channel)
DMXBreak	*Uint*	Defines the unique number of the DMX Break for which the Offset is given. Size: 1 byte; Default value 1.

11 DMX Mode Collect

11.1 General

This section is describes all DMX modes of the device. If firmware revisions change a DMX footprint, then such revisions should be specified as new DMX mode. The DMX mode collect currently does not have any attributes (XML node `DMXModes`). As a child the fixture type DMX mode collect has DMX modes.

11.2 DMX Mode

Each DMX mode describes logical control a part of the device in a specific mode (XML node `DMXMode`). The currently defined XML attributes of the DMX mode are specified in Table 44.

Table 44 — DMX Mode Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the DMX mode
Geometry	*Name*	Name of the first geometry in the device; Only top level geometries are allowed to be linked.

DMX mode children are specified in Table 45.

Table 45 — "DMX Mode Children"

XML node	Mandatory	Description
DMXChannels	Yes	Description of all DMX channels used in the mode
Relations	No	Description of relations between channels
FTMacros	No	Is used to describe macros of the manufacturer.

11.2.1 DMX Channel Collect

11.2.1.1 General

This section defines the DMX footprint of the device. The DMX channel collect currently does not have any attributes (XML node `DMXChannels`). As children, the DMX Channel Collect has a list of a *DMX channel*.

11.2.1.2 DMX Channel

11.2.1.2.1 General

This section defines the DMX channel (XML node `DMXChannel`). The name of a DMX channel cannot be user-defined and must consist of a geometry name and the attribute name of the first logical channel with the separator "_". Currently defined XML attributes of the DMX channel are specified in Table 46.

Table 46 — DMX Channel Attributes

XML Attribute Name	Value Type	Description
DMXBreak	*Int*	Number of the DMXBreak; Default value: 1; Special value: “Overwrite” – means that this number will be overwritten by Geometry Reference; Size: 4 bytes
Offset	*Array of Int*	Relative addresses of the current DMX channel from highest to least significant; Seperator of values is ","; Special value: “None” – does not have any addresses; Default value: “None”; Size per int: 4 bytes
InitialFunction	*Node*	Link to the channel function that will be activated by default for this DMXChannel;
Highlight	*DMXValue*	Highlight value for current channel; Special value: “None”. Default value: “None”.
Geometry	*Name*	Name of the geometry the current channel controls.

As children, the DMX Channel has a list of *logical channel*.

11.2.1.2.2 Logical Channel

The Fixture Type Attribute is assinged to a LogicalChannel and defines the function of the LogicalChannel. All logical channels that are children of the same DMX channel are mutually exclusive. In a DMX mode, only one logical channel with the same attribute can reference the same geometry at a time. The name of a Logical Channel cannot be user-defined and is equal to the linked attribute name. The XML node of the logical channel is `LogicalChannel`. The currently defined XML attributes of the logical channel are specified in Table 47.

Table 47 — Logical Channel Attributes

XML Attribute Name	Value Type	Description
Attribute	*Node*	Link to the attribute; The starting point is the Attribute Collect (see *Annex A*).
Snap	*Enum*	If snap is enabled, the logical channel will not fade between values. Instead, it will jump directly to the new value.; Value: “Yes”, “No”, “On”, “Off”. Default value: “No”
Master	*Enum*	Defines if all the subordinate channel functions react to a Group Control defined by the control system. Values: “None”, “Grand”, “Group”; Default value: “None”.
MibFade	*Float*	Minimum fade time for moves in black action. MibFade is defined for the complete DMX range. Default value: 0; Unit: second
DMXChangeTimeLimit	*Float*	Minimum fade time for the subordinate channel functions to change DMX values by the control system. DMXChangeTimeLimit is defined for the complete DMX range. Default value: 0; Unit: second

As a child the logical channel has a list of a *channel function*.

11.2.1.2.3 Channel Function

The Fixture Type Attribute is assinged to a Channel Function and defines the function of its DMX Range. (XML node `ChannelFunction`). The currently defined XML attributes of channel function are specified in Table 48.

Table 48 — Channel Function Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	Unique name; Default value: Name of attribute and number of channel function.
Attribute	*Node*	Link to attribute; Starting point is the attributes node. Default value: “NoFeature”.
OriginalAttribute	*String*	The manufacturer’s original name of the attribute; Default: empty
DMXFrom	*DMXValue*	Start DMX value; The end DMX value is calculated as a DMXFrom of the next channel function – 1 or the maximum value of the DMX channel. Default value: "0/1".
Default	*DMXValue*	Default DMX value of channel function when activated by the control system.
PhysicalFrom	*Float*	Physical start value; Default value: 0
PhysicalTo	*Float*	Physical end value; Default value: 1
RealFade	*Float*	Time in seconds to move from min to max of the Channel Function; Default value: 0
RealAcceleration	*Float*	Time in seconds to accelerate from stop to maximum velocity; Default value: 0
Wheel	*Node*	Optional link to wheel; Starting point: Wheel Collect

XML Attribute Name	Value Type	Description
Emitter	*Node*	Optional link to emitter in the physical description; Starting point: Emitter Collect
Filter	*Node*	Optional link to filter in the physical description; Starting point: Filter Collect
ModeMaster	*Node*	Link to DMX Channel or Channel Function; Starting point DMX mode
ModeFrom	*DMXValue*	Only used together with ModeMaster; DMX start value; Default value: 0/1
ModeTo	*DMXValue*	Only used together with ModeMaster; DMX end value; Default value: 0/1

As children, the Channel Function has list of a *channel set*.

11.2.1.2.4 Channel Set

This section defines the channel sets of the channel function (XML node `ChannelSet`). The currently defined XML attributes of the channel set are specified in Table 49.

Table 49 — Channel Set Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The name of the channel set. Default: Empty
DMXFrom	*DMXValue*	Start DMX value; The end DMX value is calculated as a DMXFrom of the next channel set – 1 or the maximum value of the current channel function; Default value: 0/1
PhysicalFrom	*Float*	Physical start value
PhysicalTo	*Float*	Physical end value
WheelSlotIndex	*Int*	If the channel function has a link to a wheel, a corresponding slot index needs to be specified. The wheel slot index results from the order of slots of the wheel which is linked in the channel function. The wheel slot index is normalized to 1. Size: 4 bytes

The channel set does not have any children.

11.2.2 Relation Collect

11.2.2.1 General

This section describes the dependencies between DMX channels and channel functions, such as multiply and override. The relation collect currently does not have any XML attributes (XML node `Relations`). As children, the Relation Collect has a list of a *relation*.

11.2.2.2 Relation

This section defines the relation between the master DMX channel and the following logical channel (XML node `Relation`). The currently defined XML attributes of the relations are specified in Table 50.

Table 50 — Relation Attributes

XML Attribute Name	Value Type	Description
Name	*String*	The unique name of the relation
Master	*Node*	Link to the master DMX channel; Starting point: DMX mode
Follower	*Node*	Link to the following channel function; Starting point: DMX mode
Type	*Enum*	Type of the relation; Values: “Multiply”, “Override”

The relation does not have any children.

11.2.3 Macro Collect

11.2.3.1 General

This section describes DMX sequences to be executed by the control system. The macro collect currently does not have any XML attributes (XML node `FTMacros`). As children, the Macro Collect has a list of *macro*.

11.2.3.2 Macro

11.2.3.2.1 General

This section defines a DMX sequence. (XML node `FTMacro`). The currently defined XML attributes of the macro are specified in Table 51.

Table 51 — Macro Attributes

XML Attribute Name	Value Type	Description
Name	*Name*	The unique name of the macro.

Macro children are specified in Table 52.

Table 52 — Macro Children

XML node	Mandatory	Description
MacroDMX	No	Defines a DMX sequence.

11.2.3.2.2 Macro DMX

This section defines the sequence of DMX values which are sent by a control system. (XML node `MacroDMX`). As children, the Macro DMX has a list of a *MacroDMXStep*.

11.2.3.2.3 Macro DMX Step

This section defines a DMX step (XML node `MacroDMXStep`). The currently defined XML attributes of the macro DMX step are specified in Table 53.

Table 53 — Macro DMX Step Attributes

XML Attribute Name	Value Type	Description
Duration	*Float*	Duration of a step; Default value: 1; Unit: seconds.

As children, the Macro DMX-Step has a list of a *DMX Value*.

11.2.3.2.4 DMX Value

This section defines the value for DMX channel (XML node `MacroDMXValue`). The currently defined XML attributes of the DMX Value are specified in Table 54.

Table 54 — DMX Value Attributes

XML Attribute Name	Value Type	Description
Value	*DMXValue*	Value of the DMX channel
DMXChannel	*Node*	Link to a DMX channel. Starting node *DMX Channel Collect*.

The DMX value does not have any children.

12 Revision Collect

12.1 General

This section defines the history of device type. Revision collect currently does not have any XML attributes (XML node `Revisions`). As children, the Revision Collect has a list of *revision*.

12.2 Revision

This section defines one revision of the device type (XML node `Revision`). Revisions are optional. Every time a GDTF file is uploaded to the database, a revision with the actual time and UserID is created by the database. The currently defined XML attributes of the revision are specified in Table 55.

Table 55 — Revision Attributes

XML Attribute Name	Value Type	Description
Text	*String*	User-defined text for this revision; Default value: empty
Date	*Date*	Revision date and time
UserID	*Uint*	UserID of the user that has uploaded the GDTF file to the database; Default value: 0

The revision does not have any children.

13 Fixture Type Preset Collect

13.1 General

This section defines fixture type specific presets. It currently does not have any XML attributes (XML node `FTPresets`). As children, the Fixture Type Preset Collect has a list of *FTPreset* (fixture type preset).

13.2 Fixture Type Preset

This section has not yet been defined (XML node `FTPreset`).

14 Supported Protocol Collect

14.1 General

If the device supports one or several additional protocols, these protocols specific information have to be specified. The supported protocol collect currently does not have any XML attributes (XML node `Protocols`). Children of supported protocol collect are specified in Table 56.

Table 56 — Supported Protocol Collect Children

XML node	Mandatory	Description
RDM	No	Describes RDM information
Art-Net	No	Describes Art-Net information
sACN	No	Describes sACN information
PosiStageNet	No	Describes PosiStageNet information
OpenSoundControl	No	Describes OpenSoundControl information
CITP	No	Describes CITP information

14.2 RDM Section

If the device supports the RDM protocol, this section defines the corresponding information (XML node `FTRDM`). The currently defined XML attributes of RDM are specified in Table 57.

Table 57 — RDM Attributes

XML Attribute Name	Value Type	Description
ManufacturerID	*Hex*	Manufacturer ESTA ID
DeviceModelID	*Hex*	Unique device model ID

As children, the FTRDM has a list of `SoftwareVersionID`.

14.2.1 SoftwareVersionID

14.2.1.1 General

For each supported software version add an XML node `SoftwareVersionID`. The currently defined XML attributes are specified in Table 58.

Table 58 — SoftwareVersionID

XML Attribute Name	Value Type	Description
Value	*Hex*	Software version ID

As children, the SoftwareVersionID has a list of `DMXPersonality`.

14.2.1.2 DMXPersonality

To define the supported software versions add an XML node `DMXPersonality`. The currently defined XML attributes are specified in Table 59.

Table 59 — DMXPersonality

XML Attribute Name	Value Type	Description
Value	*Hex*	Hex Value of the DMXPersonality
DMXMode	*Name*	Link to the DMX Mode that can be used with this software version.

The DMXPersonality does not have any children.

14.3 Art-Net Section

This section has not yet been defined (XML node `Art-Net`).

14.4 Streaming ACN Section

This section has not yet been defined (XML node `sACN`).

14.5 Posi Stage Net Section

This section has not yet been defined (XML node `PosiStageNet`).

14.6 Open Sound Control Section

This section has not yet been defined (XML node `OpenSoundControl`).

14.7 CITP Section

This section has not yet been defined (XML node `CITP`).

Annex A
(normative)

Attribute Definitions

This Annex defines the default set of Fixture Type Attributes that are used to describe the functionality of the device. (n) and (m) are wildcards for the enumeration of attributes like Gobo(n) – Gobo1 and Gobo2 or VideoEffect(n)Parameter(m) – VideoEffect1Parameter1 and VideoEffect1Parameter2. Fixture Type Attributes without wildcards (n) or (m) are not enumerated. The enumeration starts with 1. The currently defined Fixture Type Attributes are specified in Table A.1.

NOTE The predefined Fixture Type Attributes are the preferred to use Fixture Type Attribute. At any time user defined attributes can be introduced as well.

Table A.1 — Structure of Attribute

Description	Attribute
Dimmer	Controls the intensity of a fixture.
Pan	Controls the fixture's sideward movement (horizontal axis).
Tilt	Controls the fixture's upward and the downward movement (vertical axis).
PanRotate	Controls the speed of the fixture's continuous pan movement (horizontal axis).
TiltRotate	Controls the speed of the fixture's continuous tilt movement (vertical axis).
PositionEffect	Selects the predefined position effects that are built into the fixture.
PositionEffectRate	Controls the speed of the predefined position effects that are built into the fixture.
PositionEffectFade	Snaps or smooth fades with timing in running predefined position effects.
XYZ_X	Defines a fixture's x-coordinate within an *XYZ* coordinate system.
XYZ_Y	Defines a fixture's y-coordinate within an *XYZ* coordinate system.
XYZ_Z	Defines a fixture's z-coordinate within an *XYZ* coordinate system.
Rot_X	Defines rotation around *X* axis.
Rot_Y	Defines rotation around *Y* axis.
Rot_Z	Defines rotation around *Z* axis.
Scale_X	Scaling on *X* axis.
Scale_Y	Scaling on *Y* axis.
Scale_Z	Scaling on *Y* axis.
Scale_XYZ	Unified scaling on all axes.

Description	Attribute
Gobo(n)	The fixture's gobo wheel (n). This is the main attribute of gobo wheel's (n) wheel control. Selects gobos in gobo wheel (n). A different channel function sets the angle of the indexed position in the selected gobo or the angular speed of its continuous rotation.
Gobo(n)SelectSpin	Selects gobos whose rotation is continuous in gobo wheel (n) and controls the angular speed of the gobo's spin within the same channel function.
Gobo(n)SelectShake	Selects gobos which shake in gobo wheel (n) and controls the frequency of the gobo's shake within the same channel function.
Gobo(n)SelectEffects	Selects gobos which run effects in gobo wheel (n).
Gobo(n)WheelIndex	Controls angle of indexed rotation of gobo wheel (n).
Gobo(n)WheelSpin	Controls the speed and direction of continuous rotation of gobo wheel (n).
Gobo(n)WheelShake	Controls frequency of the shake of gobo wheel (n).
Gobo(n)WheelRandom	Controls speed of gobo wheel's (n) random gobo slot selection.
Gobo(n)WheelAudio	Controls audio-controlled functionality of gobo wheel (n).
Gobo(n)Pos	Controls angle of indexed rotation of gobos in gobo wheel (n). This is the main attribute of gobo wheel's (n) wheel slot control.
Gobo(n)PosRotate	Controls the speed and direction of continuous rotation of gobos in gobo wheel (n).
Gobo(n)PosShake	Controls frequency of the shake of gobos in gobo wheel (n).
AnimationWheel(n)	This is the main attribute of the animation wheel's (n) wheel control. Selects slots in the animation wheel. A different channel function sets the angle of the indexed position in the selected slot or the angular speed of its continuous rotation. Is used for animation effects with multiple slots.
AnimationWheel(n)Audio	Controls audio-controlled functionality of animation wheel (n).
AnimationWheel(n)Macro	Selects predefined effects in animation wheel (n).
AnimationWheel(n)Random	Controls frequency of animation wheel (n) random slot selection.
AnimationWheel(n)SelectEffects	Selects slots which run effects in animation wheel (n).
AnimationWheel(n)SelectShake	Selects slots which shake in animation wheel and controls the frequency of the slots shake within the same channel function.
AnimationWheel(n)SelectSpin	Selects slots whose rotation is continuous in animation wheel and controls the angular speed of the slot spin within the same channel function
AnimationWheel(n)Pos	Controls angle of indexed rotation of slots in animation wheel. This is the main attribute of animation wheel (n) wheel slot control.
AnimationWheel(n)PosRotate	Controls the speed and direction of continuous rotation of slots in animation wheel (n).
AnimationWheel(n)PosShake	Controls frequency of the shake of slots in animation wheel (n).

Description	Attribute
AnimationSystem(n)	This is the main attribute of the animation system insertion control. Controls the insertion of the fixture's animation system in the light output. Is used for animation effects where a disk is inserted into the light output.
AnimationSystem(n)Ramp	Sets frequency of animation system (n) insertion ramp.
AnimationSystem(n)Shake	Sets frequency of animation system (n) insertion shake.
AnimationSystem(n)Audio	Controls audio-controlled functionality of animation system (n) insertion.
AnimationSystem(n)Random	Controls frequency of animation system (n) random insertion.
AnimationSystem(n)Pos	This is the main attribute of the animation system spinning control. Controls angle of indexed rotation of animation system (n) disk.
AnimationSystem(n)PosRotate	Controls the speed and direction of continuous rotation of animation system (n) disk.
AnimationSystem(n)PosShake	Controls frequency of the shake of animation system (n) disk.
AnimationSystem(n)PosRandom	Controls random speed of animation system (n) disk.
AnimationSystem(n)PosAudio	Controls audio-controlled functionality of animation system (n) disk.
AnimationSystem(n)Macro	Selects predefined effects in animation system (n).
MediaFolder(n)	Selects folder that contains media content.
MediaContent(n)	Selects file with media content.
ModelFolder(n)	Selects folder that contains 3D model content. For example 3D meshes for mapping.
ModelContent(n)	Selects file with 3D model content.
PlayMode	Defines media playback mode.
PlayBegin	Defines starting point of media content playback.
PlayEnd	Defines end point of media content playback.
PlaySpeed	Adjusts playback speed of media content.
ColorEffects(n)	Selects predefined color effects built into the fixture.
Color(n)	The fixture’s color wheel (n). Selects colors in color wheel (n). This is the main attribute of color wheel’s (n) wheel control.
Color(n)WheelIndex	Controls angle of indexed rotation of color wheel (n)
Color(n)WheelSpin	Controls the speed and direction of continuous rotation of color wheel (n).
Color(n)WheelRandom	Controls frequency of color wheel’s (n) random color slot selection.
Color(n)WheelAudio	Controls audio-controlled functionality of color wheel (n).
ColorAdd_R	Controls the intensity of the fixture’s red emitters for direct additive color mixing.
ColorAdd_G	Controls the intensity of the fixture’s green emitters for direct additive color mixing

Description	Attribute
ColorAdd_B	Controls the intensity of the fixture's blue emitters for direct additive color mixing.
ColorAdd_C	Controls the intensity of the fixture's cyan emitters for direct additive color mixing.
ColorAdd_M	Controls the intensity of the fixture's magenta emitters for direct additive color mixing.
ColorAdd_Y	Controls the intensity of the fixture's yellow emitters for direct additive color mixing.
ColorAdd_RY	Controls the intensity of the fixture's amber emitters for direct additive color mixing.
ColorAdd_GY	Controls the intensity of the fixture's lime emitters for direct additive color mixing.
ColorAdd_GC	Controls the intensity of the fixture's blue-green emitters for direct additive color mixing.
ColorAdd_BC	Controls the intensity of the fixture's light-blue emitters for direct additive color mixing.
ColorAdd_BM	Controls the intensity of the fixture's purple emitters for direct additive color mixing.
ColorAdd_RM	Controls the intensity of the fixture's pink emitters for direct additive color mixing.
ColorAdd_W	Controls the intensity of the fixture's white emitters for direct additive color mixing.
ColorAdd_WW	Controls the intensity of the fixture's warm white emitters for direct additive color mixing.
ColorAdd_CW	Controls the intensity of the fixture's cool white emitters for direct additive color mixing.
ColorAdd_UV	Controls the intensity of the fixture's UV emitters for direct additive color mixing.
ColorSub_R	Controls the insertion of the fixture's red filter flag for direct subtractive color mixing.
ColorSub_G	Controls the insertion of the fixture's green filter flag for direct subtractive color mixing.
ColorSub_B	Controls the insertion of the fixture's blue filter flag for direct subtractive color mixing.
ColorSub_C	Controls the insertion of the fixture's cyan filter flag for direct subtractive color mixing.
ColorSub_M	Controls the insertion of the fixture's magenta filter flag for direct subtractive color mixing.
ColorSub_Y	Controls the insertion of the fixture's yellow filter flag for direct subtractive color mixing.
ColorMacro(n)	Selects predefined colors that are programed in the fixture's firmware.

Description	Attribute
ColorMacro(n)Rate	Controls the time between Color Macro steps.
CTO	Controls the fixture's "Correct to orange" wheel or mixing system.
CTC	Controls the fixture's "Correct to color" wheel or mixing system.
CTB	Controls the fixture's "Correct to blue" wheel or mixing system.
Tint	Controls the fixture's "Correct green to magenta" wheel or mixing system.
HSB_Hue	Controls the fixture's color attribute regarding the hue.
HSB_Saturation	Controls the fixture's color attribute regarding the saturation.
HSB_Brightness	Controls the fixture's color attribute regarding the brightness.
HSB_Quality	Controls the fixture's color attribute regarding the quality.
CIE_X	Controls the fixture's CIE 1931 color attribute regarding the chromaticity x.
CIE_Y	Controls the fixture's CIE 1931 color attribute regarding the chromaticity y.
CIE_Brightness	Controls the fixture's CIE 1931 color attribute regarding the brightness (Y).
ColorRGB_Red	Controls the fixture's red attribute for indirect RGB color mixing.
ColorRGB_Green	Controls the fixture's green attribute for indirect RGB color mixing.
ColorRGB_Blue	Controls the fixture's blue attribute for indirect RGB color mixing.
ColorRGB_Cyan	Controls the fixture's cyan attribute for indirect CMY color mixing.
ColorRGB_Magenta	Controls the fixture's magenta attribute for indirect CMY color mixing.
ColorRGB_Yellow	Controls the fixture's yellow attribute for indirect CMY color mixing.
ColorRGB_Quality	Controls the fixture's quality attribute for indirect color mixing.
VideoBoost_R	Adjusts color boost red of content.
VideoBoost_G	Adjusts color boost green of content.
VideoBoost_B	Adjusts color boost blue of content.
VideoHueShift	Adjusts color hue shift of content.
VideoSaturation	Adjusts saturation of content.
VideoBrightness	Adjusts brightness of content.
VideoContrast	Adjusts contrast of content.
VideoKeyColor_R	Adjusts red color for color keying.
VideoKeyColor_G	Adjusts green color for color keying.
VideoKeyColor_B	Adjusts blue color for color keying.
VideoKeyIntensity	Adjusts intensity of color keying.
VideoKeyTolerance	Adjusts tolerance of color keying.

Description	Attribute
StrobeDuration	Controls the length of a strobe flash.
StrobeRate	Controls the time between strobe flashes.
Shutter(n)	Controls the fixture's mechanical or electronical shutter feature.
Shutter(n)Strobe	Controls the frequency of the fixture's mechanical or electronical strobe shutter feature.
Shutter(n)StrobePulse	Controls the frequency of the fixture's mechanical or electronical pulse shutter feature.
Shutter(n)StrobePulseClose	Controls the frequency of the fixture's mechanical or electronical closing pulse shutter feature.
Shutter(n)StrobePulseOpen	Controls the frequency of the fixture's mechanical or electronical opening pulse shutter feature.
Shutter(n)StrobeRandom	Controls the frequency of the fixture's mechanical or electronical random strobe shutter feature.
Shutter(n)StrobeRandomPulse	Controls the frequency of the fixture's mechanical or electronical random pulse shutter feature.
Shutter(n)StrobeRandomPulseClose	Controls the frequency of the fixture's mechanical or electronical random closing pulse shutter feature.
Shutter(n)StrobeRandomPulseOpen	Controls the frequency of the fixture's mechanical or electronical random opening pulse shutter feature.
Shutter(n)StrobeEffect	Controls the frequency of the fixture's mechanical or electronical shutter effect feature.
Iris	Controls the diameter of the fixture's beam.
IrisStrobe	Sets frequency of the iris's strobe feature.
IrisStrobeRandom	Sets frequency of the iris's random movement.
IrisPulseClose	Sets frequency of iris's closing pulse.
IrisPulseOpen	Sets frequency of iris's opening pulse.
IrisRandomPulseClose	Sets frequency of iris's random closing pulse.
IrisRandomPulseOpen	Sets frequency of iris's random opening pulse.
Frost(n)	The ability to soften the fixture's spot light with a frosted lens.
Frost(n)PulseOpen	Sets frequency of frost's opening pulse
Frost(n)PulseClose	Sets frequency of frost's closing pulse.
Frost(n)Ramp	Sets frequency of frost's ramp.
Prism(n)	The fixture's prism wheel (n). Selects prisms in prism wheel (n). A different channel function sets the angle of the indexed position in the selected prism or the angular speed of its continuous rotation. This is the main attribute of prism wheel's (n) wheel control.
Prism(n)SelectSpin	Selects prisms whose rotation is continuous in prism wheel (n) and controls the angular speed of the prism's spin within the same channel function.

Description	Attribute
Prism(n)Macro	Macro functions of prism wheel (n).
Prism(n)Pos	Controls angle of indexed rotation of prisms in prism wheel (n). This is the main attribute of prism wheel's 1 wheel slot control.
Prism(n)PosRotate	Controls the speed and direction of continuous rotation of prisms in prism wheel (n).
Effects(n)	Generically predefined macros and effects of a fixture.
Effects(n)Rate	Frequency of running effects.
Effects(n)Fade	Snapping or smooth look of running effects.
Effects(n)Adjust(m)	Controls parameter (m) of effect (n)
Effects(n)Pos	Controls angle of indexed rotation of slot/effect in effect wheel/macro (n). This is the main attribute of effect wheel/macro (n) slot/effect control.
Effects(n)PosRotate	Controls speed and direction of slot/effect in effect wheel (n).
EffectsSync	Sets offset between running effects and effects 2.
BeamShaper	Activates fixture's beam shaper.
BeamShaperMacro	Predefined presets for fixture's beam shaper positions.
BeamShaperPos	Indexing of fixture's beam shaper.
BeamShaperPosRotate	Continuous rotation of fixture's beam shaper.
Zoom	Controls the spread of the fixture's beam/spot.
ZoomModeSpot	Selects spot mode of zoom.
ZoomModeBeam	Selects beam mode of zoom.
Focus(n)	Controls the sharpness of the fixture's spot light. Can blur or sharpen the edge of the spot.
Focus(n)Adjust	Autofocuses functionality using presets.
Focus(n)Distance	Autofocuses functionality using distance.
Control(n)	Controls the channel of a fixture.
DimmerMode	Selects different modes of intensity.
DimmerCurve	Selects different dimmer curves of the fixture.
BlackoutMode	Closes the light output under certain conditions (movement correction, gobo movement, etc.).
LEDFrequency	Controls LED frequency.
LEDZoneMode	Changes zones of LEDs.
PixelMode	Controls behavior of LED pixels.
PanMode	Selects fixture's pan mode. Selects between a limited pan range (e.g. −270 to 270) or a continuous pan range.
TiltMode	Selects fixture's pan mode. Selects between a limited tilt range (e.g. −130 to 130) or a continuous tilt range.

Description	Attribute
PanTiltMode	Selects fixture's pan/tilt mode. Selects between a limited pan/tilt range or a continuous pan/tilt range.
PositionModes	Selects the fixture's position mode.
Gobo(n)WheelMode	Changes control between selecting, indexing, and rotating the gobos of gobo wheel (n).
AnimationWheel(n)Mode	Changes control between selecting, indexing, and rotating the slots of animation wheel (n).
AnimationWheelShortcutMode	Defines whether the animation wheel takes the shortest distance between two positions.
Color(n)Mode	Changes control between selecting, continuous selection, half selection, random selection, color spinning, etc. in colors of color wheel (n).
ColorWheelShortcutMode	Defines whether the color wheel takes the shortest distance between two colors.
CyanMode	Controls how Cyan is used within the fixture's cyan CMY-mixing feature.
MagentaMode	Controls how Cyan is used within the fixture's magenta CMY-mixing.
YellowMode	Controls how Cyan is used within the fixture's yellow CMY-mixing feature.
ColorMixMode	Changes control between selecting continuous selection, half selection, random selection, color spinning, etc. in color mixing.
ChromaticMode	Selects chromatic behavior of the device.
ColorCalibrationMode	Sets calibration mode (for example on/off).
ColorConsistency	Controls consistent behavior of color.
ColorControl	Controls special color related functions.
ColorModelMode	Controls color model (CMY/RGB/HSV).
ColorSettingsReset	Resets settings of color control channel.
ColorUniformity	Controls behavior of color uniformity.
CRIMode	Controls CRI settings of output.
CustomColor	Custom color related functions (save, recall).
UVStability	Settings for UV stability color behavior.
WavelengthCorrection	Settings for WaveLength corrections of colors.
WhiteCount	Controls if White LED is proportionally added to RGB.
StrobeMode	Changes strobe style (strobe, pulse, random strobe, etc.) of the shutter attribute.
ZoomMode	Changes modes of the fixture's zoom.
FocusMode	Changes modes of the fixture's focus – manual or auto-focus.
IrisMode	Changes modes of the fixture's iris – linear, strobe, pulse.

Description	Attribute
Fan(n)Mode	Controls fan (n) mode.
FollowSpotMode	Selects follow spot control mode.
BeamEffectIndexRotateMode	Changes mode to control either index or rotation of the beam effects.
IntensityMSpeed	Movement speed of the fixture's intensity.
PositionMSpeed	Movement speed of the fixture's pan/tilt.
ColorMixMSpeed	Movement speed of the fixture's ColorMix presets.
ColorWheelSelectMSpeed	Movement speed of the fixture's color wheel.
GoboWheel(n)MSpeed	Movement speed of the fixture's gobo wheel (n).
IrisMSpeed	Movement speed of the fixture's iris.
Prism(n)MSpeed	Movement speed of the fixture's prism wheel (n).
FocusMSpeed	Movement speed of the fixture's focus.
Frost(n)MSpeed	Movement speed of the fixture's frost (n).
ZoomMSpeed	Movement speed of the fixture's zoom.
FrameMSpeed	Movement speed of the fixture's shapers.
GlobalMSpeed	General speed of fixture's features.
ReflectorAdjust	Movement speed of the fixture's frost.
FixtureGlobalReset	Generally resets the entire fixture.
ShutterReset	Resets the fixture's shutter.
BeamReset	Resets the fixture's beam features.
ColorMixReset	Resets the fixture's color mixing system.
ColorWheelReset	Resets the fixture's color wheel.
FocusReset	Resets the fixture's focus.
FrameReset	Resets the fixture's shapers.
GoboWheelReset	Resets the fixture's gobo wheel.
IntensityReset	Resets the fixture's intensity.
IrisReset	Resets the fixture's iris.
PositionReset	Resets the fixture's pan/tilt.
PanReset	Resets the fixture's pan.
TiltReset	Resets the fixture's tilt.
ZoomReset	Resets the fixture's zoom.
CTBReset	Resets the fixture's CTB.
CTOReset	Resets the fixture's CTO.
CTCReset	Resets the fixture's CTC.
AnimationSystemReset	Resets the fixture's animation system features.

Description	Attribute
FixtureCalibrationReset	Resets the fixture's calibration.
Function	Generally controls features of the fixture.
LampControl	Controls the fixture's lamp on/lamp off feature.
DisplayIntensity	Adjusts intensity of display
DMXInput	Selects DMX Input
NoFeature	Ranges without a functionality.
Blower(n)	Fog or hazer‘s blower feature.
Fan(n)	Fog or hazer's Fan feature.
Fog(n)	Fog or hazer's Fog feature.
Haze(n)	Fog or hazer's haze feature.
LampPowerMode	Controls the energy consumption of the lamp.
Fans	Fancontrols a fixture or device.
Blade(n)A	1 of 2 shutters that shape the top/right/bottom/left of the beam.
Blade(n)B	2 of 2 shutters that shape the top/right/bottom/left of the beam.
Blade(n)Rot	Rotates position of blade(n).
ShaperRot	Rotates position of blade assembly.
ShaperMacros	Predefined presets for shaper positions.
ShaperMacrosSpeed	Speed of predefined effects on shapers.
BladeSoft(n)A	1 of 2 soft edge blades that shape the top/right/bottom/left of the beam.
BladeSoft(n)B	2 of 2 soft edge blades that shape the top/right/bottom/left of the beam.
KeyStone(n)A	1 of 2 corners that shape the top/right/bottom/left of the beam.
KeyStone(n)B	2 of 2 corners that shape the top/right/bottom/left of the beam.
Video	Controls video features.
VideoEffect(n)Type	Selects dedicated effects which are used for media.
VideoEffect(n)Parameter(m)	Controls parameter (m) of VideoEffect(n)Type.
VideoCamera(n)	Selects the video camera(n).
VideoSoundVolume(n)	Adjusts sound volume.
VideoBlendMode	Defines mode of video blending.
InputSource	Defines media input source e.g. a camera input.
FieldOfView	Defines field of view.

Annex B
(normative)

Attribute Listing

This Annex defines the default Fixture Type Attribute Listing and the defined properties. (n) and (m) are wildcards for enumeration of attributes e.g., Gobo(n) – Gobo1 and Gobo2 or VideoEffect(n)Parameter(m) – VideoEffect1Parameter1 and VideoEffect1Parameter2. Attributes without the wildcards (n) or (m) are not enumerated. The enumeration starts with 1. Attributes names are considered as normalized. The upper and lower case of attribute names is not taken into account.

```
<AttributeDefinitions>
    <ActivationGroups>
        <ActivationGroup  Name="PanTilt" />
        <ActivationGroup  Name="XYZ" />
        <ActivationGroup  Name="ColorRGB" />
        <ActivationGroup  Name="ColorHSB" />
        <ActivationGroup  Name="ColorCIE" />
        <ActivationGroup  Name="ColorIndirect" />
        <ActivationGroup  Name="Gobo(n)" />
        <ActivationGroup  Name="Gobo(n)Pos" />
        <ActivationGroup  Name="AnimationWheel(n)" />
        <ActivationGroup  Name="AnimationWheel(n)Pos" />
        <ActivationGroup  Name="AnimationSystem(n)" />
        <ActivationGroup  Name="AnimationSystem(n)Pos" />
        <ActivationGroup  Name="Prism" />
        <ActivationGroup  Name="BeamShaper" />
        <ActivationGroup  Name="Shaper" />
    </ActivationGroups>
    <FeatureGroups>
        <FeatureGroup  Name="Dimmer">
            <Feature  Name="Dimmer" />
        </FeatureGroup>
        <FeatureGroup  Name="Position">
            <Feature  Name="PanTilt" />
            <Feature  Name="XYZ" />
            <Feature  Name="Rotation"/>
            <Feature  Name="Scale"/>
        </FeatureGroup>
        <FeatureGroup  Name="Gobo">
            <Feature  Name="Gobo" />
            <Feature  Name="Media"/>
        </FeatureGroup>
        <FeatureGroup  Name="Color">
            <Feature  Name="Color" />
            <Feature  Name="RGB" />
            <Feature  Name="HSB" />
            <Feature  Name="CIE" />
            <Feature  Name="Indirect" />
            <Feature Name="ColorCorrection"/>
            <Feature Name="HSBC_Shift"/>
            <Feature Name="ColorKey"/>
        </FeatureGroup>
        <FeatureGroup  Name="Beam">
            <Feature  Name="Beam" />
        </FeatureGroup>
        <FeatureGroup  Name="Focus">
            <Feature  Name="Focus" />
        </FeatureGroup>
        <FeatureGroup  Name="Control">
            <Feature  Name="Control" />
        </FeatureGroup>
        <FeatureGroup  Name="Shapers">
            <Feature  Name="Shapers" />
        </FeatureGroup>
        <FeatureGroup  Name="Video">
            <Feature  Name="Video" />
```

```
    </FeatureGroup>
</FeatureGroups>
<Attributes>
    <Attribute Name="Dimmer" Pretty="Dim" Feature="Dimmer.Dimmer" />
    <Attribute
        Name="Pan"
        Pretty="P"
        ActivationGroup="PanTilt"
        Feature="Position.PanTilt"
        PhysicalUnit="Angle" />
    <Attribute
        Name="Tilt"
        Pretty="T"
        ActivationGroup="PanTilt"
        Feature="Position.PanTilt"
        PhysicalUnit="Angle" />
    <Attribute
        Name="PanRotate"
        Pretty="P Rotate"
        Feature="Position.PanTilt"
        PhysicalUnit="AngularSpeed" />
    <Attribute
        Name="TiltRotate"
        Pretty="T Rotate"
        Feature="Position.PanTilt"
        PhysicalUnit="AngularSpeed" />
    <Attribute
        Name="PositionEffect"
        Pretty="Pos FX"
        Feature="Position.PanTilt" />
    <Attribute
        Name="PositionEffectRate"
        Pretty="Pos FX Rate"
        Feature="Position.PanTilt" />
    <Attribute
        Name="PositionEffectFade"
        Pretty="Pos FX Fade"
        Feature="Position.PanTilt" />
    <Attribute
        Name="XYZ_X"
        Pretty="X"
        ActivationGroup="XYZ"
        Feature="Position.XYZ"
        PhysicalUnit="Length" />
    <Attribute
        Name="XYZ_Y"
        Pretty="Y"
        ActivationGroup="XYZ"
        Feature="Position.XYZ"
        PhysicalUnit="Length" />
    <Attribute
        Name="XYZ_Z"
        Pretty="Z"
        ActivationGroup="XYZ"
        Feature="Position.XYZ"
        PhysicalUnit="Length" /
    <Attribute
        Name="Rot_X"
        Pretty="Rot X"
        Feature="Position.Rotation"
        PhysicalUnit="Angle" />
    <Attribute
        Name="Rot_Y"
        Pretty="Rot Y"
        Feature="Position.Rotation"
        PhysicalUnit="Angle" />
    <Attribute
        Name="Rot_Z"
        Pretty="Rot Z"
        Feature="Position.Rotation"
        PhysicalUnit="Angle" />
    <Attribute
        Name="Scale_X"
        Pretty="Scale X"
        Feature="Position.Scale"
```

```
     PhysicalUnit="Percent" />
<Attribute
     Name="Scale_Y"
     Pretty="Scale Y"
     Feature="Position.Scale"
     PhysicalUnit="Percent" />
<Attribute
     Name="Scale_Z"
     Pretty="Scale Z"
     Feature="Position.Scale"
     PhysicalUnit="Percent" />
<Attribute
     Name="Scale_XYZ"
     Pretty="Scale XYZ"
     Feature="Position.Scale"
     PhysicalUnit="Percent" />
<Attribute
     Name="Gobo(n)"
     Pretty="G(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo" />
<Attribute
     Name="Gobo(n)SelectSpin"
     Pretty="Select Spin"
     MainAttribute="Gobo(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="AngularSpeed" />
<Attribute
     Name="Gobo(n)SelectShake"
     Pretty="Select Shake"
     MainAttribute="Gobo(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Gobo(n)SelectEffects"
     Pretty="Select Effects"
     MainAttribute="Gobo(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo" />
<Attribute
     Name="Gobo(n)WheelIndex"
     Pretty="Wheel Index"
     MainAttribute="Gobo(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Angle" />
<Attribute
     Name="Gobo(n)WheelSpin"
     Pretty="Wheel Spin"
     MainAttribute="Gobo(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="AngularSpeed" />
<Attribute
     Name="Gobo(n)WheelShake"
     Pretty="Wheel Shake"
     MainAttribute="Gobo(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Gobo(n)WheelRandom"
     Pretty="Wheel Random"
     MainAttribute="Gobo(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Gobo(n)WheelAudio"
     Pretty="Wheel Audio"
     MainAttribute="Gobo(n)"
     ActivationGroup="Gobo(n)"
     Feature="Gobo.Gobo" />
```

```
<Attribute
     Name="Gobo(n)Pos"
     Pretty="G(n) <>"
     ActivationGroup="Gobo(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="Angle" />
<Attribute
     Name="Gobo(n)PosRotate"
     Pretty="Rotate"
     MainAttribute="Gobo(n)Pos"
     ActivationGroup="Gobo(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="AngularSpeed" />
<Attribute
     Name="Gobo(n)PosShake"
     Pretty="Shake"
     MainAttribute="Gobo(n)Pos"
     ActivationGroup="Gobo(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency" />
<Attribute
     Name="AnimationWheel(n)"
     ActivationGroup="AnimationWheel(n)"
     Pretty="Anim(n)"
     Feature="Gobo.Gobo" />
<Attribute
     Name="AnimationWheel(n)Audio"
     ActivationGroup="AnimationWheel(n)"
     MainAttribute="AnimationWheel(n)"
     Feature="Gobo.Gobo"
     Pretty="Anim Audio" />
<Attribute
     Name="AnimationWheel(n)Macro"
     ActivationGroup="AnimationWheel(n)"
     MainAttribute="AnimationWheel(n)"
     Feature="Gobo.Gobo"
     Pretty="Anim FX" />
<Attribute
     Name="AnimationWheel(n)Random"
     ActivationGroup="AnimationWheel(n)"
     MainAttribute="AnimationWheel(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency"
     Pretty="Anim Random" />
<Attribute
     Name="AnimationWheel(n)SelectEffects"
     ActivationGroup="AnimationWheel(n)"
     MainAttribute="AnimationWheel(n)"
     Feature="Gobo.Gobo"
     Pretty="Anim Select FX" />
<Attribute
     Name="AnimationWheel(n)SelectShake"
     ActivationGroup="AnimationWheel(n)"
     MainAttribute="AnimationWheel(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency"
     Pretty="Anim Select Shake" />
<Attribute
     Name="AnimationWheel(n)SelectSpin"
     ActivationGroup="AnimationWheel(n)"
     MainAttribute="AnimationWheel(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="AngularSpeed"
     Pretty="Anim Select Spin" />
<Attribute
     Name="AnimationWheel(n)Pos"
     ActivationGroup="AnimationWheel(n)Pos"
     Feature="Gobo.Gobo" PhysicalUnit="Angle"
     Pretty="Anim Pos" />
<Attribute
     Name="AnimationWheel(n)PosRotate"
     ActivationGroup="AnimationWheel(n)Pos"
     MainAttribute="AnimationWheel(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="AngularSpeed"
```

```
     Pretty="Anim Rotate" />
<Attribute
     Name="AnimationWheel(n)PosShake"
     ActivationGroup="AnimationWheel(n)Pos"
     MainAttribute="AnimationWheel(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency"
     Pretty="Anim Shake" />
 <Attribute
     Name="AnimationSystem(n)"
     ActivationGroup="AnimationSystem(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Percent"
     Pretty="Anim System"/>
<Attribute
     Name="AnimationSystem(n)Ramp"
     ActivationGroup="AnimationSystem(n)"
     MainAttribute="AnimationSystem(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency"
     Pretty="Anim System Ramp"/>
<Attribute
     Name="AnimationSystem(n)Shake"
     ActivationGroup="AnimationSystem(n)"
     MainAttribute="AnimationSystem(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency"
     Pretty="Anim System Shake/>
<Attribute
     Name="AnimationSystem(n)Audio"
     ActivationGroup="AnimationSystem(n)"
     MainAttribute="AnimationSystem(n)"
     Feature="Gobo.Gobo" PhysicalUnit="None"
     Pretty="Anim System Audio/>
<Attribute
     Name="AnimationSystem(n)Random"
     ActivationGroup="AnimationSystem(n)"
     MainAttribute="AnimationSystem(n)"
     Feature="Gobo.Gobo"
     PhysicalUnit="None"
     Pretty="Anim System Random/>
<Attribute
     Name="AnimationSystem(n)Pos"
     ActivationGroup="AnimationSystem(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="Angle"
     Pretty="Anim System Pos"/>
<Attribute
     Name="AnimationSystem(n)PosRotate"
     ActivationGroup="AnimationSystem(n)Pos"
     MainAttribute="AnimationSystem(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="AngularSpeed"
     Pretty="Anim System Rotate"/>
<Attribute
     Name="AnimationSystem(n)PosShake"
     ActivationGroup="AnimationSystem(n)Pos"
     MainAttribute="AnimationSystem(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="Frequency"
     Pretty="Anim System Shake"/>
<Attribute
     Name="AnimationSystem(n)PosRandom"
     ActivationGroup="AnimationSystem(n)Pos"
     MainAttribute="AnimationSystem(n)Pos"
     Feature="Gobo.Gobo" PhysicalUnit="None"
     Pretty="Anim System Rot Random"/>
<Attribute
     Name="AnimationSystem(n)PosAudio"
     ActivationGroup="AnimationSystem(n)Pos"
     MainAttribute="AnimationSystem(n)Pos"
     Feature="Gobo.Gobo"
     PhysicalUnit="None"
     Pretty="Anim System Rot Audio"/>
```

```
<Attribute
     Name="AnimationSystem(n)Macro"
     Feature="Gobo.Gobo"
     PhysicalUnit="None"
     Pretty="Anim System Macro"/>
<Attribute Name="MediaFolder(n)" Pretty="Media Folder(n)" Feature="Gobo.Media" />
<Attribute Name="MediaContent(n)" Pretty="Media Content(n)" Feature="Gobo.Media" />
<Attribute
     Name="ModelFolder(n)"
     Pretty="Model Folder(n)"
     Feature="Gobo.Media"
     PhysicalUnit="None" />
<Attribute
     Name="ModelContent(n)"
     Pretty="Model Content(n)"
     Feature="Gobo.Media"
     PhysicalUnit="None" />
<Attribute Name="Playmode" Pretty="Playmode" Feature="Gobo.Media" />
<Attribute
     Name="PlayBegin"
     Pretty="Play Begin"
     Feature="Gobo.Media"
     PhysicalUnit="Time" />
<Attribute Name="PlayEnd" Pretty="Play End" Feature="Gobo.Media" PhysicalUnit="Time" />
<Attribute
     Name="PlaySpeed"
     Pretty="Play Speed"
     Feature="Gobo.Media"
     PhysicalUnit="Percent" />
<Attribute Name="ColorEffects(n)" Pretty="Color FX(n)" Feature="Color.Color" />
<Attribute
     Name="Color(n)"
     Pretty="C(n)"
     ActivationGroup="ColorRGB"
     Feature="Color.Color" />
<Attribute
     Name="Color(n)WheelIndex"
     Pretty="Wheel Index"
     MainAttribute="Color(n)"
     ActivationGroup="ColorRGB"
     Feature="Color.Color"
     PhysicalUnit="Angle" />
<Attribute
     Name="Color(n)WheelSpin"
     Pretty="Wheel Spin"
     MainAttribute="Color(n)"
     ActivationGroup="ColorRGB"
     Feature="Color.Color"
     PhysicalUnit="AngularSpeed" />
<Attribute
     Name="Color(n)WheelRandom"
     Pretty="Wheel Random"
     MainAttribute="Color(n)"
     ActivationGroup="ColorRGB"
     Feature="Color.Color"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Color(n)WheelAudio"
     Pretty="Wheel Audio"
     MainAttribute="Color(n)"
     ActivationGroup="ColorRGB"
     Feature="Color.Color" />
<Attribute
     Name="ColorAdd_R"
     Pretty="R"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.64,0.33,21.3" />
<Attribute
     Name="ColorAdd_G"
     Pretty="G"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
```

```
     Color="0.3,0.6,71.5" />
<Attribute
     Name="ColorAdd_B"
     Pretty="B"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.15,0.06,7.2" />
<Attribute
     Name="ColorAdd_C"
     Pretty="C"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.225,0.329,78.7" />
<Attribute
     Name="ColorAdd_M"
     Pretty="M"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.321,0.154,28.5" />
<Attribute
     Name="ColorAdd_Y"
     Pretty="Y"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.419,0.505,92.8" />
<Attribute
     Name="ColorAdd_RY"
     Pretty="Amber"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.477,0.460,57.0" />
<Attribute
     Name="ColorAdd_GY"
     Pretty="Lime"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.372,0.543,82.1" />
<Attribute
     Name="ColorAdd_GC"
     Pretty="Blue-Green"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.250,0.419,75.1" />
<Attribute
     Name="ColorAdd_BC"
     Pretty="Light-Blue"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.200,0.239,43.0" />
<Attribute
     Name="ColorAdd_BM"
     Pretty="Purple"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.254,0.117,17.9" />
<Attribute
     Name="ColorAdd_RM"
     Pretty="Pink"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.403,0.200,24.9" />
<Attribute
     Name="ColorAdd_W"
     Pretty="White"
     ActivationGroup="ColorRGB"
```

```
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.313,0.329,100.0" />
<Attribute
     Name="ColorAdd_WW"
     Pretty="WW"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.319,0.340,99.3" />
<Attribute
     Name="ColorAdd_CW"
     Pretty="CW"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.306,0.329,97.9" />
<Attribute
     Name="ColorAdd_UV"
     Pretty="UV"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.176,0.005,0.6"/>
<Attribute
     Name="ColorSub_R"
     Pretty="R"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.64,0.33,21.3" />
<Attribute
     Name="ColorSub_G"
     Pretty="G"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.3,0.6,71.5" />
<Attribute
     Name="ColorSub_B"
     Pretty="B"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.15,0.06,7.2" />
<Attribute
     Name="ColorSub_C"
     Pretty="C"
     ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.225,0.329,78.7" />
<Attribute
     Name="ColorSub_M"
     Pretty="M" ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.321,0.154,28.5" />
<Attribute
     Name="ColorSub_Y"
     Pretty="Y" ActivationGroup="ColorRGB"
     Feature="Color.RGB"
     PhysicalUnit="ColorComponent"
     Color="0.419,0.505,92.8" />
<Attribute Name="ColorMacro(n)" Pretty="Color Macro(n)" Feature="Color.RGB" />
<Attribute Name="ColorMacro(n)Rate" Pretty="Color Macro(n) Rate" Feature="Color.RGB" />
<Attribute Name="CTO" Pretty="CTO" Feature="Color.Color" PhysicalUnit="Temperature" />
<Attribute
     Name="CTC"
     Pretty="CTC"
     Feature="Color.Color"
     PhysicalUnit="Temperature" />
<Attribute
     Name="CTB"
     Pretty="CTB"
```

```
     Feature="Color.Color"
     PhysicalUnit="Temperature" />
<Attribute Name="Tint" Pretty="Tint" Feature="Color.Color" />
<Attribute
     Name="HSB_Hue"
     Pretty="H"
     ActivationGroup="ColorHSB"
     Feature="Color.HSB"
     PhysicalUnit="Angle" />
<Attribute
     Name="HSB_Saturation"
     Pretty="S" ActivationGroup="ColorHSB"
     Feature="Color.HSB"
     PhysicalUnit="Percent" />
<Attribute
     Name="HSB_Brightness"
     Pretty="B"
     ActivationGroup="ColorHSB"
     Feature="Color.HSB"
     PhysicalUnit="Percent" />
<Attribute
     Name="HSB_Quality"
     Pretty="Q"
     ActivationGroup="ColorHSB"
     Feature="Color.HSB"
     PhysicalUnit="Percent" />
<Attribute
     Name="CIE_X"
     Pretty="X"
     ActivationGroup="ColorCIE"
     Feature="Color.CIE" />
<Attribute
     Name="CIE_Y"
     Pretty="Y"
     ActivationGroup="ColorCIE"
     Feature="Color.CIE" />
<Attribute
     Name="CIE_Brightness"
     Pretty="B"
     ActivationGroup="ColorCIE"
     Feature="Color.CIE"
     PhysicalUnit="Percent" />
<Attribute
     Name="ColorRGB_Red"
     Pretty="R"
     ActivationGroup="ColorIndirect"
     Feature="Color.Indirect" />
<Attribute
     Name="ColorRGB_Green"
     Pretty="G"
     ActivationGroup="ColorIndirect"
     Feature="Color.Indirect" />
<Attribute
     Name="ColorRGB_Blue"
     Pretty="B" ActivationGroup="ColorIndirect"
     Feature="Color.Indirect" />
<Attribute
     Name="ColorRGB_Cyan"
     Pretty="C" ActivationGroup="ColorIndirect"
     Feature="Color.Indirect" />
<Attribute
     Name="ColorRGB_Magenta"
     Pretty="M" ActivationGroup="ColorIndirect"
     Feature="Color.Indirect" />
<Attribute
     Name="ColorRGB_Yellow"
     Pretty="Y"
     ActivationGroup="ColorIndirect"
     Feature="Color.Indirect" />
<Attribute
     Name="ColorRGB_Quality"
     Pretty="Q"
     ActivationGroup="ColorIndirect"
     Feature="Color.Indirect" />
<Attribute
```

```
     Name="VideoBoost_R"
     Feature="Color.ColorCorrection"
     PhysicalUnit="None"
     Pretty="Boost R"
     Color="0.64,0.33,21.3" />
<Attribute
     Name="VideoBoost_G"
     Pretty="Boost G"
     Feature="Color.ColorCorrection"
     PhysicalUnit="None"
     Color="0.3,0.6,71.5" />
<Attribute
     Name="VideoBoost_B"
     Pretty="Boost B"
     Feature="Color.ColorCorrection"
     PhysicalUnit="None"
     Color="0.15,0.06,7.2" />
<Attribute
     Name="VideoHueShift"
     Pretty="Hue Shift"
     Feature="Color.HSBC_Shift"
     PhysicalUnit="Angle" />
<Attribute
     Name="VideoSaturation"
     Pretty="S"
     Feature="Color.HSBC_Shift"
     PhysicalUnit="Percent" />
<Attribute
     Name="VideoBrightness"
     Pretty="B"
     Feature="Color.HSBC_Shift"
     PhysicalUnit="Percent" />
<Attribute
     Name="VideoContrast"
     Pretty="C"
     Feature="Color.HSBC_Shift"
     PhysicalUnit="Percent" />
<Attribute
     Name="VideoKeyColor_R"
     Pretty="R"
     Feature="Color.ColorKey"
     PhysicalUnit="None"
     Color="0.64,0.33,21.3" />
<Attribute
     Name="VideoKeyColor_G"
     Pretty="G" Feature="Color.ColorKey"
     PhysicalUnit="None"
     Color="0.3,0.6,71.5" />
<Attribute
     Name="VideoColorKey_B"
     Pretty="B" Feature="Color.ColorKey"
     PhysicalUnit="None"
     Color="0.15,0.06,7.2" />
<Attribute
     Name="VideoKeyIntensity"
     Pretty="Intensity"
     Feature="Color.ColorKey"
     PhysicalUnit="Percent" />
<Attribute
     Name="VideoKeyTolerance"
     Pretty="Tolerance"
     Feature="Color.ColorKey"
     PhysicalUnit="None" />
<Attribute
     Name="StrobeDuration"
     Pretty="Strobe Duration"
     Feature="Beam.Beam"
     PhysicalUnit="Time" />
<Attribute
     Name="StrobeRate"
     Pretty="Strobe Rate"
     Feature="Beam.Beam" />
<Attribute
     Name="Shutter(n)"
     Pretty="Sh(n)"
```

```
     Feature="Beam.Beam" />
<Attribute
     Name="Shutter(n)Strobe"
     Pretty="Strobe(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Shutter(n)StrobePulse"
     Pretty="Pulse(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Shutter(n)StrobePulseClose"
     Pretty="Pulse Close(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Shutter(n)StrobePulseOpen"
     Pretty="Pulse Open(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Shutter(n)StrobeRandom"
     Pretty="Random(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Shutter(n)StrobeRandomPulse"
     Pretty="Random Pulse(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Shutter(n)StrobeRandomPulseClose"
     Pretty="Random Pulse Close(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Shutter(n)StrobeRandomPulseOpen"
     Pretty="Random Pulse Open(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Shutter(n)StrobeEffect"
     Pretty="Effect(n)"
     MainAttribute="Shutter(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute Name="Iris" Pretty="Iris" Feature="Beam.Beam" />
<Attribute
     Name="IrisStrobe"
     Pretty="Strobe"
     MainAttribute="Iris"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="IrisStrobeRandom"
     Pretty="Random Strobe"
     MainAttribute="Iris"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="IrisPulseClose"
     Pretty="Pulse Close"
     MainAttribute="Iris"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
```

```
     Name="IrisPulseOpen"
     Pretty="Pulse Open"
     MainAttribute="Iris"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="IrisRandomPulseClose"
     Pretty="Random Pulse Close"
     MainAttribute="Iris"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="IrisRandomPulseOpen"
     Pretty="Random Pulse Open"
     MainAttribute="Iris"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Frost(n)"
     Pretty="Frost(n)"
     Feature="Beam.Beam" />
<Attribute
     Name="Frost(n)PulseOpen"
     Pretty="Pulse Open (n)"
     MainAttribute="Frost(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Frost(n)PulseClose"
     Pretty="Pulse Close (n)"
     MainAttribute="Frost(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Frost(n)Ramp"
     Pretty="Ramp (n)"
     MainAttribute="Frost(n)"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
<Attribute
     Name="Prism(n)"
     Pretty="Prism(n)"
     ActivationGroup="Prism"
     Feature="Beam.Beam" />
<Attribute
     Name="Prism(n)SelectSpin"
     Pretty="Select Spin(n)"
     MainAttribute="Prism(n)"
     ActivationGroup="Prism"
     Feature="Beam.Beam"
     PhysicalUnit="AngularSpeed" />
<Attribute
     Name="Prism(n)Macro"
     Pretty="Prism(n) Macro"
     MainAttribute="Prism(n)"
     ActivationGroup="Prism"
     Feature="Beam.Beam" />
<Attribute
     Name="Prism(n)Pos"
     Pretty="Prism(n) Pos"
     Feature="Beam.Beam"
     PhysicalUnit="Angle" />
<Attribute
     Name="Prism(n)PosRotate"
     Pretty="Rotate(n)"
     MainAttribute="Prism(n)Pos"
     ActivationGroup="Prism"
     Feature="Beam.Beam"
     PhysicalUnit="AngularSpeed" />
<Attribute Name="Effects(n)" Pretty="FX(n)" Feature="Beam.Beam" />
<Attribute
     Name="Effects(n)Rate"
     Pretty="FX(n) Rate"
     Feature="Beam.Beam"
     PhysicalUnit="Frequency" />
```

```
<Attribute Name="Effects(n)Fade" Pretty="FX(n) Fade" Feature="Beam.Beam" />
<Attribute Name="Effects(n)Adjust(m)" Pretty="FX(n) Adjust(m)" Feature="Beam.Beam" />
<Attribute
     Name="Effects(n)Pos"
     Pretty="FX(n) Pos"
     Feature="Beam.Beam"
     PhysicalUnit="Angle" />
<Attribute
     Name="Effects(n)PosRotate"
     Pretty="FX(n) Rotate"
     MainAttribute="Effects(n)Pos"
     Feature="Beam.Beam"
     PhysicalUnit="AngularSpeed" />
<Attribute Name="EffectsSync" Pretty="FX Sync" Feature="Beam.Beam" />
<Attribute
     Name="BeamShaper"
     Pretty="Beam Shaper"
     ActivationGroup="BeamShaper"
     Feature="Beam.Beam" />
<Attribute
     Name="BeamShaperMacro"
     Pretty="Beam Shaper Macro"
     ActivationGroup="BeamShaper"
     Feature="Beam.Beam" />
<Attribute
     Name="BeamShaperPos"
     Pretty="Beam Shaper <>"
     ActivationGroup="BeamShaper"
     Feature="Beam.Beam" />
<Attribute
     Name="BeamShaperPosRotate"
     Pretty="Beam Shaper Rotate"
     ActivationGroup="BeamShaper"
     Feature="Beam.Beam" />
<Attribute Name="Zoom" Pretty="Zoom" Feature="Focus.Focus" PhysicalUnit="Angle" />
<Attribute
     Name="ZoomModeSpot"
     Pretty="Zoom Spot"
     Feature="Focus.Focus"
     PhysicalUnit="Angle" />
<Attribute
     Name="ZoomModeBeam"
     Pretty="Zoom Beam"
     Feature="Focus.Focus"
     PhysicalUnit="Angle" />
<Attribute Name="Focus(n)" Pretty="Focus(n)" Feature="Focus.Focus" />
<Attribute Name="Focus(n)Adjust" Pretty="Focus(n) Adjust" Feature="Focus.Focus" />
<Attribute
     Name="Focus(n)Distance"
     Pretty="Focus(n) Distance"
     Feature="Focus.Focus"
     PhysicalUnit="Length" />
<Attribute Name="Control(n)" Pretty="Ctrl(n)" Feature="Control.Control" />
<Attribute Name="DimmerMode" Pretty="Dim Mode" Feature="Control.Control" />
<Attribute Name="DimmerCurve" Pretty="Dim Curve" Feature="Control.Control" />
<Attribute Name="BlackoutMode" Pretty="Blackout Mode" Feature="Control.Control" />
<Attribute
     Name="LEDFrequency"
     Pretty="LED Frequency"
     Feature="Control.Control"
     PhysicalUnit="Frequency"/>
<Attribute Name="LEDZoneMode" Pretty="LED Zone Mode" Feature="Control.Control" />
<Attribute Name="PixelMode" Pretty="Pixel Mode" Feature="Control.Control" />
<Attribute Name="PanMode" Pretty="Pan Mode" Feature="Control.Control" />
<Attribute Name="TiltMode" Pretty="Tilt Mode" Feature="Control.Control" />
<Attribute Name="PanTiltMode" Pretty="PanTilt Mode" Feature="Control.Control" />
<Attribute Name="PositionModes" Pretty="Pos Modes" Feature="Control.Control" />
<Attribute Name="Gobo(n)WheelMode" Pretty="G(n) Mode" Feature="Control.Control" />
<Attribute Name="AnimationWheel(n)Mode" Feature="Control.Control" Pretty="Anim Mode" />
<Attribute
     Name="AnimationWheelShortcutMode"
     Pretty="Anim Shortcut Mode"
     Feature="Control.Control" />
<Attribute Name="Color(n)Mode" Pretty="C(n) Mode" Feature="Control.Control" />
<Attribute
```

```
     Name="ColorWheelShortcutMode"
     Pretty="Color Wheel Shortcut Mode"
     Feature="Control.Control" />
<Attribute Name="CyanMode" Pretty="Cyan Mode" Feature="Control.Control" />
<Attribute Name="MagentaMode" Pretty="Magenta Mode" Feature="Control.Control" />
<Attribute Name="YellowMode" Pretty="Yellow Mode" Feature="Control.Control" />
<Attribute Name="ColorMixMode" Pretty="Color Mix Mode" Feature="Control.Control" />
<Attribute Name="ChromaticMode" Pretty="Chroma Mode" Feature="Control.Control" />
<Attribute Name="ColorCalibrationMode" Pretty="CCalib Mode" Feature="Control.Control" />
<Attribute
     Name="ColorConsistency"
     Pretty="Color consistency"
     Feature="Control.Control" />
<Attribute Name="ColorControl" Pretty="Color Ctrl" Feature="Control.Control" />
<Attribute Name="ColorModelMode" Pretty="ColorModel" Feature="Control.Control" />
<Attribute Name="ColorSettingsReset" Pretty="Color Ctrl Rst" Feature="Control.Control" />
<Attribute Name="ColorUniformity" Pretty="ColorUniform" Feature="Control.Control" />
<Attribute Name="CRIMode" Pretty="CRI Mode" Feature="Control.Control" />
<Attribute Name="CustomColor" Pretty="Custom Color" Feature="Control.Control" />
<Attribute Name="UVStability" Pretty="UV Stab" Feature="Control.Control" />
<Attribute Name="WaveLengthCorrection" Pretty="WaveLength" Feature="Control.Control" />
<Attribute Name="WhiteCount" Pretty="White Count" Feature="Control.Control" />
<Attribute Name="StrobeMode" Pretty="Strobe Mode" Feature="Control.Control" />
<Attribute Name="ZoomMode" Pretty="Zoom Mode" Feature="Control.Control" />
<Attribute Name="FocusMode" Pretty="Focus Mode" Feature="Control.Control" />
<Attribute Name="IrisMode" Pretty="Iris Mode" Feature="Control.Control" />
<Attribute Name="FanMode" Pretty="Fan Mode" Feature="Control.Control" />
<Attribute Name="FollowSpotMode" Pretty="FollowSpot Mode" Feature="Control.Control" />
<Attribute
     Name="BeamEffectIndexRotateMode"
     Pretty="Beam Effect Index Rotate Mode"
     Feature="Control.Control" />
<Attribute Name="IntensityMSpeed" Pretty="Intensity MSpeed" Feature="Control.Control" />
<Attribute Name="PositionMSpeed" Pretty="Pos MSpeed" Feature="Control.Control" />
<Attribute Name="ColorMixMSpeed" Pretty="Color Mix MSpeed" Feature="Control.Control" />
<Attribute
     Name="ColorWheelSelectMSpeed"
     Pretty="Color Wheel Select MSpeed"
     Feature="Control.Control" />
<Attribute
     Name="GoboWheel(n)MSpeed"
     Pretty="Gobo Wheel(n) MSpeed"
     Feature="Control.Control" />
<Attribute Name="IrisMSpeed" Pretty="Iris MSpeed" Feature="Control.Control" />
<Attribute Name="Prism(n)MSpeed" Pretty="Prism(n) MSpeed" Feature="Control.Control" />
<Attribute Name="FocusMSpeed" Pretty="Focus MSpeed" Feature="Control.Control" />
<Attribute Name="Frost(n)MSpeed" Pretty="Frost(n) MSpeed" Feature="Control.Control" />
<Attribute Name="ZoomMSpeed" Pretty="Zoom MSpeed" Feature="Control.Control" />
<Attribute Name="FrameMSpeed" Pretty="Frame MSpeed" Feature="Control.Control" />
<Attribute Name="GlobalMSpeed" Pretty="Global MSpeed" Feature="Control.Control" />
<Attribute Name="ReflectorAdjust"  Pretty="Reflector Adj" Feature="Control.Control" /> />
<Attribute
     Name="FixtureGlobalReset"
     Pretty="Fixture Global Reset"
     Feature="Control.Control" />
<Attribute Name="ShutterReset" Pretty="Shutter Reset" Feature="Control.Control" />
<Attribute Name="BeamReset" Pretty="Beam Reset" Feature="Control.Control" />
<Attribute Name="ColorMixReset" Pretty="Color Mix Reset" Feature="Control.Control" />
<Attribute Name="ColorWheelReset" Pretty="Color Wheel Reset" Feature="Control.Control" />
<Attribute Name="FocusReset" Pretty="Focus Reset" Feature="Control.Control" />
<Attribute Name="FrameReset" Pretty="Frame Reset" Feature="Control.Control" />
<Attribute Name="GoboWheelReset" Pretty="G Reset" Feature="Control.Control" />
<Attribute Name="IntensityReset" Pretty="Intensity Reset" Feature="Control.Control" />
<Attribute Name="IrisReset" Pretty="Iris Reset" Feature="Control.Control" />
<Attribute Name="PositionReset" Pretty="Pos Reset" Feature="Control.Control" />
<Attribute Name="PanReset" Pretty="Pan Reset" Feature="Control.Control" />
<Attribute Name="TiltReset" Pretty="Tilt Reset" Feature="Control.Control" />
<Attribute Name="ZoomReset" Pretty="Zoom Reset" Feature="Control.Control" />
<Attribute Name="CTBReset" Pretty="CTB Reset" Feature="Control.Control" />
<Attribute Name="CTOReset" Pretty="CTO Reset" Feature="Control.Control" />
<Attribute Name="CTCReset" Pretty="CTC Reset" Feature="Control.Control" />
<Attribute
     Name="AnimationSystemReset"
     Pretty="Anim System Reset"
     Feature="Control.Control" />
```

```
<Attribute
     Name="FixtureCalibrationReset"
     Pretty="Fixture Calibration Reset"
     Feature="Control.Control" />
<Attribute Name="Function" Pretty="Function" Feature="Control.Control" />
<Attribute Name="LampControl" Pretty="Lamp Ctrl" Feature="Control.Control" />
<Attribute Name="DisplayIntensity" Pretty="Display Int" Feature="Control.Control" />
<Attribute Name="DMXInput" Pretty="DMX Input" Feature="Control.Control" />
<Attribute Name="NoFeature" Pretty="NoFeature" Feature="Control.Control" />
<Attribute Name="Dummy" Pretty="Dummy" Feature="Control.Control" />
<Attribute Name="Blower(n)" Pretty="Blower(n)" Feature="Control.Control" />
<Attribute Name="Fan(n)" Pretty="Fan(n)" Feature="Control.Control" />
<Attribute Name="Fog(n)" Pretty="Fog(n)" Feature="Control.Control" />
<Attribute Name="Haze(n)" Pretty="Haze(n)" Feature="Control.Control" />
<Attribute Name="LampPowerMode" Pretty="Lamp Power Mode" Feature="Control.Control" />
<Attribute Name="Fans" Pretty="Fans" Feature="Control.Control" />
<Attribute
     Name="Blade(n)A"
     Pretty="Blade(n)A"
     ActivationGroup="Shaper"
     Feature="Shapers.Shapers" />
<Attribute
     Name="Blade(n)B"
     Pretty="Blade(n)B"
     ActivationGroup="Shaper"
     Feature="Shapers.Shapers" />
<Attribute
     Name="Blade(n)Rot"
     Pretty="Blade(n) Rot"
     ActivationGroup="Shaper"
     Feature="Shapers.Shapers"
     PhysicalUnit="Angle" />
<Attribute
     Name="ShaperRot"
     Pretty="Shaper Rot"
     ActivationGroup="Shaper"
     Feature="Shapers.Shapers"
     PhysicalUnit="Angle" />
<Attribute Name="ShaperMacros" Pretty="Shaper Macros" Feature="Shapers.Shapers" />
<Attribute
     Name="ShaperMacrosSpeed"
     Pretty="Shaper Macros Speed"
     Feature="Shapers.Shapers" />
<Attribute
     Name="BladeSoft(n)A
     Pretty="BladeS(n)A"
     Feature="Shapers.Shapers"
     PhysicalUnit="None" />
<Attribute
     Name="BladeSoft(n)B"
     Pretty="BladeS(n)B"
     Feature="Shapers.Shapers"
     PhysicalUnit="None" />
<Attribute
     Name="KeyStone(n)A"
     Pretty="KS(n)A"
     Feature="Shapers.Shapers"
     PhysicalUnit="None" />
<Attribute
     Name="KeyStone(n)B"
     Pretty="KS(n)B"
     Feature="Shapers.Shapers"
     PhysicalUnit="None" />
<Attribute
     Name="Video"
     Pretty="Video"
     Feature="Video.Video" />
<Attribute
     Name="VideoEffect(n)Type"
     Pretty="Video Effect(n)
     Type" Feature="Video.Video" />
<Attribute
     Name="VideoEffect(n)Parameter(m)"
     Pretty="Video Effect(n) Parameter(m)"
     Feature="Video.Video" />
```

```
        <Attribute Name="VideoCamera(n)" Pretty="Video Camera(n)" Feature="Video.Video" />
        <Attribute Name="FieldOfView" Pretty="FOV" Feature="Video.Video" PhysicalUnit="Angle" />
        <Attribute Name="InputSource" Pretty="ISrc" Feature="Video.Video" PhysicalUnit="None" />
        <Attribute
             Name="VideoBlendMode"
             Pretty="BlendMode"
             Feature="Video.Video"
             PhysicalUnit="None" />
        <Attribute
             Name="VideoSoundVolume(n)"
             Pretty="Volume(n)"
             Feature="Video.Video"
             PhysicalUnit="Percent" />
    </Attributes>
</AttributeDefinitions>
```

Example for enumeration:

```
<Attribute Name="Gobo1" Pretty="G1" ActivationGroup="Gobo1" Feature="Gobo.Gobo" />
<Attribute Name="Gobo2" Pretty="G2" ActivationGroup="Gobo2" Feature="Gobo.Gobo" />
```

Annex C
(informative)

Name Character Table

Names are UTF-8 literals. In the first 128 characters only use characters listed in Table C.1. All characters above `127` are available.

Table C.1 — UTF-8 table

Unicode code point	Character	UTF-8 (dec.)	Name
U+0020		32	SPACE
U+0022	"	34	QUOTATION MARK
U+0023	#	35	NUMBER SIGN
U+0025	%	37	PERCENT SIGN
U+0027	'	39	APOSTROPHE
U+0028	(	40	LEFT PARENTHESIS
U+0029	)	41	RIGHT PARENTHESIS
U+002A	*	42	ASTERISK
U+002B	+	43	PLUS SIGN
U+002D	-	45	HYPHEN-MINUS
U+002F	/	47	SOLIDUS
U+0030	0	48	DIGIT ZERO
U+0031	1	49	DIGIT ONE
U+0032	2	50	DIGIT TWO
U+0033	3	51	DIGIT THREE
U+0034	4	52	DIGIT FOUR
U+0035	5	53	DIGIT FIVE
U+0036	6	54	DIGIT SIX
U+0037	7	55	DIGIT SEVEN
U+0038	8	56	DIGIT EIGHT
U+0039	9	57	DIGIT NINE
U+003A	:	58	COLON
U+003B	;	59	SEMICOLON
U+003C	<	60	LESS-THAN SIGN
U+003D	=	61	EQUALS SIGN
U+003E	>	62	GREATER-THAN SIGN

Unicode code point	Character	UTF-8 (dec.)	Name
U+0040	@	64	COMMERCIAL AT
U+0041	A	65	LATIN CAPITAL LETTER A
U+0042	B	66	LATIN CAPITAL LETTER B
U+0043	C	67	LATIN CAPITAL LETTER C
U+0044	D	68	LATIN CAPITAL LETTER D
U+0045	E	69	LATIN CAPITAL LETTER E
U+0046	F	70	LATIN CAPITAL LETTER F
U+0047	G	71	LATIN CAPITAL LETTER G
U+0048	H	72	LATIN CAPITAL LETTER H
U+0049	I	73	LATIN CAPITAL LETTER I
U+004A	J	74	LATIN CAPITAL LETTER J
U+004B	K	75	LATIN CAPITAL LETTER K
U+004C	L	76	LATIN CAPITAL LETTER L
U+004D	M	77	LATIN CAPITAL LETTER M
U+004E	N	78	LATIN CAPITAL LETTER N
U+004F	O	79	LATIN CAPITAL LETTER O
U+0050	P	80	LATIN CAPITAL LETTER P
U+0051	Q	81	LATIN CAPITAL LETTER Q
U+0052	R	82	LATIN CAPITAL LETTER R
U+0053	S	83	LATIN CAPITAL LETTER S
U+0054	T	84	LATIN CAPITAL LETTER T
U+0055	U	85	LATIN CAPITAL LETTER U
U+0056	V	86	LATIN CAPITAL LETTER V
U+0057	W	87	LATIN CAPITAL LETTER W
U+0058	X	88	LATIN CAPITAL LETTER X
U+0059	Y	89	LATIN CAPITAL LETTER Y
U+005A	Z	90	LATIN CAPITAL LETTER Z
U+005F	_	95	LOW LINE
U+0060	`	96	GRAVE ACCENT
U+0061	a	97	LATIN SMALL LETTER A
U+0062	b	98	LATIN SMALL LETTER B
U+0063	c	99	LATIN SMALL LETTER C
U+0064	d	100	LATIN SMALL LETTER D
U+0065	e	101	LATIN SMALL LETTER E

Unicode code point	Character	UTF-8 (dec.)	Name
U+0066	f	102	LATIN SMALL LETTER F
U+0067	g	103	LATIN SMALL LETTER G
U+0068	h	104	LATIN SMALL LETTER H
U+0069	i	105	LATIN SMALL LETTER I
U+006A	j	106	LATIN SMALL LETTER J
U+006B	k	107	LATIN SMALL LETTER K
U+006C	l	108	LATIN SMALL LETTER L
U+006D	m	109	LATIN SMALL LETTER M
U+006E	n	110	LATIN SMALL LETTER N
U+006F	o	111	LATIN SMALL LETTER O
U+0070	p	112	LATIN SMALL LETTER P
U+0071	q	113	LATIN SMALL LETTER Q
U+0072	r	114	LATIN SMALL LETTER R
U+0073	s	115	LATIN SMALL LETTER S
U+0074	t	116	LATIN SMALL LETTER T
U+0075	u	117	LATIN SMALL LETTER U
U+0076	v	118	LATIN SMALL LETTER V
U+0077	w	119	LATIN SMALL LETTER W
U+0078	x	120	LATIN SMALL LETTER X
U+0079	y	121	LATIN SMALL LETTER Y
U+007A	z	122	LATIN SMALL LETTER Z

Annex D
(informative)

Predefined Connector Types

Table D.1 shows the predefined connector types.

Table D.1 — Predefined Connector Types

Type	Description
BNC	BNC connector
TBLK	Tag block
TAG	Solder tag
KRN	Krone block
STJ	Stereo jack
MSTJ	Mini stereo jack
RCA	Phono connector
SCART	SCART connector
SVIDEO	4-pin mini-DIN
MDIN4	4-pin mini-DIN
MDIN5	5-pin mini-DIN
MDIN6	6-pin mini-DIN
XLR3	3-pin XLR
XLR4	4-pin XLR
XLR5	5-pin XLR
RJ45	10/100 BaseT ethernet type
RJ11	Telephone type
DB9	9-pin D-type
DB15	15-pin D-type
DB25	25-pin D-type
DB37	37-pin D-type
DB50	50-pin D-type
HD15	15-pin D-type hi-density
HD25	25-pin D-type hi-density
DIN3	3-pin DIN
DIN5	5-pin DIN
EDAC20	EDAC 20-pin

Type	Description
EDAC38	EDAC 38-pin
EDAC56	EDAC 56-pin
EDAC90	EDAC 90-pin
EDAC120	EDAC 120-pin
DL96	DL 96-pin
SCSI68	SCSI connector 68-pin
IEE488	IEE488 connector 36-pin
CENT50	Centronics 50-pin
CENT36	Centronics 36-pin
CENT24	Centronics 24-pin
DisplayPort	DisplayPort connector
DVI	DVI connector
HDMI	HDMI connector
PS2	PS2 connector
TL-ST	TosLink connector
LCDUP	Fibre optic LC DUPLEX-type
SCDUP	Fibre optic SC DUPLEX-type
SC	Fibre optic SC-type
ST	Fibre optic ST-type
NL4	Speakon
CACOM	8-pin LS conn
USB	USB connector
N_CON	N connector
F_CON	F connector
IEC 60320-C7/C8	Eurostecker
CEE 7/7	Schutzkontakt
IEC 60320-C13/14	IEC 60320
Edison	Edison
Wieland	Wieland
16A-CEE-2P	16A-Blue
16A-CEE-2P-110	16A-Yellow
16A-CEE	16A-CEE
32A-CEE	32A-CEE
32A-CEE-2P	32A-Blue

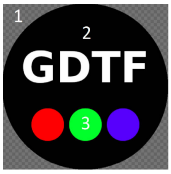
Type	Description
32A-CEE-2P-110	32A-Yellow
63A-CEE	63A-CEE
125A-CEE	125A-CEE
Powerlock	Powerlock
Powerlock 120A	Powerlock 120A
Powerlock 400A	Powerlock 400A
Powerlock 660A	Powerlock 660A
Powerlock 800A	Powerlock 800A
Camlock	Camlock
NAC3FCA	Powercon Blue
NAC3FCB	Powercon Grey
PowerconTRUE1	Powercon TRUE1
powerCONTRUE1TOP	powerCON TRUE1 TOP
Socapex-16	Socapex-16
Socapex-7	Socapex-7
Socapex-9	Socapex-9
HAN-16	HAN-16
HAN-4	HAN-4
L6-20	L6-20
L15-30	L15-30
Stagepin	Stagepin
HUBBELL-6-4	HUBBELL 6-4
DIN 56905	Eberl

Annex E
(normative)

Wheel Slot Image Definition

Definition of images used in wheel slots to visualize gobos, animation wheels or color wheels.

Gobo images shall be in PNG format with an alpha channel. Indexed, Greyscale and Alpha, 8-bit RGB and Alpha, or 16-bit RGB and Alpha are accepted pixel formats.



Key
1 transparent Background
2 opaque region
3 colored region

Figure E.1 —Sample gobo image

A gobo image is comprised of a transparent Background (1) and the image itself on top. The Background shall be fully transparent and should be considered to be the equivalent of a gobo holder. The Image region shall be fully opaque aside from anti-aliasing and shall be as large as possible.

NOTE This allows a data consumer to determine the precise pixel extents of the Image and place the provided PNG over an arbitrary GUI background without additional processing.

The Background region, the equivalent of gobo holder, is defined by full transparency (Alpha 0). In the Image region, Pure Black (RGB 0,0,0) is opaque (2), and Pure White (RGB 255,255,255) is transparent (GDTF).

Colored gobos (3) shall use an RGB approximation. The RGB approximation shall be calculated on the basis of Pure White being the CCT of the fixture light source and the ICC color profile embedded within the PNG. (See ISO 15076-1:2010) The default shall be sRGB.

Bibliography

ANSI E1.11, *Entertainment Technology — USITT DMX512-A — Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories*

DIN EN ISO/CIE 11664-1, *Colorimetry — Part 1: CIE standard colorimetric observers (ISO/CIE 11664-1)*

ISO 15076-1:2010, *Image technology colour management — Architecture, profile format and data structure — Part 1: Based on ICC.1:2010*

ESTA Control Protocol Working Group — Manufacturer IDs, Online table (according to clauses E4 and D5 of ANSI E1.11): https://tsp.esta.org/tsp/working_groups/CP/mfctrIDs.php

W3C Recommendation — *Extensible Markup Language (XML) 1.0 (Fifth Edition)*[2]

GDTF Share, Website with additional information around GTDF: www.gdtf-share.com

2 Available at: https://www.w3.org/TR/2008/REC-xml-20081126/.