

ICS 35.240.99; 97.200.10

Ersatz für
DIN SPEC 15800:2020-07

**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 101 Seiten

Dieses Dokument wurde durch die im Vorwort genannten Verfasser erarbeitet und verabschiedet.

Contents

	Page
Foreword	5
Introduction.....	6
1 Scope	7
2 Normative references	7
3 Terms and definitions.....	7
4 File Format Definition.....	8
5 Fixture Type Node	11
6 Attribute Definitions	12
6.1 General	12
6.2 Activation Groups	13
6.2.1 General	13
6.2.2 Activation Group	13
6.3 Feature Groups	13
6.3.1 General	13
6.3.2 Feature Group	13
6.4 Attributes.....	14
6.4.1 General	14
6.4.2 Attribute.....	14
6.4.3 Subphysical Unit.....	15
7 Wheel Collect.....	15
7.1 General	15
7.2 Wheel	15
7.2.1 General	15
7.2.2 Wheel Slot.....	16
8 Physical Descriptions.....	18
8.1 General	18
8.2 Emitter Collect	18
8.2.1 General	18
8.2.2 Emitter.....	18
8.3 Filter Collect	19
8.3.1 General	19
8.3.2 Filter	19
8.4 Measurement	19
8.4.1 General	19
8.4.2 Measurement Point.....	20
8.5 Color Space Collect	20
8.5.1 General	20
8.5.2 Color Space.....	20
8.6 Gamut Collect	21
8.6.1 General	21
8.6.2 General	21
8.7 DMX Profile Collect	22
8.7.1 General	22
8.7.2 DMX Profile	22

8.7.3	Point	22
8.8	Color Rendering Index Collect	24
8.8.1	General	24
8.8.2	Color Rendering Index Group	24
8.9	Connector Collect	24
8.9.1	General	24
8.9.2	Connector	24
8.10	Properties Collect	25
8.10.1	General	25
8.10.2	OperatingTemperature	26
8.10.3	Weight	26
8.10.4	LegHeight	26
9	Model Collect	26
9.1	General	26
9.2	Model	27
9.3	Regarding glTF Files	31
9.4	Regarding SVG Files	31
10	Geometry Collect	31
10.1	General	31
10.2	General Geometry	32
10.3	Geometry Type Axis	33
10.4	Geometry Type Beam Filter	33
10.5	Geometry Type Color Filter	33
10.6	Geometry Type Gobo Filter	34
10.7	Geometry Type Shaper Filter	34
10.8	Geometry Type Beam	34
10.9	Geometry Type Media Server Layer	36
10.10	Geometry Type Media Server Camera	36
10.11	Geometry Type Media Server Master	37
10.12	Geometry Type Display	37
10.13	Geometry Type Laser	37
10.13.1	General	37
10.13.2	Protocol	38
10.14	Geometry Type Reference	38
10.14.1	General	38
10.14.2	Break	39
10.15	Geometry Type Wiring Object	39
10.15.1	General	39
10.15.2	Pin Patch	41
10.16	Geometry Type Inventory	41
10.17	Geometry Type Structure	41
10.18	Geometry Type Support	42
10.19	Geometry Type Magnet	43
11	DMX Mode Collect	43
11.1	General	43
11.2	DMX Mode	43
11.2.1	DMX Channel Collect	44
11.2.2	Relation Collect	48
11.2.3	Macro Collect	49
12	Revision Collect	50
12.1	General	50
12.2	Revision	50
13	Fixture Type Preset Collect	51

13.1	General	51
13.2	Fixture Type Preset.....	51
14	Supported Protocol Collect	51
14.1	General	51
14.2	RDM Section.....	51
14.2.1	SoftwareVersionID	52
14.3	Art-Net Section.....	52
14.3.1	General	52
14.3.2	Map.....	52
14.4	Streaming ACN Section	53
14.4.1	General	53
14.4.2	Map.....	53
14.5	Posi Stage Net Section	53
14.6	Open Sound Control Section	53
14.7	CITP Section.....	53
Annex A (normative) Attribute Definitions		54
Annex B (normative) Attribute Listing.....		65
Annex C (informative) Name Character Table		91
Annex D (informative) Predefined Connector Types		94
Annex E (normative) Wheel Slot Image Definition		97
Annex F (normative) SubPhysicalUnit precisions.....		98
F.1	Pulse	98
F.2	PulseClose.....	99
F.3	PulseOpen.....	99
Bibliography		101

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-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 SPECs are not part of the body of German Standards.

A draft of this DIN SPEC has not been published.

Despite great efforts to ensure the accuracy, reliability and precision of technical and non-technical information, the DIN SPEC-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 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. 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.

Amendments

This standard differs from DIN SPEC 15800:2020-07 as follows:

- a) alignment with the updated GDTF format reflecting the development of the format in the recent time;
- b) refined and restructured paragraphs, e.g. addition of new attributes;
- c) addition of Annex F.

Previous editions

DIN SPEC 15800: 2020-07

Introduction

In the lighting and entertainment industry, lighting fixtures (luminaires and other controllable devices) have become more and more complex, and the development of these devices happens faster than ever. New devices are designed with very involved structures; they have numerous complex color-mixing systems and mode dependencies. To give the user access to the enormous flexibility of existing devices, a method of providing accurate fixture type data, and controlling and pre-visualizing devices quickly and precisely as possible, is needed. The “General Device Type Format” (GDTF) is that method.

There are many different lighting consoles and software manufacturers on the market, and all of them use different file formats and methods of getting the fixture control information into their systems. The development of new high-end fixtures at an amazing rate creates a lack of available control data on the side of the console and pre-visualization software manufacturers. Also, fixture manufacturers are often approached directly by their clients with requests 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. A format description is needed that not only provides all of the required control information, but also structures it in a hierarchical way matching that of the described device.

The lighting designer who would like to use these devices has to deal with such obstacles. Designers often receive the device control data of a specific new fixture later than expected. Also, the data may be incomplete, because it was not created with the latest information from the fixture manufacturer.

These issues demonstrate that our industry needs a standardized way of defining the description of intelligent and complex devices. This document defines such 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 which will soon become necessary, are therefore already included in this DIN SPEC, but with a note that no specifications can be made now.

1 Scope

This document specifies the “General Device Type Format” (GDTF), 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 E1.54-2021, *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

General Device Type Format

GDTF

descriptive name of the specification

3.2

Fixture Type Attribute

singular mutually 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

element that's groups the Fixture Type Attributes into a structured way for easier access and search

3.5

Feature Groups

element that's groups the Fixture Type Features into a logical way for easier access and search

3.6

DMXBreak

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

3.7

glTF

file format for three-dimensional scenes and models

Note 1 to entry: glTF is the acronym for Graphics Language Transmission Format.

3.8

glb

selfcontained binary file format of a glTF scene or model.

4 File Format Definition

To describe the device type, a 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 subfolders for a better structural overview:
– ./models/3ds – ./models/gltf – ./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/gltf/base.glb

./models/gltf/yoke.glb

./models/3ds_low/base.3ds

```

./models/3ds_low/yoke.3ds
./models/gltf_low/base.glb
./models/gltf_low/yoke.glb
./models/gltf_high/base.glb
./models/gltf_high/yoke.glb
./models/3ds_high/base.3ds
./models/3ds_high/yoke.3ds
./models/svg/base.svg
./models/svg/yoke.svg
./models/svg_side/base.svg
./models/svg_side/yoke.svg
./models/svg_front/base.svg
./models/svg_front/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 1 “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

Value Type	Format	Description
Vector3	{float,float,float}	Vector with 3 float components
Matrix	{float,float,float,float} {float,float,float,float} {float,float,float,float} {float,float,float,float}	The transformation matrix consists 4×4 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}	Rotation matrix, consist 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
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 2 "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">
<GDTF DataVersion="1.2">
```

The example above shows the XML node for GDTF version 1.2.

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. As it is based on Name, it is safe for parsing.
ShortName	String	Shortened name of the fixture type. Non detailed version or an abbreviation. Can use any characters or symbols.
LongName	String	Detailed, complete name of the fixture type, can include any characters or extra symbols.
Manufacturer	String	Manufacturer of the fixture type.
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 × 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.
ThumbnailOffsetX	Int	Horizontal offset in pixels from the top left of the viewbox to the insertion point on a label.
ThumbnailOffsetY	Int	Vertical offset in pixels from the top left of the viewbox to the insertion point on a label.
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	Describes 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	Is used to specify 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

6.1 General

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

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

All currently defined Fixture Type Attributes are specified in Annex B “Attribute Listing”.

All currently defined activation groups are specified in Annex B “Attribute Listing”.

All currently defined feature groups are specified 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. Predefines fixtury type attributes are specified in Annex A.

6.2 Activation Groups

6.2.1 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 an activation group.

6.2.2 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.

6.3 Feature Groups

6.3.1 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 they must be grouped in a Feature PanTilt.

As children, the feature groups has a list of a feature group.

6.3.2 Feature Group

6.3.2.1 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.

6.3.2.2 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.

6.4 Attributes

6.4.1 General

This section defines the Fixture Type Attributes (XML node <Attributes>). As children, the attributes node has a list of attributes.

6.4.2 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.
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.

As children the attribute node has a list of a subphysical units.

6.4.3 Subphysical Unit

This section defines the Attribute Subphysical Unit (XML node `<SubPhysicalUnit>`). The currently defined XML attributes of the subphysical unit are specified in Table 10.

Table 10 — XML Attributes of the Subphysical Unit

XML Attribute Name	Value Type	Description
Type	Enum	The currently defined values are: "PlacementOffset", "Amplitude", "AmplitudeMin", "AmplitudeMax", "Duration", "DutyCycle", "TimeOffset", "MinimumOpening", "Value", "RatioHorizontal", "RatioVertical".
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".
PhysicalFrom	Float	The default physical from of the subphysical unit; Unit: as defined in PhysicalUnit; Default value: 0
PhysicalTo	Float	The default physical to of the subphysical unit; Unit: as defined in PhysicalUnit; Default value: 1

The subphysical unit does not have any children. For examples of applying SubPhysicalUnits refer to Annex F.

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 can have a list of a wheels.

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 11.

Table 11 — Wheel Attributes

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

As children, Wheel has 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 12.

Table 12 — 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: 1 024 × 1 024; — Recommended resolution of gobo: 256 × 256; — Recommended resolution of animation wheel: 256 × 256 These resource files are located in a folder called . /wheels in the zip archive. Default value: empty.

More information on the definitions of images used in wheel slots to visualize gobos, animation wheels or color wheels are specified 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 for the prism facet (XML node <Facet>). The currently defined XML attributes of the prism facet are specified in Table 13.

Table 13 — 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 14.

Table 14 — AnimationSystem Attributes

XML Attribute Name	Value Type	Description
P1	Array of Float	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 Float is X-axis and second is Y-axis.
P2	Array of Float	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 Float is X-axis and second is Y-axis.
P3	Array of Float	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 Float is X-axis and second is Y-axis.
Radius	Float	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.

An example of AnimationSystem is given in Figure 1.

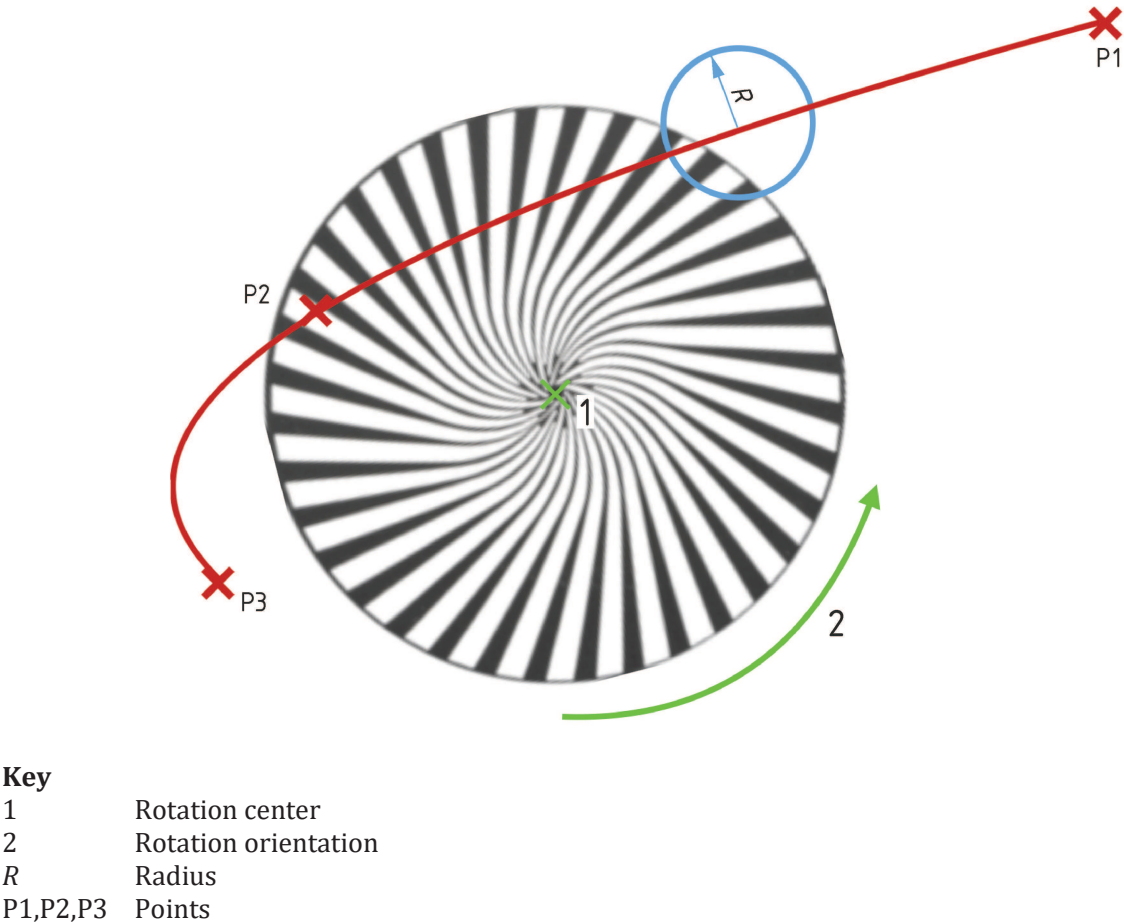


Figure 1 — Example of animation system

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 15.

Table 15 — Physical Description Children

XML node	Mandatory	Description
Emitters	No	Describes device emitters
Filters	No	Describes device filters
ColorSpace	No	Describes device color space
AdditionalColorSpaces	No	Describes additional device color spaces
Gamuts	No	Describes device gamuts
DMXProfiles	No	Describes nonlinear correlation between DMX input and physical output of a channel.
CRIs	No	Describes color rendering according to ANSI/IES TM-30-15 (99 color samples).
Connectors	No	Obsolete now. See Geometry Collect, WiringObject. 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 an 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 16.

Table 16 — 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 has a list of measurements.

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 filters.

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 17.

Table 17 — 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 has a list of a measurements.

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 18.

Table 18 — 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 measurement.
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 19.

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 19 — 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 Collect

8.5.1 General

This section defines color spaces. Currently it does not have any XML attributes (XML node `<AdditionalColorSpaces>`). As children, color space collect has a list of a ColorSpace.

8.5.2 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 20.

Table 20 — Color Space Attributes

XML Attribute Name	Value Type	Description
Name	Name	Unique Name of the Color Space. Default Value: "Default". Note that the name need to be unique for the default colorspace and all color spaces in the AdditionalColorSpaces node.
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 21.

Table 21 — 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 E1.54-2021
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 Gamut Collect

8.6.1 General

This section defines gamuts. Currently it does not have any XML attributes (XML node <Gamuts>). As children, gamut collect has a list of a Gamut.

8.6.2 General

This section defines the color gamut of the fixture (XML node <Gamut>), which is the set of attainable colors by the fixture. The currently defined XML attributes of a gamut are specified in Table 22.

Table 22 — Gamut Attributes

XML Attribute Name	Value Type	Description
Name	Name	Unique Name of the Gamut.
Points	Array of ColorCIE	Set of points defining the vertice of the gamut's polygon.

The gamut does not have any children.

8.7 DMX Profile Collect

8.7.1 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 profiles.

8.7.2 DMX Profile

This section defines the DMX profile description (XML node `<DMXProfile>`). The currently defined XML attributes of the DMX profile are specified in Table 23.

Table 23 — DMX Profile Attributes

XML Attribute Name	Value Type	Description
Name	Name	Unique name of the DMX profile

As children a DMX Profile has a list of point.

8.7.3 Point

This section contains points to define the DMX profile (XML node `<Point>`). The currently defined XML attributes of a point are specified in Table 24.

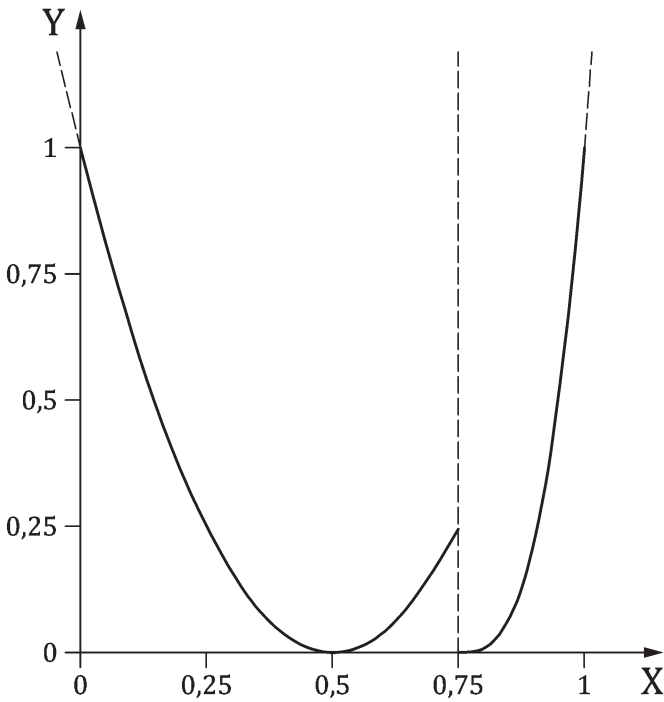
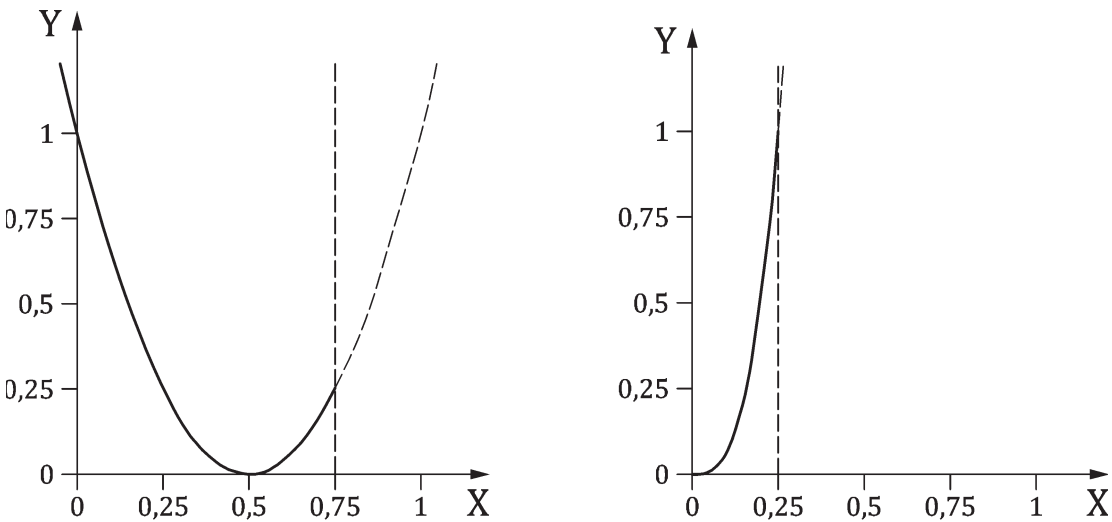
Table 24 — Point Attributes

XML Attribute Name	Value Type	Description
DMXPercentage	Float	DMX percentage of the point; Unit: Percentage; Default value: 0
CFC0	Float	Cubic Function Coefficient for x^0 ; Default value: 0
CFC1	Float	Cubic Function Coefficient for x ; Default value: 0
CFC2	Float	Cubic Function Coefficient for x^2 ; Default value: 0
CFC3	Float	Cubic Function Coefficient for x^3 ; Default value: 0

Find the Point with the biggest DMXPercentage below or equal x . If there is none, the output is expected to be 0.

$$\text{Output}(x) = \text{CFC3} \cdot (x - \text{DMXPercent})^3 + \text{CFC2} \cdot (x - \text{DMXPercent})^2 + \text{CFC1} \cdot (x - \text{DMXPercent}) + \text{CFC0}$$

Figure 2 is an example where the output follows a function f for 75 % of the DMX Range and another function g for the last 25 %. The Point attributes are given to illustrate how they are defined.



$f(x) = 4x^2 - 4x + 1$	$g(x) = 64x^3$
<Point>	<Point>
DMXPercentage = 0	DMXPercentage = 0,75
CFC3=0	CFC3=64
CFC2=4	CFC2=0
CFC1=-4	CFC1=0
CFC0=1	CFC02=0

Figure 2 — DMXProfile example

A Point does not have any children.

8.8 Color Rendering Index Collect

8.8.1 General

This section contains TM-30-15 Fidelity Index (Rf) for 99 color samples. Currently it does not have any XML attributes (XML node <CRIs>). As children, CRIs Collect has a list of CRI groups.

8.8.2 Color Rendering Index Group

8.8.2.1 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 25.

Table 25 — CRI Group Attributes

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

As children, the CRIGroup has an optional list of Color Rendering Index.

8.8.2.2 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 measurement point are specified in Table 26.

Table 26 — 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.

8.9 Connector Collect

8.9.1 General

This section defined the physical connectors and is kept for backwards compatibility. From this document or GDTF v1.2 onwards physical connectors shall be described as WiringObjects in the Geometry Collect.

It currently does not have any XML attributes (XML node <Connectors>). As children, the Connector Collect has a list of a connectors.

8.9.2 Connector

See Geometry Collect WiringObject. For easier transition find below the equivalent of the WiringObject.

This section defines the connector (XML node <Connector>). The currently defined XML attributes of the connector are specified in Table 27.

Table 27 — Connector Attributes

XML Attribute Name	Value Type	Description
Name	Name	Unique Name of the connector. Now: Geometry Type WiringObject, XML Attribute: Name
Type	Name	The type of the connector. Find a list of predefined types in Annex D. Now: Geometry Type WiringObject, XML Attribute: ConnectorType. 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. Obsolete now.
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. Obsolete now.
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" Obsolete now.

The connector does not have any children.

8.10 Properties Collect

8.10.1 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 28.

Table 28 — 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. Obsolete now.
LegHeight	0 or 1	Height of the legs.

8.10.2 OperatingTemperature

This section defines the ambient operating temperature range (XML node <OperatingTemperature>). The currently defined XML attributes of the OperatingTemperature are specified in Table 29.

Table 29 — 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.

8.10.3 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 30.

Table 30 — 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.

8.10.4 LegHeight

This section defines the height of the legs (XML node <LegHeight>). The currently defined XML attributes of the LegHeight are specified in Table 31.

Table 31 — LegHeight

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 32.

Table 32 — 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.
SVGOffsetX	Float	Optional. File name without extension and without subfolder containing description of the model.
SVGOffsetY	Float	Offset in Y from the 0,0 point to the desired insertion point of the top view svg. Unit based on the SVG. Default value: 0
SVGSideOffsetX	Float	Offset in X from the 0,0 point to the desired insertion point of the side view svg. Unit based on the SVG. Default value: 0
SVGSideOffsetY	Float	Offset in Y from the 0,0 point to the desired insertion point of the side view svg. Unit based on the SVG. Default value: 0
SVGFrontOffsetX	Float	Offset in X from the 0,0 point to the desired insertion point of the front view svg. Unit based on the SVG. Default value: 0
SVGFrontOffsetY	Float	Offset in Z from the 0,0 point to the desired insertion point of the front view svg. Unit based on the SVG. Default value: 0

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, svg). The path for 3ds files would be `./models/3ds`. For glb files, it would be `./models/gltf`.

Software that is utilizing GDTF files should always be able to read both 3ds and GLTF file formats and should be able to write at least one of these formats. It is preferable that only one type of 3D model file formats is used within one GDTF file.

The model currently does not have any children.

All models of a device combined should not exceed a maximum vertices count of 1 200 for the default mesh level of detail.

Three level of mesh level details are defined (see Table 33).

Table 33 — Mesh level of detail

LOD	Description	Folder 3DS / gltf
Low	Optional; This is the mesh for fixtures that are far away from the camera. It should have 30 % of the the vertexes from the default mesh vertex count.	3ds_low / gltf_low
Default	This is the default mesh that is used for real time visualization in preprogramming tool. It should have the minimum vertex count possible, while still looking like the fixture in 3D.	3ds / gltf
High	Optional; This is high quality mesh targeting non-realtime applications, where the vertex count is not that important. There is no limit for the vertex count.	3ds_high / gltf_high

Low and High meshes definitions are optional. Place the file with the same name in the defined folder.

The device shall be drawn in a hanging position displaying the front view. That results in 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 3.

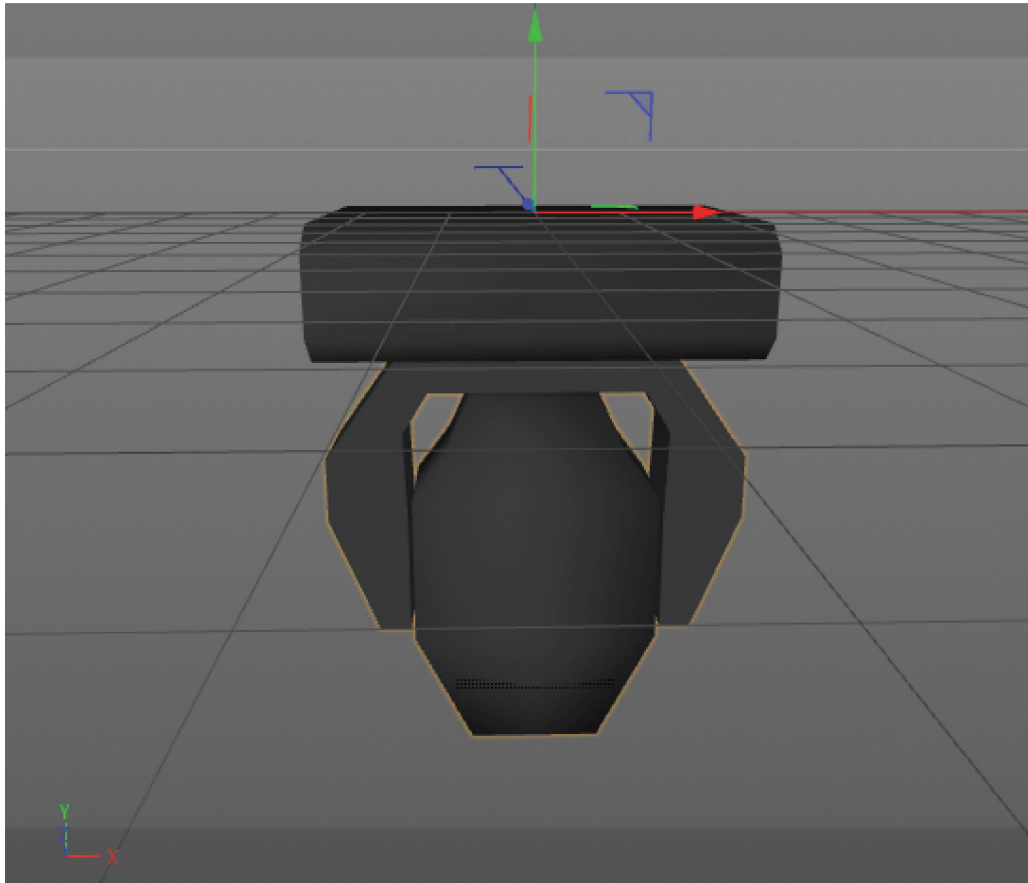


Figure 3 — Device in a hanging position — front view

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.

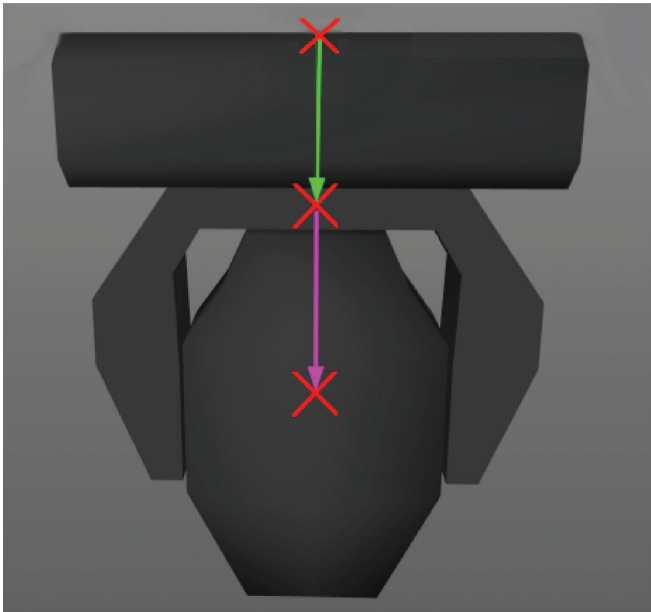


Figure 4 — Offsets of the parts

In Figure 4 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 36 to Table 56). It is important that the axis of rotation of each device part is exactly positioned (see Figure 5).

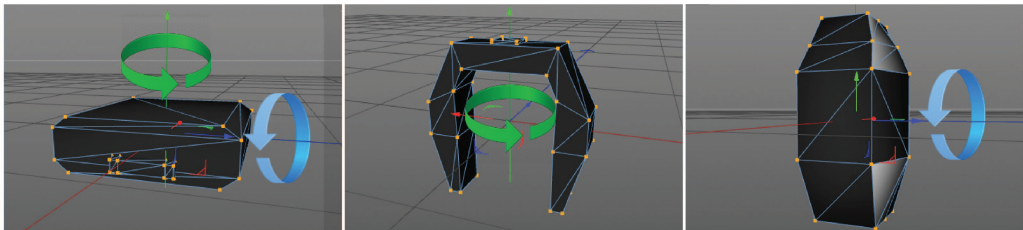


Figure 5 — Positions of rotation axis

The dimension XML attributes of model (see Table 32) are always used, no matter the scaling and ratio of the mesh. 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 (see Table 34).

SVG use viewboxes to align their content. The viewbox is always defined by the top left of the bounding box. With the attributes `SVG_VIEW_Offset_XY` you can define the insertion point in relation to the view box (see Figure 6).

SVG can have background filling. This filling should have the color `#C8C8C8`. By this color, any software can replace it with another color.

Table 34 — SVG views

Type	Description	Folder 3DS / gltf
Top View	View from top in -Z direction.	svg
Front View	View from front in Y direction	svg_front
Side View	View from front in -X direction	svg_side

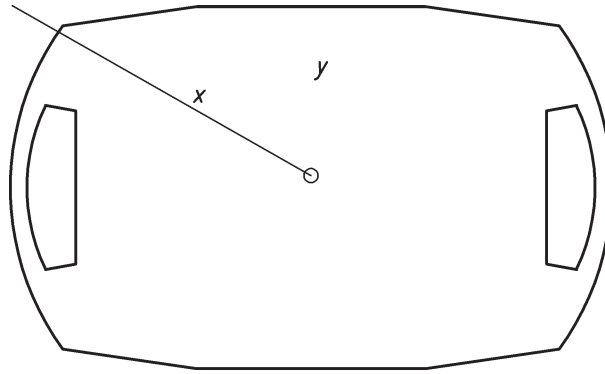


Figure 6 — Insertion point

SVG Images should be drawn in a 1 : 1 scale to the actual device. Use mm as base unit. Scaling operation from 3D meshes will not be applied to the SVG information.

9.3 Regarding glTF Files

The used glTF files should follow these requirements:

- Use the glb binary format;
- Only use the 2.0 version;
- Do not use extension;
- Do not use animations;
- Only use jpeg or png texture resource;
- all vertex attributes are `GL_FLOAT`.

9.4 Regarding SVG Files

The SVG files used should follow these requirements:

- SVG 1.1 spec;
- do not embed bitmap images;
- align the viewBox to the top left of the device.

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 35.