

DIN SPEC 15801



ICS 35.240.99; 97.200.10

Veranstaltungstechnik – My virtual rig (MVR); Text Englisch

Entertainment technology –
My virtual rig (MVR);
Text in English

Technologies du spectacle –
My virtual rig (MVR);
Texte en anglais

Gesamtumfang 63 Seiten

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

Contents

	Page
Foreword	6
Introduction	7
1 Scope	8
2 Normative references	8
3 Terms and definitions	8
4 MVR Definitions	9
4.1 General	9
4.2 File Format Definition	9
4.3 Generic Value Types	10
4.4 Node Definitions	11
4.5 GeneralSceneDescription Node	11
4.6 Node Definition: UserData	12
4.6.1 General	12
4.6.2 Node Definition: Data	12
4.7 Node Definition: Scene	12
4.8 Node Definition: AUXData	13
4.8.1 General	13
4.8.2 Node Definition: Symdef	13
4.8.3 Node Definition: Position	14
4.8.4 Node Definition: MappingDefinition	14
4.8.5 Node Definition: Class	15
4.8.6 Node Definition: Layers	15
4.9 Node Definition: ChildList	16
4.10 Node Definition for Parametric Objects	16
4.10.1 Node Definition: SceneObject	16
4.10.2 Node Definition: GroupObject	18
4.10.3 Node Definition: FocusPoint	18
4.10.4 Node Definition: Fixture	19
4.11 Node Definition: Truss	22
4.12 Node Definition: Support	24
4.13 Node Definition: VideoScreen	25
4.14 Node Definition: Projector	27
4.15 Other Node Definition	29
4.15.1 Node Definition: Matrix	29
4.15.2 Node Definition: Gobo	30
4.15.3 Node Definition: Sources	30
4.15.4 Node Definition: ScaleHandeling	31
4.16 Node Definition: Geometries	32
4.17 Node Definition: Symbol	32
4.18 Node Definition: Geometry3D	33
4.18.1 General	33
4.18.2 Supported 3D file formats	33
4.18.3 Node Definition: Projections	33
4.18.4 Node Definition: Addresses	34
4.18.5 Node Definition: Protocols	35
4.18.6 Node Definition: Protocol	36
4.18.7 Node Definition: Alignments	36
4.18.8 Node Definition: CustomCommands	37
4.18.9 Node Definition: Overwrites	37
4.18.10Node Definition: Connections	38
4.18.11Node Definition: Mappings	39

5	Communication Format Definition	40
5.1	General	40
5.2	TCP Mode of protocol	41
5.3	WebSocket Mode of protocol	42
5.4	Packet & Message definition	42
5.4.1	General	42
5.4.2	TCP Mode	42
5.4.3	WebSocket Mode	43
5.5	MVR_JOIN message	43
5.5.1	General	43
5.5.2	TCP Mode	43
5.5.3	WebSocket Mode	44
5.6	MVR_LEAVE message	48
5.7	MVR_COMMIT message	50
5.7.1	General	50
5.7.2	TCP Mode	50
5.7.3	WebSocket Mode	50
5.8	MVR_REQUEST message	53
5.9	MVR_NEW_SESSION_HOST message	58
5.9.1	General	58
5.9.2	Change Service URL/Name	58
5.10	Switch Mode of a Network	59
Annex A	(informative) Object ID for Selection purposes	61
Annex B	(informative) UUID purposes	62
Bibliography		63

Figures

Figure 1	ScaleHandeling Node Attributes	32
Figure 2	MVR-xchange mode of operation	41
Figure 3	TCP mode: MVR-xchange client joining MVR-xchange group	44
Figure 4	Websocket mode: MVR-xchange client joining MVR-xchange group	46
Figure 5	MVR_LEAVE message to quit MVR-xchange group	49
Figure 6	TCP mode: MVR-xchange client commits to MVR-xchange group.	50
Figure 7	Websocket mode: MVR-xchange client commits to MVR-xchange group.	51
Figure 8	Server makes the MVR_COMMIT itself, and only sends it to connected MVR-xchange clients	52
Figure 9	Websocket mode: MVR-xchange client requesting file	55
Figure 10	TCP mode: MVR-xchange client requesting file	57

Tables

Table 1	<i>XML Generic Value Types</i>	10
Table 2	<i>GeneralSceneDescription Node Attributes</i>	11
Table 3	<i>GeneralSceneDescription Node Children</i>	12
Table 4	<i>UserData Node Children</i>	12
Table 5	<i>Data Node Attributes</i>	12
Table 6	<i>Scene Node Children</i>	12
Table 7	<i>AUXData Node Children</i>	13
Table 8	<i>Symdef Node Attributes</i>	13
Table 9	<i>Symdef Node Children</i>	13
Table 10	<i>Symdef Childlist Node Children</i>	13
Table 11	<i>Position Node Attributes</i>	14

Table 12 — <i>MappingDefinition Node Attributes</i>	14
Table 13 — <i>MappingDefinition Node Children</i>	14
Table 14 — <i>Class Node Attributes</i>	15
Table 15 — <i>Layers Node Children</i>	15
Table 16 — <i>Layer Node Attributes</i>	15
Table 17 — <i>Layer Node Children</i>	16
Table 18 — <i>ChildList Node Children</i>	16
Table 19 — <i>SceneObject Node Attributes</i>	17
Table 20 — <i>SceneObject Node Children</i>	17
Table 21 — <i>GroupObject Node Attributes</i>	18
Table 22 — <i>GroupObject Node Children</i>	18
Table 23 — <i>FocusPoint Node Attributes</i>	19
Table 24 — <i>FocusPoint Node Children</i>	19
Table 25 — <i>Fixture Node Attributes</i>	19
Table 26 — <i>Fixture Node Children</i>	20
Table 27 — <i>Truss Node Attributes</i>	23
Table 28 — <i>Truss Node Children</i>	23
Table 29 — <i>Support Node Attributes</i>	24
Table 30 — <i>Support Node Children</i>	24
Table 31 — <i>VideoScreen Node Attributes</i>	26
Table 32 — <i>VideoScreen Node Children</i>	26
Table 33 — <i>Projector Node Attributes</i>	28
Table 34 — <i>Projector Node Children</i>	28
Table 35 — <i>Matrix Node Value Types</i>	30
Table 36 — <i>Gobo Node Attributes</i>	30
Table 37 — <i>Sources Node Children</i>	30
Table 38 — <i>Source Node Attributes</i>	31
Table 39 — <i>Source Node Value Types</i>	31
Table 40 — <i>ScaleHandeling Node Attributes</i>	31
Table 41 — <i>Geometries Node Children</i>	32
Table 42 — <i>Symbol Node Attributes</i>	32
Table 43 — <i>Symbol Node Children</i>	32
Table 44 — <i>Geometry3D Node Attributes</i>	33
Table 45 — <i>Geometry3D Node Children</i>	33
Table 46 — <i>Supported 3D file formats</i>	33
Table 47 — <i>Projections Node Children</i>	34
Table 48 — <i>Projection Node Children</i>	34
Table 49 — <i>Adresses Node Children</i>	34
Table 50 — <i>Address Node Attributes</i>	34
Table 51 — <i>Address Node Children</i>	35
Table 52 — <i>Network Node Attributes</i>	35
Table 53 — <i>Protocols Node Children</i>	35
Table 54 — <i>Protocol Node Attributes</i>	36
Table 55 — <i>Alignments Node Children</i>	36
Table 56 — <i>Alignment Node Attributes</i>	37
Table 57 — <i>CustomCommands Node Children</i>	37
Table 58 — <i>Overwrites Node Children</i>	37
Table 59 — <i>Overwrite Node Attributes</i>	38
Table 60 — <i>Connections Node Children</i>	38
Table 61 — <i>Connection Node Attributes</i>	39
Table 62 — <i>Mappings Node Children</i>	39
Table 63 — <i>Mapping Node Attributes</i>	39
Table 64 — <i>Mapping Node Children</i>	40
Table 65 — <i>TXT Record Attributes</i>	41
Table 66 — <i>Packet & Message Definitions</i>	42
Table 67 — <i>Data Type MVR-xchange package</i>	43
Table 68 — <i>MVR_JOIN message Attributes</i>	46

Table 69 — *MVR_JOIN response Attributes* 47

Table 70 — *MVR_LEAVE message Attributes* 49

Table 71 — *MVR_LEAVE response Attributes* 49

Table 72 — *MVR_COMMIT message Attributes* 52

Table 73 — *MVR_COMMIT response Attributes* 53

Table 74 — *MVR_REQUEST message Attributes* 57

Table 75 — *MVR_REQUEST error response Attributes* 58

Table 76 — *MVR_NEW_SESSION_HOST message Attributes* 59

Table 77 — *MVR_NEW_SESSION_HOST error response Attributes* 60

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 consortia 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
- MA Lighting Technology GmbH
Gerhard Krude, Daniel Müller, Rene Berhorst
- ROBE lighting s.r.o.
Josef Valchář, Petr Vanek
- Vectorworks Incorporated
Jeremy Powell, Biplap Sarkar
- Verband für Medien- und Veranstaltungstechnik (VPLT)
Randell Greenlee

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 DIN SPEC 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 document, please go to DIN's website (www.din.de) and search for the document number in question.

Introduction

MVR — My Virtual Rig — specified in this document will unify the information exchange between different applications within the entertainment industry. Based on GDTF, as specified in DIN SPEC 15800, MVR allows the exchange of scenic and environmental information and complete show setups as planning status. Furthermore, the MVR file format allows programs to share data and geometry for a scene. A scene is a set of parametric objects such as fixtures, trusses, video screens, and other objects that are used specifically in the entertainment industry.

Typical workflow:

- 1) Program A saves an MVR file containing a scene;
- 2) Program B imports this file;
- 3) Program B changes some parametric data in the scene;
- 4) Program B saves an MVR containing the scene;
- 5) Program A imports this file and applies the changes to the existing objects.

1 Scope

This document specifies “My Virtual Rig” (MVR), which is designed to provide a unified way of listing and describing the hierarchical and logical structure based on DIN SPEC 15800 “General Device Type Format” (GDTF) — and further environmental information of a show setup in the entertainment business. It will be used as a foundation for the exchange of extended device and environmental data between lighting consoles, CAD and 3D-pre-visualization applications. The purpose of an MVR-file is to reflect the real-world physical components of a show setup and the logical patch information of the devices while maintaining the kinematic chain.

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.

DIN SPEC 15800:2022-02, *Entertainment Technology — General Device Type Format (GDTF); Text in English*

ISO/IEC 21778:2017, *Information technology — The JSON data interchange syntax*

Extensible Markup Language (XML) 1.0¹

PKWARE 6.3.3²

RFC1035³, *Domain Names — Implementation and Specification*

RFC3629⁴, *UTF-8, a transformation format of ISO 10646*

RFC4122⁵, *A Universally Unique IDentifier (UUID) URN Namespace*

RFC6455⁶, *The WebSocket Protocol*

RFC6762⁷, *Multicast DNS*

RFC6763⁸, *DNS-Based Service Discovery*

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>

1 Available at <https://www.w3.org/TR/2008/REC-xml-20081126/>

2 Available at <https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT>

3 Available at <https://www.ietf.org/rfc/rfc1035.txt>

4 Available at <https://datatracker.ietf.org/doc/html/rfc3629>

5 Available at <https://www.rfc-editor.org/rfc/rfc4122>

6 Available at <https://www.ietf.org/rfc/rfc6455.txt>

7 Available at <https://www.ietf.org/rfc/rfc6762.txt>

8 Available at <https://www.ietf.org/rfc/rfc6763.txt>

— DKE-IEV: available at <https://www.dke.de/DKE-IEV>

3.1

My Virtual Rig

MVR

descriptive name of the specification

3.2

MVR-xchange

protocol to share MVR files over the network

3.3

MVR-xchange client

application that participates in the MVR-xchange

3.4

MVR-xchange group

group of MVR-xchange clients that work on the same project and communicate together

3.5

TCP Mode

MVR-xchange communication via TCP packages and discovery via mDNS

3.6

WebSocket Mode

MVR-xchange communication via WebSockets and discovery via DNS

4 MVR Definitions

4.1 General

MVR consists of two parts to enable any application to exchange GDTF but also general information in the same common format. Firstly the MVR file format as described in the following section. Secondly a MVR communication format to simplify exchange between applications.

4.2 File Format Definition

To describe all information within one file, a zip file according to PKWARE 6.3.3 with the extension *.mvr is used. The archive shall contain one root file named `GeneralSceneDescription.xml`, along with all other resource files referenced via this Root File. The root file `GeneralSceneDescription.xml` is mandatory inside the archive to be a valid MVR file.

UTF-8 according to RFC3629 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 node 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.

- The archive shall not use encryption or password protection.
- All files referenced by the Root File shall be placed at the root level. They shall not be placed in folders.
- Files shall be placed using either STORE (uncompressed) or DEFLATE compression. No other compression algorithms are supported.
- Files may be placed into the archive in any order.

- A `Universal.gdtt` GDTF template file with a `.gdtt` extension can be added to define Gobos, Emitters, Filter and such for referencing.
- Filenames within the archive shall not differ only by case. E.g. it is prohibited to have the files `GE01.glb` and `ge01.glb` within the same archive.
- The file name of the ZIP archive can be chosen freely.

All objects used have a persistent unique ID to track changes between the different applications. If there are no changes to the original GDTF file it is mandatory to keep it in the MVR during export. If there are changes to the GDTF file it is mandatory to add a revision to the GDTF file in order to reflect it.

Only user-intended modifications of any part of the MVR file shall be processed. This is particular important if applications in the workflow do not need or accept all data of the MVR file. Such behaviour guarantees that all later steps in the workflow have access to the original intended data.

EXAMPLE An example of a typical MVR archive is shown below:

```
GeneralSceneDescription.xml
Custom@Fixture1.gdttf
Custom@Fixture2.gdttf
ge01.3ds
ge01.glb
Textr12.png
Universal.gdtt
```

4.3 Generic Value Types

Table 1 contains a list of the available types for node or attribute values:

Table 1 — XML Generic Value Types

Value Type Name	Description
Integer	A signed or unsigned integer value represented in base 10. Uses a dash '-' (U+002D) as a prefix to denote negative numbers. E.g. 15 or -6
Float	A floating-point numeric value represented in #attrType-Bool base 10 decimal or scientific format. Uses full stop '.' (U+002E) to delimit the whole and decimal part and 'e' or 'E' to delimit mantissa and exponent. Implementations shall write sufficient decimal places to precisely round-trip their internal level of precision. Infinities and not-a-number (NaN) are not permitted. E.g. 1.5, 3.9265e+2
Bool	A Boolean value. When representing <code>true</code> indicate with <code>true</code> , when <code>false</code> indicate with <code>false</code> .
String	Any sequence of Unicode codepoints, encoded as necessary for XML. E.g. The following XML encodings (with their meaning in brackets): <code>&lt;</code> ; (<), <code>&amp;</code> ; (&), <code>&gt;</code> ; (>), <code>&quot;</code> ; ("), and <code>&apos;</code> ; (')
Enum	Possible values are predefined
UUID	A UUID to RFC4122 in text representation. The nil UUID (all zeros) is not permitted. Formatted as <code>xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx</code> . Used to link objects. Refer to Annex B for additional information regarding the use of UUID's.

Table 1 (continued)

Value Type Name	Description
Vector	Three Float values separated by ', ' defining a 3D vector's X, Y, and Z components. E.g. 1.0, 2.0, 3.0
FileName	The case-sensitive name of a file within the archive including the extension. The filename shall not contain any FAT32 or NTFS reserved characters. The extension is delimited from the base name by full stop '.' and the base name shall not be empty. It is recommended to limit filenames to the POSIX "Fully Portable Filenames" character set: [A-Z], [a-z], [0-9], the symbols '_' (U+005F), '-' (U+002D) and a maximum of one '.' (U+002E) E.g. My-Fixture_5.gdtf
CIE Color	CIE 1931 xyY absolute color point. Formatted as three Floats x, y, Y E.g. 0.314303, 0.328065, 87.699166
IPv4 Address	Common IPv4 Address in the format of dotted decimal notation. E.g. 192.168.1.10
IPv6 Address	Common IPv6 Address in the format of hexadecimal notation. E.g. 2001:0db8:85a3:0000:0000:8a2e:0370:7344

4.4 Node Definitions

The first XML node is always the XML description node: `<?xml version="1.0" encoding="UTF-8"?>`

4.5 GeneralSceneDescription Node

The second XML node is the mandatory GeneralSceneDescription node. The attributes of this node are listed in Table 2, the children of this node are given in Table 3.

Table 2 — GeneralSceneDescription Node Attributes

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
verMajor	Integer	Not Optional	Denotes the major version of the format used when saving this file.
verMinor	Integer	Not Optional	Denotes the minor version of the format used when saving this file.
provider	String	Not Optional	The name of the application that is generating the MVR export. This should stay the same between multiple versions.
providerVersion	String	Not Optional	The version of the software that is generating the MVR export. This should be different for each version that is available.

The current version of MVR reflected by this document is 1.6.

Table 3 — *GeneralSceneDescription Node Children*

Child Node	Allowed Count	Description
UserData	0 or 1	Specifies user data associated with this file.
Scene	1	Defines the scene described in this file.

4.6 Node Definition: UserData

4.6.1 General

This node contains a collection of user data nodes defined and used by provider applications if required. User data should not be expected to be preserved in the workflow of multiple applications importing and exporting the data. The defined *UserData Node Children* are specified in Table 4.

Node name: `UserData`

Table 4 — *UserData Node Children*

Child Node	Allowed Count	Description
Data (4.6.2)	0 or many	Defines a block of user data.

4.6.2 Node Definition: Data

This node contains a collection of data specified by the provider application. The defined *Data Node Attributes* are specified in Table 5.

Node name: `Data`

Table 5 — *Data Node Attributes*

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
provider	String	Not Optional	Specifies the name of the provider application that created this data.
ver	String	1	Version information of the data as specified by the provider application.

4.7 Node Definition: Scene

This node contains information about the scene. The defined *Scene Node Children* are specified in Table 6.

Node name: `Scene`

Table 6 — *Scene Node Children*

Child Node	Allowed Count	Description
AUXData (4.8)	0 or 1	Defines auxiliary data for the scene.
Layers (4.8.6)	1	A list of layers in the scene.

4.8 Node Definition: AUXData

4.8.1 General

This node contains auxiliary data for the scene node. The defined *AUXData Node Children* are specified in Table 7.

Node name: `AUXData`

Table 7 — *AUXData Node Children*

Child Node	Allowed Count	Description
Symdef (4.8.2)	0 or any	Graphical representation that will be instantiated in the scene.
Position (4.8.3)	0 or any	Defines a logical group of lighting devices.
MappingDefinition (4.8.4)	0 or any	Defines an input source for fixture color mapping applications.
Class (4.8.5)	0 or any	Defines a Class for object visibility filtering.

4.8.2 Node Definition: Symdef

This node contains the graphics so the scene can refer to this, thus optimizing repetition of the geometry. The child objects are located within a local coordinate system. The defined *Symdef Node Attributes* are specified in Table 8.

Node name: `Symdef`

Table 8 — *Symdef Node Attributes*

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the object.
name	String	Empty	The name of the object

The Symdef node (Table 9) contains the following children.

Table 9 — *Symdef Node Children*

Child Node	Allowed Count	Description
ChildList (4.9)	0 or 1	A list of graphic objects that are part of Symdef.

The child list (Table 10) contains a list of the following nodes:

Table 10 — *Symdef Childlist Node Children*

Child Node	Description
Geometry3D (4.18)	The geometry of this definition that will be instantiated in the scene.
Symbol (4.17)	The symbol instance that will provide a geometry of this definition.

4.8.3 Node Definition: Position

This node defines a logical grouping of lighting devices and trusses. The defined *Position Node Attributes* are specified in Table 11.

Node name: Position

Table 11 — Position Node Attributes

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the object.
name	String	Empty	The name of the object

4.8.4 Node Definition: MappingDefinition

This node specifies an input source for fixture color mapping applications. The defined *MappingDefinition Node Attributes* are specified in Table 12.

Node name: MappingDefinition

Table 12 — MappingDefinition Node Attributes

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the object.
name	String		The name of the source for the mapping.

The child list (Table 13) contains a list of the following nodes:

Table 13 — MappingDefinition Node Children

Child Node	Allowed Count	Value Type	Description
SizeX	1	Integer	The size in x direction in pixels of the source.
SizeY	1	Integer	The size in y direction in pixels of the source.
Source (4.15.3)	1		The video source that will be used for the Mapping.
ScaleHandeling (4.15.4)	0 or 1		How the source will be scaled to the mapping.

```
<MappingDefinition name="MappingStyle for View 1"
  uuid="BEF95EB8-98AC-4217-B10D-FB4B83381398">
  <SizeX>1920</SizeX>
  <SizeY>1080</SizeY>
  movie.mov
  <ScaleHandeling>ScaleKeepRatio</ScaleHandeling>
</MappingDefinition>
```

4.8.5 Node Definition: Class

This node defines a logical grouping across different layers. Primarily used for controlling object visibility of objects across multiple Layers. The defined *Class Node Attributes* are specified in Table 14.

Node name: `Class`

Table 14 — *Class Node Attributes*

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the class.
name	String		The name of the Class.

4.8.6 Node Definition: Layers

This node defines a list of layers inside the scene. The layer is a container of graphical objects defining a local coordinate system.

Node name: `Layers`

The child list (Table 15) contains a list of layer nodes:

Table 15 — *Layers Node Children*

Child Node	Description
Layer (4.8.6)	A layer representation.

Node Definition: Layer

This node defines a layer. The layer is a spatial representation of a geometric container. The child objects are located inside a local coordinate system. The defined *Layer Node Attributes* are specified in Table 16.

Node name: `Layer`

Table 16 — *Layer Node Attributes*

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the object.
name	String	Empty	The name of the object

The child list (Table 17) contains a list of layer nodes:

Table 17 — Layer Node Children

Child Node	Allowed Count	Description
Matrix (4.15.1)	0 or 1	The transformation matrix that defines the location and orientation of this the layer inside its global coordinate space. This effectively defines local coordinate space for the objects inside. The Matrix of the Layer is only allowed to have a vertical Transform (elevation). Rotation and scale shall be identity. Rotation and scale shall be identity, means no rotation and no scale.
ChildList (4.9)	0 or 1	A list of graphic objects that are part of the layer.

4.9 Node Definition: ChildList

This node defines a list of graphical objects.

Node name: `ChildList`

The child list (Table 18) contains a list of the following nodes:

Table 18 — ChildList Node Children

Child Node	Description
SceneObject (4.10.1)	A generic graphical object from the scene.
GroupObject (4.10.2)	A grouping object of other graphical objects inside local coordinate system.
FocusPoint (4.10.3)	A definition of a focus point.
Fixture (4.10.4)	A definition of a fixture.
Support (4.12)	A definition of a support.
Truss (4.11)	A definition of a truss.
VideoScreen (4.13)	A definition of a video screen.
Projector (4.14)	A definition of a projector.

4.10 Node Definition for Parametric Objects

4.10.1 Node Definition: SceneObject

This node defines a generic graphical object. The defined *SceneObject Node Attributes* are specified in Table 19.

Node name: `SceneObject`

Table 19 — *SceneObject Node Attributes*

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the object.
name	String	Empty	The name of the object
multipatch	UUID	Empty	The unique identifier of the parent multipatch fixture. When this value is set, you may not define a FixtureID or CustomId for this fixture. The FixtureID and CustomId from the object defined as multipatch parent also applies to this object.

The child list (Table 20) contains a list of one of the following nodes:

Table 20 — *SceneObject Node Children*

Child Node	Allowed Count	Value Type (see Table 1)	Description
Matrix (4.15.1)	0 or 1		The location and orientation of the object inside the parent coordinate system.
Classing	0 or 1	UUID	The Class the object belongs to.
Geometries	1		A list of geometrical representation objects that are part of the object.
GDTFSpec	0 or 1	FileName	The name of the file containing the GDTF information for this object, conforming to the DIN SPEC 15800.
GDTFMode	0 or 1	String	The name of the used DMX mode. This shall match the name of a DMXMode in the GDTF file. Mandatory when GDTFSpec is defined.
CastShadow	0 or 1	Bool	Defines if an object cast shadows.
Addresses	0 or 1		The container for DMX Addresses for this object.
Alignments	0 or 1		The container for Alignments for this object.
CustomCommands	0 or 1		The container for custom command for this object.
Overwrites	0 or 1		The container for overwrites for this object.
Connections	0 or 1		The container for connections for this object.
FixtureID	1	String	The Fixture ID is an identifier for the instance of this fixture that can be used to activate/select them for programming.
FixtureIDNumeric	1	Integer	The Fixture ID is an identifier for the instance of this fixture that can be used to activate/select them for programming.
UnitNumber	0 or 1	Integer	The identification of a fixture on its position. Use this as an alternative numbering scheme if the planning and programming numbering is different.

Table 20 (continued)

Child Node	Allowed Count	Value Type (see Table 1)	Description
CustomId	0 or 1	Integer	The CustomId is an identifier for the instance of this fixture within the CustomIdType that can be used to activate/select them for programming.
CustomIdType	0 or 1	Integer	Defines the CustomIdType this fixture belongs to. A CustomIdType defines to which group of objects this object belongs as an additional object identifier. The types for the CustomIdTypes are defined below.
ChildList	0 or 1		A list of graphic objects that are part of the layer.

4.10.2 Node Definition: GroupObject

This node defines logical group of objects. The child objects are located inside a local coordinate system. The defined *GroupObject Node Attributes* are specified in Table 21.

Node name: GroupObject

Table 21 — GroupObject Node Attributes

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the object.
name	String	Empty	The name of the object

The child list (Table 22) contains a list of one of the following nodes:

Table 22 — GroupObject Node Children

Child Node	Allowed Count	Value Type	Description
Matrix	0 or 1		The location and orientation of the object inside the parent coordinate system.
Classing	0 or 1	UUID	The Class the object belongs to.
ChildList	0 or 1		A list of graphic objects that are part of the group.

4.10.3 Node Definition: FocusPoint

This node defines a focus point object. The defined *FocusPoint Node Attributes* are specified in Table 23.

Node name: FocusPoint

Table 23 — FocusPoint Node Attributes

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the object.
name	String	Empty	The name of the object

The child list (Table 24) contains a list of one of the following nodes:

Table 24 — FocusPoint Node Children

Child Node	Allowed Count	Value Type	Description
Matrix	0 or 1		The location and orientation of the object inside the parent coordinate system.
Classing	0 or 1	UUID	The Class the object belongs to.
Geometries	1		A list of geometrical representation objects that are part of the object.

4.10.4 Node Definition: Fixture

This node defines an entertainment fixture object. The defined *Fixture Node Attributes* are specified in Table 25.

Node name: Fixture

Table 25 — Fixture Node Attributes

Attribute Name	Attribute Value Type (see Table 1)	Default Value when Optional	Description
uuid	UUID	Not Optional	The unique identifier of the object.
name	String	Empty	The Name is the value that represents the fixture object. It is not unique, and normally pairs with FID in Display.
multipatch	UUID	Empty	The unique identifier of the parent multipatch fixture. When this value is set, you may not define a FixtureID or CustomId for this fixture. The FixtureID and CustomId from the object defined as multipatch parent also applies to this object.