



**International  
Standard**

**ISO 14306-4**

**Industrial automation systems  
and integration — JT file format  
specification for 3D visualization —**

**Part 4:  
Version 3**

*Systèmes d'automatisation industrielle et intégration —  
Spécification de format de fichier JT pour visualisation 3D —*

*Partie 4: Version 3*

**First edition  
2026-04**

Sample Document

get full document from standards.iteh.ai

# Sample Document

get full document from [standards.iteh.ai](https://standards.iteh.ai)



## **COPYRIGHT PROTECTED DOCUMENT**

© ISO 2026

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office  
CP 401 • Ch. de Blandonnet 8  
CH-1214 Vernier, Geneva  
Phone: +41 22 749 01 11  
Email: [copyright@iso.org](mailto:copyright@iso.org)  
Website: [www.iso.org](http://www.iso.org)

Published in Switzerland

# Contents

	Page
<b>Foreword</b> .....	<b>vii</b>
<b>Introduction</b> .....	<b>ix</b>
<b>1 Scope</b> .....	<b>1</b>
<b>2 Normative references</b> .....	<b>1</b>
<b>3 Terms, definitions and abbreviated terms</b> .....	<b>2</b>
3.1 Terms and definitions.....	2
3.2 Abbreviated terms.....	2
<b>4 Notational conventions</b> .....	<b>3</b>
4.1 Diagrams and field descriptions.....	3
4.2 Data types.....	6
4.3 Empty field.....	8
<b>5 File format</b> .....	<b>9</b>
5.1 General.....	9
5.2 Object types.....	9
5.3 File structure.....	9
5.3.1 File header.....	10
5.3.2 TOC segment.....	11
5.3.3 Data segment.....	12
5.4 Data segments.....	17
<b>6 LSG segment</b> .....	<b>18</b>
6.1 Segment overview.....	18
6.2 Graph elements.....	18
6.3 Node elements.....	19
6.3.1 Base node element.....	19
6.3.2 Base node data.....	19
6.3.3 Partition node element.....	20
6.3.4 Group node element.....	22
6.3.5 Instance node element.....	23
6.3.6 Part node element.....	24
6.3.7 Metadata node element.....	25
6.3.8 LOD node element.....	25
6.3.9 Range LOD node element.....	26
6.3.10 Switch node element.....	27
6.3.11 Base shape node element.....	28
6.3.12 Vertex shape node element.....	31
6.3.13 Tri-strip set shape node element.....	32
6.3.14 Polyline set shape node element.....	32
6.3.15 Point set shape node element.....	33
6.3.16 Polygon set shape node element.....	34
6.3.17 NULL shape node element.....	34
6.3.18 Primitive set shape node element.....	34
6.4 LSG attribute elements.....	36
6.4.1 Material attribute element.....	38
6.4.2 Texture image attribute element.....	42
6.4.3 Draw style attribute element.....	54
6.4.4 Light set attribute element.....	56
6.4.5 Linestyle attribute element.....	57
6.4.6 Pointstyle attribute element.....	59
6.4.7 Geometric transform attribute element.....	60
6.4.8 Palette map attribute element.....	62
6.4.9 Infinite light attribute element.....	63
6.4.10 Point light attribute element.....	65
6.5 Property atom elements.....	68

6.5.1	Base property atom element.....	68
6.5.2	String property atom element.....	69
6.5.3	Integer property atom element.....	70
6.5.4	Floating point property atom element.....	70
6.5.5	JT object reference property atom element.....	71
6.5.6	Date property atom element.....	72
6.5.7	Late loaded property atom element.....	73
6.5.8	Vector4f Property atom element.....	74
6.6	Property table.....	74
<b>7</b>	<b>Shape LOD segment.....</b>	<b>76</b>
7.1	Shape LOD segment overview.....	76
7.1.1	Tri-strip set shape LOD element.....	76
7.1.2	Polyline set shape LOD element.....	77
7.1.3	Point set shape LOD element.....	77
7.1.4	Polygon set LOD element.....	78
7.1.5	Null shape LOD element.....	89
7.1.6	Primitive set shape element.....	89
<b>8</b>	<b>Geometry segments.....</b>	<b>96</b>
8.1	Geometry segments overview.....	96
8.2	STEP B-Rep segment.....	96
8.3	Wireframe segment.....	96
8.4	XT B-Rep element.....	97
<b>9</b>	<b>MetaData segment.....</b>	<b>97</b>
9.1	MetaData segment overview.....	97
9.2	Property proxy MetaData element.....	97
9.3	PMI manager MetaData element.....	100
9.3.1	PMI design group entities.....	101
9.3.2	PMI associations.....	103
9.3.3	PMI user attributes.....	105
9.3.4	PMI string table.....	106
9.3.5	PMI model views.....	107
9.3.6	Generic PMI entities.....	109
9.3.7	PMI CAD tag data.....	119
9.3.8	PMI polygon data.....	120
9.3.9	PMI model view sort orders.....	123
9.3.10	PMI association properties.....	123
9.3.11	Generic PMI additions.....	125
<b>10</b>	<b>Info segment.....</b>	<b>128</b>
<b>11</b>	<b>Data compression and encoding.....</b>	<b>129</b>
11.1	Data compression and encoding overview.....	129
11.2	Common compression data collection formats.....	130
11.2.1	Int32 compressed data packet.....	130
11.2.2	Int64 compressed data packet.....	135
11.2.3	Compressed vertex coordinate array.....	137
11.2.4	Compressed vertex normal array.....	138
11.2.5	Compressed vertex texture coordinate array.....	140
11.2.6	Compressed vertex colour array.....	141
11.2.7	Compressed vertex flag array.....	143
11.2.8	Compressed auxiliary fields array.....	143
11.2.9	Point quantizer data.....	147
11.2.10	Texture quantizer data.....	147
11.2.11	Colour quantizer data.....	148
11.2.12	Uniform quantizer data.....	149
11.2.13	Compressed entity list for non-trivial knot vector.....	149
11.2.14	Compressed control point weights data.....	153
11.2.15	Compressed curve data.....	153

## ISO 14306-4:2026(en)

11.2.16	Compressed CAD Tag Data .....	156
11.3	Encoding algorithms .....	158
11.3.1	Uniform data quantization .....	158
11.3.2	Bitlength CODEC .....	158
11.3.3	Arithmetic CODEC .....	159
11.3.4	Deering normal CODEC .....	164
11.4	LZMA compression .....	166
<b>12</b>	<b>Conformance requirements</b> .....	<b>166</b>
12.1	General conformance .....	166
12.2	Conformance classes .....	166
12.2.1	Approximated geometry visualization (CC01) .....	167
12.2.2	Precise geometry visualization (CC02) .....	167
<b>13</b>	<b>Common data conventions and constructs</b> .....	<b>167</b>
13.1	Overview .....	167
13.2	Late-loading data .....	167
13.3	TOC segment location .....	167
13.4	Bit fields .....	167
13.5	Empty field .....	168
13.6	Hash value .....	168
13.7	Scene graph construction .....	168
13.8	Metadata conventions .....	169
13.8.1	Property key naming conventions .....	169
13.8.2	PMI properties .....	170
13.8.3	CAD properties .....	171
13.8.4	Tessellation properties .....	172
13.8.5	Miscellaneous properties .....	173
13.8.6	The SUBNODE property and reference sets .....	174
13.8.7	Material and surface designation properties .....	178
13.9	LSG attribute accumulation semantics .....	178
13.10	LSG part structure .....	179
13.11	Range LOD node alternative rep selection .....	179
13.12	B-Rep face group associations .....	180
13.13	Smart topology table (STT) segment .....	180
13.14	Watermark image .....	181
13.15	State flags .....	181
<b>Annex A (normative) Information object registration</b> .....		<b>182</b>
<b>Annex B (informative) Object type identifiers</b> .....		<b>183</b>
<b>Annex C (informative) Coding algorithms – An implementation</b> .....		<b>185</b>
<b>Annex D (informative) Hashing – An Implementation</b> .....		<b>208</b>
<b>Annex E (informative) Polygon mesh topology coder</b> .....		<b>211</b>
<b>Annex F (informative) Per face group attributes</b> .....		<b>229</b>
<b>Annex G (normative) STEP B-Rep</b> .....		<b>233</b>
<b>Annex H (normative) STEP schema</b> .....		<b>236</b>
<b>Annex I (informative) XT B-Rep data segment</b> .....		<b>303</b>
<b>Annex J (informative) Smart topology table (STT) segment</b> .....		<b>405</b>
<b>Annex K (normative) Wireframe segment</b> .....		<b>431</b>
<b>Annex L (informative) Mapping table 14306-4 XT B-Rep to STEP</b> .....		<b>434</b>
<b>Annex M (informative) Procedural geometry – Evaluation and approximation</b> .....		<b>440</b>
<b>Annex N (normative) Information Object Registration Scheme</b> .....		<b>475</b>
<b>Annex O (informative) Change history</b> .....		<b>477</b>

# Sample Document

get full document from [standards.iteh.ai](https://standards.iteh.ai)

## Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see [www.iso.org/directives](http://www.iso.org/directives)).

ISO draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at [www.iso.org/patents](http://www.iso.org/patents). ISO shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see [www.iso.org/iso/foreword.html](http://www.iso.org/iso/foreword.html).

This document was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 4, *Industrial data*.

This first edition of ISO 14306-4, together with ISO 14306-1, ISO 14306-2, ISO 14306-3, cancels and replaces the second edition ISO 14306:2017, which has been technically revised.

The main changes are as follows:

- this document defines the requirements based upon the JT 10.5 file format requirements, whereas ISO 14306:2017 defines the requirements based upon the JT 9.5 file format requirements. A detailed list of the changes are shown in [Annex O](#);
- the version string in the File Header has been updated to reflect ISO 14306-4 Version 3;
- the file compression format has changed from Z-Lib to the publicly available LZMA format;
- the technical description for MbStrings Composite Data Types has been updated to include UTF 16;
- new segment types have been added for Smart Topology Table (STT) and Info Segment;
- the Base Attribute Data description has been expanded to include the logical collection Base Attribute Data Fields V2;
- a new attribute element has been added to the LSG definition, Palette Map Attribute Element;
- a new attribute element, “Sabot Attribute Element”, has been added to the LSG definition. Sabot is used to insulate earlier readers from attributes with non-fallback palette Index attributes in order to preserve forward compatibility;
- new PMI Association values have been included to support partial scene sectioning;

## ISO 14306-4:2026(en)

- a complete description of all PMI String Property Atom Element values is made available as a Reference Data Library in electronic format;
- generic PMI Entity Type values have a new value; 0x0309 Weld Note Type;
- an [Annex F](#) (Per Face Group Attributes) has been added;
- two new PMI Associations Reason Code values have been added, code 256 and code 257;
- [Annex I](#) is related to the XT BREP geometry definition. The following changes have been made in I.1.1.1.5

Updates in the “Topology” description.

Updates in the “World Topologies” table.

Updates to the “Body fields” table.

Updates to the “Region fields” table.

A list of all parts in the ISO 14306 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at [www.iso.org/members.html](http://www.iso.org/members.html).

# Sample Document

get full document from [standards.iteh.ai](https://standards.iteh.ai)

## Introduction

The ISO 14306 format is an industry focused, high-performance, lightweight, flexible file format for capturing and repurposing 3D product definition data for visualization to enable collaboration and validation throughout the extended enterprise. The ISO 14306 format is streamable and contains compression for compact and efficient representation.

Some of the highlights of the ISO 14306 format include:

- built-in support for assemblies, sub-assemblies and part constructs;
- flexible partitioning scheme, supporting single or multiple files;
- b-rep solid shape representations to provide precision to the light-weight viewing processes;
- product manufacturing information in support of paperless manufacturing initiatives;
- precise and imprecise wireframe shape representations;
- discrete purpose-built levels of detail;
- triangle sets, polygon sets, point sets, line sets and implicit primitive sets (such as cylinder, cone and sphere);
- full array of visual attributes such as for materials, textures, lights;
- hierarchical bounding box and bounding spheres;
- data compression that allows producers of ISO 14306 files to fine tune the trade-off between compression ratio and fidelity of the data.

Beyond the data contents description of the ISO 14306 format, the overall physical structure/organization of the format is also designed to support operations such as:

- offline optimizations of the data contents, therefore file granularity and flexibility optimized to meet the needs of enterprise data translation solutions;
- asynchronous streaming of content, therefore viewing optimizations;
- occlusion culling and fixed-framerate display modes;
- layers, and layer filters.

NOTE This document is based on the JT Open version 10.5 specification, which defines a logical scene graph with geometry specific node and attribute support, extended to support the ISO 10303 series.

[Annex A](#) contains an identifier that conforms to ISO/IEC 8824-1. The identifier unambiguously identifies this document in an open information system.

[Annex A](#) contains an identifier that conforms to ISO/IEC 8824-1. The identifier unambiguously identifies this document and the schema in an open information system. The ISO 14306 series applies the method as described in [Annex N](#).

# Sample Document

get full document from [standards.iteh.ai](https://standards.iteh.ai)

# Industrial automation systems and integration — JT file format specification for 3D visualization —

## Part 4: Version 3

### 1 Scope

This document defines the syntax and semantics of a file format for the 3D visualization and interrogation of lightweight geometry and product manufacturing information derived from CAD systems, using visualization software tools that do not need the full capability of a CAD system.

This document has been adopted as a 3D visualization capability in addition to the ISO 10303 series.

The ISO 10303 series are the ISO standards adopted for the engineering data exchange, sharing and long-term archiving of product definition information throughout the product lifecycle.

In this document 3D visualization is defined as the visual presentation on a screen or another media of graphical and textual 3-dimensional representations of a set of data representing an object, information or results of a computational process in order to enable decision process by a human looking at the data visualized in a medium.

The ISO 14306 file format specification for 3D visualization includes data descriptions that can represent the following data:

- facet information (triangles), stored with geometry compression techniques;
- visual attributes such as lights, textures and materials;
- product manufacturing information (PMI);
- boundary representation (b-rep) solid model shape representation and associated metadata;
- configuration representations; and
- delivery methods such as asynchronous streaming of content.

The file format specification for 3D visualization does not specify the implementation of, or definition of a run-time architecture for viewing and/or processing ISO 14306 data.

### 2 Normative references

ISO 10303-21:2016, *Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure*

ISO 10303-42:2024, *Industrial automation systems and integration — Product data representation and exchange — Part 42: Integrated generic resource: Geometric and topological representation*

ISO 14306-2:2024, *Industrial automation systems and integration — JT file format specification for 3D visualization — Part 2: Vocabulary*

### 3 Terms, definitions and abbreviated terms

#### 3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 14306-2 apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

- ISO Online browsing platform: available at <https://www.iso.org/obp>
- IEC Electropedia: available at <https://www.electropedia.org/>

#### 3.2 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

Abs	absolute value
Bbox	bounding box
B-Rep	boundary representation
CAD	computer aided design
CODEC	coder-decoder
GD&T	geometric dimensioning and tolerancing
GUID	globally unique identifier
HSV	hue, saturation, value
LCS	local coordinate system
LsbFirst	least significant byte first
LZMA	Lempel–Ziv–Markov
Max	maximum
MCS	machine coordinate system
Min	minimum
MsbFirst	most significant byte first
N/A	not applicable
PCS	parameter coordinate space
PLM	product lifecycle management
RGB	red, green, blue
RGBA	red, green, blue, alpha
TOC	table of contents
VPCS	viewpoint coordinate system

URL uniform resource locator



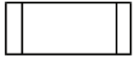


WCS world coordinate system

## 4 Notational conventions

### 4.1 Diagrams and field descriptions

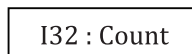
Symbolic diagrams are used to describe the structure of the ISO 14306 file. The symbols used in these diagrams are shown in [Table 1](#) and have the following meaning:

**Table 1 — Symbols**

Symbol	Description
	Rectangles represent a data field of one of the standard data types.
	Folders represent a logical collection of one or more of the standard data types. This information is grouped for clarity and the basic data types that compose the group are detailed in following sections of the document.
	Rectangles with extra lines at left and the right sides corners clipped off represent information logical steps that has been compressed.
	Rectangles with the right side corners clipped off represent information that has been compressed.
	Arrows convey the ordering of the information.

The format used to title the diagram symbols is dependent upon the symbol type as follows:

Diagram “rectangle box” (therefore, standard data types) symbols are titled using a format of “Data\_Type : Field\_Name.” The Data\_Type is an abbreviated data type symbol as defined in [3.2](#) data types. In the example shown in [Figure 1](#) the Data\_Type is “I32” (a signed 32 bit integer) and Field\_Name is “Count.”



**Figure 1 — Rectangle box diagram**

Diagram “folder” (therefore, logical data collections) symbols are simply titled with a collection name. In the example shown in [Figure 2](#) the collection name is “Graph Elements.”



**Figure 2 — Folder diagram**

Diagram “rectangle box with lines at left and right sides” are simply titled with a logic step name. In the example shown in [Figure 3](#) the logic step name is “Recover First Shell Indices”.



**Figure 3 — Rectangle box with lines at left and right sides diagram**

Diagram “rectangle box with clipped right side corners” (therefore, compressed/encoded data fields) are titled using one of the following three formats:

Data type; followed by open brace “{”, number of bits used to store value, closed brace “}”, and a colon “:”; followed by the field name. This format for titling the diagram symbol indicates that the data is compressed but not encoded. The compression is achieved by using only a portion of the total bit range of the data type to store the value (for example, if a count value can never be larger than the value “63” then only 6 bits are needed to store all possible count values). In the example shown in [Figure 4](#) the data type is “U32”, “6” bits are used to store the value, and field name is “Count”

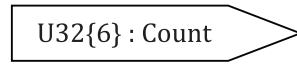


Figure 4 — Rectangle box with clipped right-side corners

Data type followed by open brace “{”, compressed data packet type, “,”, predictor type, closed brace “}”, and a colon “:”; followed by the field name. This format for titling the diagram indicates that a vector of “Data Type” data (therefore, primal values) is ran through “Predictor Type” algorithm and the resulting output array of residual values is then compressed and encoded into a series of symbols using one of the two supported compressed data packet types.

The two supported compressed data packet types are:

Int32CDP – the Int32CDP (therefore, Int32 compressed data packet) represents a third-generation format used to encode/compress a collection of data into a series of Int32 based symbols. This version of the Int32CDP supersedes the two similarly-named ones from the ISO 14306:2017 and should not be confused with either of its predecessors. A complete description for Int32 compressed data packet can be found in [11.2.2](#).

The Int32 compressed data packet type is used for compressing/encoding both “integer” and “float” (through quantization) data.

In the example shown in [Figure 5](#) the data type is “VecU32”, Int32 compressed data packet type is used, Lag1 predictor type is used, and field name is “First Shell Index.”



Figure 5 — Compressed data packet diagram

As mentioned above (with predictor-type algorithm), the primal input data values are not always what is encoded/compressed. This is because the primal input data is first run through a predictor-type algorithm, which produces an output array of residual values (therefore, the difference from the predicted value), and this resulting output array of residual values is the data which is actually encoded/compressed. This document format supports several predictor-type algorithms and each use of Int32CDP specifies, using the notation format described above, what predictor-type algorithm is being used on the data. The supported predictor type algorithms are as shown in [Table 2](#) (a sample implementation of decoding the predictor residual values back into the primal values can be found in [Annex C](#)).

Table 2 — Predictor type

Predictor type	Description
Lag1	Predicts as last value
Xor1	Predicts as last, but uses XOR instead of subtract to compute residual
NULL	No prediction is applied

Each predictor type can be combined with additional processing steps, and in such case the predictor type is prefixed with “Combined:”. For example, “Combined:Lag1” means that predictor type “Lag1” is combined with additional preprocessing steps. Additional description about the processing steps is provided whenever such combined predictor is used.

The format “Data Type : Field Name” used for titling the diagram symbol indicates that the data is both compressed and encoded. The Data\_Type is an abbreviated data type symbol as defined in data types and usually represents a vector/array of data. How the data is compressed and encoded into the data type is indicated by a CODEC type and other information stored before the particular data in the file. In the example shown in [Figure 6](#) the Data\_Type is “VecU32” and Field\_Name is “CodeText.”

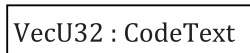


Figure 6 — Data\_Type: field name diagram

For some file segment types there is LZMA compression also applied to all bytes of element data stored in the segment. This LZMA compression applied to all the segment’s data is not indicated in the diagrams through the use of “rectangle box with clipped right side corners”. Instead, one shall examine information stored with the first element in the file segment to determine if LZMA compression is applied to all data in the segment. A complete description of the format data compression and encoding can be found in [5.4](#) and [Clause 11](#).

Each data collection diagram includes detailed descriptions for each entry in the data diagram.

For rectangles this detail includes the abbreviated data type symbol, field name, verbal data description, and compression technique/algorithm where appropriate. If the data field is documented as a collection of flags, then the field is to be treated as a bit mask where the bit mask is formed by combining the flags using the binary OR operator. Each bits usage is documented, and bit ON indicates flag value is TRUE and bit OFF indicates flag value is FALSE. All bits fields that are not defined as in use shall be set to “0”.

For folders (therefore, data collections), if the collection is not detailed under a sub-section of the particular document section referencing the data collection, then a comment is included following the diagram indicating where in the document the particular data collection is detailed.

If an arrow appears with a branch in its shaft, then there are two or more options for data to be stored in the file. Which data is stored will depend on information previously read from the file. The following example, shown in [Figure 7](#), shows data field A followed by (depending on value of A) either data field B, C, or D.

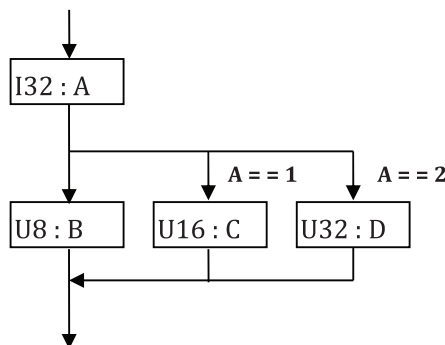


Figure 7 — Data field dependency example

In cases where the same data type repeats, a loop construct is used where the number of iterations appears next to the loop line. There are two forms of this loop construct. The first form is used when the number of iterations is not controlled by some previous read count value. Instead the number of iterations is either a hard-coded count (for example, always 80 characters) or is indicated by some end-of-list marker in the data itself (thus the count is always minimum of 1). This first form of the loop construct looks as shown in [Figure 8](#):