PUBLICLY AVAILABLE SPECIFICATION

ISO/PAS 14306

First edition
2011-12-15

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

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

Reference number
ISO/PAS 14306:2011(E)

Contents

Page

Tables

Figures

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.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of technical committees is to prepare International Standards. Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

In other circumstances, particularly when there is an urgent market requirement for such documents, a technical committee may decide to publish other types of document:

— an ISO Publicly Available Specification (ISO/PAS) represents an agreement between technical experts in an ISO working group and is accepted for publication if it is approved by more than 50 % of the members of the parent committee casting a vote;

— an ISO Technical Specification (ISO/TS) represents an agreement between the members of a technical committee and is accepted for publication if it is approved by 2/3 of the members of the committee casting a vote.

An ISO/PAS or ISO/TS is reviewed after three years in order to decide whether it will be confirmed for a further three years, revised to become an International Standard, or withdrawn. If the ISO/PAS or ISO/TS is confirmed, it is reviewed again after a further three years, at which time it must either be transformed into an International Standard or be withdrawn.

ISO/PAS 14306 was prepared by Technical Committee ISO/TC 184, *Automation systems and integration*, Subcommittee SC 4, *Industrial data*.

ISO/PAS 14306 is based on Siemens JT File Format Reference Version 8.1 Rev-C.

Introduction

This Publicly Available Specification was transposed by an ad hoc committee focused on industrial requirements for 3D product data visualization under the ISO/TC 184/SC 4 Harvesting Process, as defined in SC4 Standing Document (SC4N1198), Procedures for Transposing Externally Developed Specifications into ISO Deliverables.

The ad hoc committee was formed by members of ISO/TC 184/SC 4 in response to requests from the global industrial community for information on visualization formats. The group assessed several 3D visualization formats including COLLADA, JT, U3D and X3D against a list of 36 requirements. The final results concluded that these candidate formats are complementary to the ISO 10303 "STEP" series of standards concerning visualization data exchange. These formats are not intended for use for CAx data exchange or product data exchange.

The JT file format presented in this Publicly Available Specification is intended to provide data that can be used for further engineering activities in a PLM domain. The other formats were found to support product documentation (U3D) and visualisation data exchange based on XML (COLLADA, X3D) in a similar domain.

The International Organization for Standardization (ISO) draws attention to the fact that it is claimed that compliance with this document may involve the use of patent USA 20110199382.

ISO takes no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO that he is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO. Information may be obtained from:

Siemens PLM Software Product Development
345 Woodcliff Drive
Fairport
NY 14450
USA

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

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

1 Scope

This Publicly Available Specification defines the syntax and semantics of the JT Version 8.1 file format.

The JT format is an industry focused, high-performance, lightweight, flexible file format for capturing and repurposing 3D Product Definition data that enables collaboration, validation and visualization throughout the extended enterprise. JT format is the de-facto standard 3D Visualization format in the automotive industry, and the single most dominant 3D visualization format in Aerospace, Heavy Equipment and other mechanical CAD domains.

The JT format is both robust, and streamable, and contains best-in-class compression for compact and efficient representation. The JT format was designed to be easily integrated into enterprise translation solutions, producing a single set of 3D digital assets that support a full range of downstream processes from lightweight web-based viewing to full product digital mockups.

At its core the JT format is a scene graph with CAD specific node and attributes support. Facet information (triangles), is stored with sophisticated geometry compression techniques. Visual attributes such as lights, textures, materials and shaders (Cg and OGLSL) are supported. Product and Manufacturing Information (PMI), Precise Part definitions (B-Rep) and Metadata as well as a variety of representation configurations are supported by the format. The JT format is also structured to enable support for various delivery methods including asynchronous streaming of content.

Some of the highlights of the JT format include:

- Built-in support for assemblies, sub-assemblies and part constructs
- Flexible partitioning scheme, supporting single or multiple files
- B-Rep, including integrated support for industry standard Parasolid® (XT) format
- Product Manufacturing Information in support of paperless manufacturing initiatives
- Precise and imprecise wireframe
- Discrete purpose-built Levels of Detail
- Wire harness information
- Triangle sets, Polygon sets, Point sets, Line sets and Implicit Primitive sets (cylinder, cone, sphere, etc…)
- Full array of visual attributes: Materials, Textures, Lights, Shaders
- Hierarchical Bounding Box and Bounding Spheres
- Advanced data compression that allows producers of JT files to fine tune the trade off between compression ratio and fidelity of the data.

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

- Offline optimizations of the data contents
- File granularity and flexibility optimized to meet the needs of Enterprise Data Translation Solutions
- Asynchronous streaming of content
- Viewing optimizations such as view frustum and occlusion culling and fixed-framerate display modes.
- Layers, and Layer Filters.

Along with the pure syntactical definition of the JT Format, there is also series of conventions which although not required to have a reference compliant JT file, have become commonplace within JT format translators. These conventions have been documented in the “Best Practices” section of this JT format reference.

This JT format reference does not specifically address implementation of, nor define, a run-time architecture for viewing and/or processing JT data. This is because although the JT format is closely aligned with a run-time data representation for fast and efficient loading/unloading of data, no interaction behavior is defined within the format itself, either in the form of specific viewer controls, viewport information, animation behavior or other event-based interactivity. This exclusion of interaction behavior from the JT format makes the format more easily reusable for dissimilar application interoperation and also facilitates incremental update, without losing downstream authored data, as the original CAD asset revises.

2 References and Additional Information

[1] *JT Open Program* (http://www.jtopen.com) --- A program to help members leverage the benefits of open collaboration across the extended enterprise through the adoption of the JT format, a technology that makes it possible to view and share product information throughout the product lifecycle. Membership in the JT Open Program provides access to the JT Open Toolkit library, which among other things, provides read and write access to JT data and enforces certain JT conventions to ensure data compatibility with other JT-enabled applications.

[2] *JT2Go download* (http://www.jt2go.com) --- JT2Go is the no-charge 3D JT viewer from Siemens. JT2Go puts 3D data at your fingertips by allowing anyone to download the no-charge viewer. JT2Go also allows anyone to embed 3D JT data directly into Microsoft Office documents. JT2Go offers full 3D interactivity on parts, assemblies, and even 2D drawings (CGM & TIF).

[3] *Siemens: PLM Components: Parasolid: XT Pipeline* (http://www.ugs.com/products/open/parasolid/pipeline.shtml) --- This web page provides information on the Parasolid precise boundary representation format (XT) and how this XT format fits within the Siemens vision of seamless exchange of digital product models across enterprises, between different disciplines, using their PLM applications of choice.

[4] *OpenGL Programming Guide : the official guide to learning OpenGL Version 2*, Fifth Edition, by OpenGL Architecture Review Board, Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis (Addison-Wesley 2005) --- This book gives in-depth explanation of the OpenGL Specification and will provide further insight into the significance of some of the data (e.g. Materials, Textures) that can exist in a JT file. Information in this book may also serve as a guide for how one could process the data contained in a JT file to produce/render an image on the screen.

[5] Michael Deering, *Geometry Compression*, Computer Graphics, Proceedings SIGGRAPH '95, August 1995, pp. 13-20.

[6] Michael Deering, Craig Gotsman, Stefan Gumhold, Jarek Rossignac, and Gabriel Taubin, *3D Geometry Compression*, Course Notes for SIGGRAPH 2000, July 25, 2000.

[7] *OpenGL Shading Language Specification* (http://www.opengl.org/documentation/glsl/) --- OpenGL Shading Language (GLSL) as defined by the OpenGL Architectural Review Board, the governing body of OpenGL.

[8] *Cg Toolkit Users Manual* (http://developer.nvidia.com/object/cg_users_manual.html) --- Explains everything you need to learn and use the Cg language as well as the Cg runtime library.

[9] *The Cg Tutorial: The Definitive Guide to Programmable Real-Time Graphics*, Randima Fernando and Mark J. Kilgard, nVIDIA Corporation, Addison Wesley Publishing Company, April 2003

[10] K. Weiler. *Topological Structures for Geometric Modeling*, PhD thesis, Rensselaer Polytechnic Institute, Troy, NY, 1986.

[11] C. M. Hoffmann. *Geometric and Solid Modeling: An Introduction*. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1989.

[12] *Planetmath.org - Huffman Coding* (http://planetmath.org/encyclopedia/HuffmanCoding.html) --- This web page provides a technical overview of Huffman coding which is one form of data encoding used within the JT format.

[13] Michael Schindler, *Practical Huffman Coding* (http://www.compressconsult.com/huffman/#encoding) --- This web page provides some coding hints for implementing Huffman coding which is one form of data encoding used within the JT format.

[14] Glen G. Langdon Jr., *An Introduction to Arithmetic Coding*, IBM Journal of Research and Development, Volume 28, Number 2, March 1984, pp. 135-149.

[15] Paul G. Howard and Jeffrey Scott Vitter, *Practical Implementation of Arithmetic Coding*. *Image and Text Compression*, ed. J. A. Storer, Kluwer Academic Publishers, April 1992, pp. 85-112.

[16] zlib.net (http://www.zlib.net/) --- This web page provides (either directly or through links) complete detailed information on ZLIB compression including frequently asked questions, technical documentation, source code downloads, etc.

3 Definitions

3.1 Terms

It is assumed that readers of this document are familiar with concepts in the area of computer graphics and solid modeling. The intention of this section is not to provide comprehensive definitions, but is to provide a short introduction and clarification of the usage of terms within this document.

Assembly – A related collection of *model* parts, represented in a JT format logical scene graph as a logical graph branch

Attribute – Objects associated with nodes in a *logical scene graph* and specifying one of several appearances, positioning, or rendering characteristics of a *shape*

Boundary Representation – A solid model representation where the solid volume is specified by its surface boundary (both its geometric and topological boundaries).

CodeText – A collection of data in encoded form.

Directed Acyclic Graph – A *graph* is a set of nodes, and a set of edges connecting the nodes in a tree like structure. A *directed graph* is one in which every edge has a direction such that edge (u,v), connecting node-u with node-v, is different from edge (v,u). A *Directed Acyclic Graph* is a directed graph with no cycles; where a cycle is a path (sequence of edges) from a node to itself. So with a *Directed Acyclic Graph* there is no path that can be followed within the graph such that the first node in the path is the same as the last node in the path.

JT Enabled Application – Application which supports reading and/or writing reference compliant JT Format files.

Level of Detail – One alternative graphical representation for some *model* component (e.g. part).

Logical Scene Graph – A *scene graph* representing the logical organization of a *model*. Contains *shapes* and *attributes* representing the *model's* physical components, *properties* identifying arbitrary metadata (e.g. names, semantic roles) of those components, and a hierarchical structure expressing the component relationships.