INTERNATIONAL STANDARD

ISO/IEC 14496-22

Fourth edition
2019-01

Information technology — Coding of audio-visual objects —

Part 22: Open Font Format

Technologies de l'information — Codage des objets audiovisuels —
Partie 22: Format de police de caractères ouvert

Reference number
ISO/IEC 14496-22:2019(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

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 document 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).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see http://patents.iec.ch).

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.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This fourth edition cancels and replaces the third edition (ISO/IEC 14496-22:2015), which has been technically revised. It also incorporates the Amendments ISO/IEC 14496-22:2015/Amd.1:2017 and ISO/IEC 14496-22:2015/Amd.2:2017.

The main changes compared to the previous edition are as follows:

— new technology clauses were added;

— many existing clauses, subclauses, tables, figures and annexes were editorially revised.

A list of all parts in the ISO/IEC 14496 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.

Introduction

Multimedia applications require a broad range of media-related standards. In addition to the typical audio and video applications, multimedia presentations include scalable 2D graphics and text supporting all languages of the world. Faithful reproduction of scalable multimedia content requires additional components including scalable font technology. The Open Font Format, which is based on the OpenType®[1] font format, was originally developed as an extension of the TrueType®[2] font format, adding support for PostScript®[3] Compact Font Format (CFF) font data. OFF fonts and the operating system services which support OFF fonts provide users with a simple way to install and use fonts, whether the fonts contain TrueType outlines or CFF (PostScript Type1) outlines.

The Open Font Format addresses the following goals:

— broader multi-platform support;

— excellent support for international character sets;

— excellent protection for font data;

— smaller file sizes to make font distribution more efficient;

— excellent support for advanced typographic control.

CFF data included in OFF fonts may be directly rasterized or converted to the TrueType outline format for rendering, depending on which rasterizers have been installed in the host operating system. But the user model is the same: OFF fonts just work. Users will not need to be aware of the type of outline data in OFF fonts. And font creators can use whichever outline format they feel provides the best set of features for their work, without worrying about limiting a font's usability.

OFF fonts can include the OFF Layout tables, which allow font creators to design broader international and high-end typographic fonts. The OFF Layout tables contain information on glyph substitution, glyph positioning, justification, and baseline positioning, enabling text-processing applications to improve text layout.

As with TrueType fonts, OFF fonts allow the handling of large glyph sets using Unicode encoding. Such encoding allows broad international support, as well as support for typographic glyph variants.

Additionally, OFF fonts may contain digital signatures, which allows operating systems and browsing applications to identify the source and integrity of font files, including embedded font files obtained in web documents, before using them. Also, font developers can encode embedding restrictions in OFF fonts which cannot be altered in a font signed by the developer.

The International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of patents.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights. The holders of these patent rights have assured ISO and IEC that they 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 holders of these patent rights is registered with ISO and IEC.

---

[1] OpenType® is the trademark of a product supplied by Microsoft. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of this product.

[2] TrueType® is the trademark of a product supplied by Apple Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of this product.

[3] PostScript® is the trademark of a product supplied by Adobe Systems Inc. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO or IEC of this product.

Information may be obtained from:

Apple Inc.
1 Infinite Loop MS 3-PAT
US-Cupertino, CA 95014-2084
Tel.: +1 408 974 9453
Email: iplaw@apple.com

Microsoft Corporation
Interoperability Group 3460 157th Avenue NE
US-Redmond, WA 98052
Tel.: +1 425 882 80 80

Monotype Imaging Inc.
500 Unicorn Park Drive
US-Woburn, MA 01801
Tel.: +1 781-970-6088
E-mail: vladimir.levantovsky@monotypeimaging.com

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 and IEC shall not be held responsible for identifying any or all such patent rights.

Information technology — Coding of audio-visual objects —

Part 22:
Open Font Format

1 Scope

This document specifies the Open Font Format (OFF) specification, including the TrueType and Compact Font Format (CFF) outline formats. Many references to both TrueType and PostScript exist throughout this document, as Open Font Format fonts combine the two technologies. The document defines data structures for variaous font tables, and provides the necessary details for developers to build a font rendering and taxt layout/shaping engines in compliance with this document.

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.

ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*

ISO/IEC 14496-18, *Information technology — Coding of audio-visual objects — Part 18: Font compression and streaming*

ISO/IEC 15948, *Information technology — Computer graphics and image processing — Portble Network Graphics: Functional specification* [4]

IEC 61966-2-1/Amd 1:2003: *Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB*.

TrueType Instruction Set, http://www.microsoft.com/typography/otspec/ttinst.htm

Unicode 11.0, http://www.unicode.org/versions/Unicode11.0.0/

Scalable Vector Graphics (SVG) 1.1 (2nd edition), W3C Recommendation, 16 August 2011 http://www.w3.org/TR/SVG11/

IETF BCP 47 specification, “Tags for Identifying Languages”. http://tools.ietf.org/html/bcp47

3 Terms, definitions and abbreviated terms

3.1 Terms and definitions

No terms and definitions are listed in this document.

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

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

[4] Also available as a W3C Recommendation (Reference [15]).

3.2 Abbreviated terms

ACF	average character face
ANSI	American National Standards Institute
ASCII	American Standard Code for Information Interchange
BMP	[Unicode] basic multilingual plane
BTBD	baseline to baseline distance
CFF	compact font format
CID	character identifier
CJK	Chinese Japanese Korean [characters, ideographs, fonts, etc.]
CJKV	Chinese Japanese Korean and Vietnamese
CV	control value
CVT	control value table
DLL	dynamically linked library
FDEF	function definition
GID	glyph ID
ICF	ideographic character face
IDEF	instruction definition
IETF	Internet Engineering Task Force
JIS	Japanese Industrial Standard
LTR	left to right
NLC	National Language Council of Japan
OFF	open font format
OMPL	OFF mirroring pairs list
OTF	OpenType font
PCL	printer control language
PPM, PPEM	pixels per em
RTL	right to left
TTC	TrueType collection
TTF	TrueType font
UCS	universal character set
UTF	Unicode transformation format
UVS	Unicode variation sequence
VM	virtual memory
W3C	world wide web consortium

4 The Open Font file format

4.1 Description

An Open Font file contains data, in table format, that comprises either a TrueType or a CFF outline font. Rasterizers use combinations of data from the tables contained in the font to render the TrueType or PostScript glyph outlines. Some of this supporting data is used no matter which outline format is used; some of the supporting data is specific to either TrueType or PostScript.

References to the Universally Coded Character Set and the Unicode standard are used throughout this document; the users of the OFF cannot meet the requirements of this document without strict adherence to these standards.

4.2 Filenames

OFF font files may have the extension .OTF, .TTF, .OTC or .TTC. The extensions .OTC and .TTC should only be used for font collection files. For additional information on filename extension conventions, see subclause 8.4.

4.3 Data types

The following data types are used in the OFF font file. All OFF fonts use big-endian (network byte order):

Data Type	Description
uint8	8-bit unsigned integer.
int8	8-bit signed integer.
uint16	16-bit unsigned integer.
int16	16-bit signed integer.
uint24	24-bit unsigned integer.
uint32	32-bit unsigned integer.
int32	32-bit signed integer.
Fixed	32-bit signed fixed-point number (16.16)
FWORD	int16 that describes a quantity in font design units.
UFWORD	uint16 that describes a quantity in font design units.
F2DOT14	16-bit signed fixed number with the low 14 bits of fraction (2.14).
LONGDATETIME	Date represented in number of seconds since 12:00 midnight, January 1, 1904. The value is represented as a signed 64-bit integer.
Tag	Array of four uint8s (length = 32 bits) used to identify a table, design-variation axis, script, language system, feature, or baseline
Offset16	Short offset to a table, same as uint16, NULL offset = 0x0000
Offset32	Long offset to a table, same as uint32, NULL offset = 0x00000000

The F2DOT14 format consists of a signed, 2's complement integer and an unsigned fraction. To compute the actual value, take the integer and add the fraction. Examples of 2.14 values are:

Decimal Value	Hex Value	Integer	Fraction
1.999939	0x7fff	1	16383/16384
1.75	0x7000	1	12288/16384
0.000061	0x0001	0	1/16384
0.0	0x0000	0	0/16384
-0.000061	0xffff	-1	16383/16384
-2.0	0x8000	-2	0/16384

A Tag value is a uint8 array. Each byte within the array shall have a value in the range 0x20 to 0x7E. This corresponds to the range of values of Unicode Basic Latin characters in UTF-8 encoding, which is the same as the printable ASCII characters. As a result, a Tag value can be re-interpreted as a four-character sequence, which is conventionally how they are referred to. Formally, however, the value is a byte array.

When re-interpreted as characters, the Tag value is case sensitive. It shall have one to four non-space characters, padded with trailing spaces (byte value 0x20). A space character cannot be followed by a non-space character.

4.4 Table version numbers

Most tables have version numbers, and the version number for the entire font is contained in the Table Directory. Note that there are five different table version number types, each with its own numbering scheme.

- A single uint16 field. This is used in a number of tables, usually with versions starting at zero (0).
- Separate, uint16 major and minor version fields. This is used in a number of tables, usually with versions starting at 1.0.
- A Fixed field for major/minor version numbers. This is used in the maxp, post and vhea tables.
- A uint32 field with enumerated values.
- A uint32 field with a numeric value. This is used only in the DSIG and meta tables.

Only certain tables use a Fixed value for version, and only for reasons of backward compatibility. Fixed values will not be used in the future for any new tables that may be introduced. When a Fixed number is used as a version, the upper 16 bits comprise a major version number and the lower 16 bits a minor version. The representation of a non-zero minor version, however, is not consistent with the normal treatment of Fixed values, in which the lower 16 bits represent a fractional value, N * 2 ^ -16. Rather, tables with non-zero minor version numbers always specify the literal value of the version number. For example, the version number of 'maxp' table version 0.5 is 0x00005000, and that of 'vhea' table version 1.1 is 0x00011000. When Fixed is indicated as the type for a version field, the possible values should be treated as an enumeration of specific values, rather than as a numeric value capable of representing many potential major and minor versions.

The Table Directory uses a uint32 field with an enumeration of defined values that represent four-character tags; see subclause 4.5 (Top-level OFF organization) for details.

Implementations reading tables must include code to check version numbers so that if and when the format and therefore the version number changes, older implementations will handle newer versions gracefully.

Minor version number changes always imply format changes that are compatible extensions. If an implementation understands a major version number, then it can safely proceed reading the table. If the minor version is greater than the latest version recognized by the implementation, then the extension fields will be undetectable to the implementation.

For purposes of compatibility, version numbers that are represented using a single uint16 or uint32 value are treated like a minor version number, and any format changes are compatible extensions.

Note that some field values that were undefined or reserved in an earlier revision may be assigned meanings in a minor version change. Implementations should never make assumptions regarding reserved or unassigned values or bits in bit fields, and can ignore them if encountered. When writing font data, tools should always write zero for reserved fields or bits. Validators should warn of any non-zero values for fields or bits that are not defined for the given version against which data is being validated.

If the major version is not recognized, the implementation must not read the table as it can make no assumptions regarding interpretation of the binary data. The implementation should treat the table as missing.

4.5 Top-level OFF organization

4.5.1 Offset table

A key characteristic of the OFF format is the TrueType sfnt "wrapper", which provides organization for a collection of tables in a general and extensible manner.

The OFF font starts with the Offset Table. If the font file contains only one font, the Offset Table will begin at byte 0 of the file. If the font file is a font collection, the beginning point of the Offset Table for each font is indicated in the TTCHeader.

Offset Table		
Type	**Name**	**Description**
uint32	sfntVersion	0x00010000 or 0x4F54544F ('OTTO') – see below.
uint16	numTables	Number of tables.
uint16	searchRange	(Maximum power of 2 <= numTables) x 16.
uint16	entrySelector	Log2(maximum power of 2 <= numTables).
uint16	rangeShift	NumTables x 16 - searchRange.

OFF fonts that contain TrueType outlines should use the value of 0x00010000 for sfntVersion. OFF fonts containing CFF data (version 1 or 2) should use 0x4F54544F ('OTTO', when re-interpreted as a Tag) for sfntVersion.

4.5.2 Table directory

The Offset Table is followed immediately by the Table Record entries. Entries in the Table Record shall be sorted in ascending order by tag. Offset values in the Table Record are measured from the start of the font file.

Table Record		
Type	**Name**	**Description**
Tag	tableTag	Table identifier.
uint32	checksum	CheckSum for this table.
Offset32	Offset	Offset from beginning of TrueType font file.
uint32	Length	Length of this table.

