

DISPATCH Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 1, 2016

A. Johnston
Unaffiliated
B. Aboba
Microsoft
A. Hutton
Unify
L. Liess
Deutsche Telekom
T. Stach
Unaffiliated
February 29, 2016

An Opportunistic Approach for Secure Real-time Transport Protocol
(OSRTP)
draft-johnston-dispatch-osrtp-02

Abstract

Opportunistic Secure Real-time Transport Protocol (OSRTP) allows encrypted media to be used in environments where support for encryption is not known in advance, and not required. OSRTP is an implementation of Opportunistic Security, as defined in RFC 7435. OSRTP does not require advanced SDP extensions or features and is fully backwards compatible with existing secure and insecure implementations. OSRTP is not specific to any key management technique for SRTP. OSRTP is a transitional approach useful for migrating existing deployments of real-time communications to a fully encrypted and authenticated state.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 1, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
 1.1. Applicability Statement 3
 2. Requirements Language 3
 3. Definition of Opportunistic Security for SRTP 3
 4. Security Considerations 4
 5. Implementation Status 5
 6. Acknowledgements 5
 7. References 5
 7.1. Normative References 5
 7.2. Informative References 7
 Authors' Addresses 7

1. Introduction

Opportunistic Security [RFC7435] (OS) is an approach to security that defines a third mode for security between "cleartext" and "comprehensive protection" that allows encryption and authentication to be used if supported but will not result in failures if it is not supported. In terms of secure media, cleartext is RTP [RFC3550] media which is negotiated with the AVP (Audio Video Profile) profile defined [RFC3551]. Comprehensive protection is Secure RTP [RFC3711], negotiated with a secure profile, such as SAVP or SAVPF [RFC5124]. OSRTP allows SRTP to be negotiated with the AVP profile, with fallback to RTP if SRTP is not supported.

There have been some extensions to SDP to allow profiles to be negotiated such as SDP Capabilities Negotiation (capneg) [RFC5939] . However, these approaches are complex and have very limited deployment in communication systems. Other key management protocols for SRTP have been developed which by design use OS, such as ZRTP [RFC6189]. This approach for OSRTP is based on

[I-D.kaplan-mmusic-best-effort-srtp] where it was called "best effort SRTP". [I-D.kaplan-mmusic-best-effort-srtp] has a full discussion of the motivation and requirements for opportunistic secure media.

OSRTP uses the presence of SRTP keying-related attributes in an SDP offer to indicate support for opportunistic secure media. The presence of SRTP keying-related attributes in the SDP answer indicates that the other party also supports OSRTP and encrypted and authenticated media will be used. OSRTP requires no additional extensions to SDP or new attributes and is defined independently of the key agreement mechanism used. OSRTP is only usable when media is negotiated using the Offer/Answer protocol [RFC3264].

1.1. Applicability Statement

OSRTP is a transitional approach that provides a migration path from unencrypted communication (RTP) to fully encrypted communication (SRTP). It is only to be used in existing deployments which are attempting to transition to fully secure communications. New applications and new deployments will not use OSRTP.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Definition of Opportunistic Security for SRTP

To indicate support for OSRTP in an SDP offer, the offerer uses the AVP profile [RFC3551] but includes SRTP keying attributes. OSRTP is not specific to any key management technique for SRTP. For example:

If the offerer supports DTLS-SRTP key agreement [RFC5763], then an a=fingerprint attribute will be present, or

If the offerer supports SDP Security Descriptions key agreement [RFC4568], then an a=crypto attribute will be present, or

If the offerer supports ZRTP key agreement [RFC6189], then an a=zrtp-hash attribute will be present.

To accept OSRTP, an answerer receiving an offer indicating support for OSRTP generates an SDP answer containing SRTP keying attributes which match one of the keying methods in the offer. The answer MUST NOT contain attributes from more than one keying method, even if the offer contained multiple keying method attributes. The selected SRTP

key management approach is followed and SRTP media is used for this session. If the SRTP key management fails for any reason, the media session MUST fail. To decline OSRTP, the answerer generates an SDP answer omitting SRTP keying attributes, and the media session proceeds with RTP with no encryption or authentication used.

If the offerer of OSRTP receives an SDP answer which does not contain SRTP keying attributes, then the media session proceeds with RTP. If the SDP answer contains SRTP keying attributes, then that particular SRTP key management approach is followed and SRTP media is used for this session. If the SRTP key management fails, the media session MUST fail.

It is important to note that OSRTP makes no changes, and has no effect on media sessions in which the offer contains a secure profile of RTP, such as SAVP or SAVPF. As discussed in [RFC7435], this is the "comprehensive protection" for media mode.

4. Security Considerations

The security considerations of [RFC7435] apply to OSRTP, as well as the security considerations of the particular SRTP key agreement approach used. However, the authentication requirements of a particular SRTP key agreement approach are relaxed when that key agreement is used with OSRTP. For example:

For DTLS-SRTP key agreement [RFC5763], an authenticated signaling channel does not need to be used with OSRTP if it is not available.

For SDP Security Descriptions key agreement [RFC4568], an authenticated signaling channel does not need to be used with OSRTP if it is not available, although an encrypted signaling channel must still be used.

For ZRTP key agreement [RFC6189], the security considerations are unchanged, since ZRTP does not rely on the security of the signaling channel.

As discussed in [RFC7435], OSRTP is used in cases where support for encryption by the other party is not known in advance, and not required. For cases where it is known that the other party supports SRTP or SRTP needs to be used, OSRTP MUST NOT be used. Instead, a secure profile of RTP is used in the offer.

5. Implementation Status

Note to RFC Editor: Please remove this entire section prior to publication, including the reference to [RFC6982].

This section records the status of known implementations of the protocol defined by this specification at the time of posting of this Internet-Draft, and is based on a proposal described in [RFC6982]. The description of implementations in this section is intended to assist the IETF in its decision processes in progressing drafts to RFCs. Please note that the listing of any individual implementation here does not imply endorsement by the IETF. Furthermore, no effort has been spent to verify the information presented here that was supplied by IETF contributors. This is not intended as, and must not be construed to be, a catalog of available implementations or their features. Readers are advised to note that other implementations may exist.

According to [RFC6982], "this will allow reviewers and working groups to assign due consideration to documents that have the benefit of running code, which may serve as evidence of valuable experimentation and feedback that have made the implemented protocols more mature. It is up to the individual working groups to use this information as they see fit".

There are implementations of [I-D.kaplan-mmusic-best-effort-srtp] in deployed products by Microsoft and Unify. The IMTC "Best Practices for SIP Security" document [IMTC-SIP] recommends this approach. The SIP Forum plans to include support in the SIPconnect 2.0 SIP trunking recommendation [SIPCONNECT] which is under development. There are many deployments of ZRTP [RFC6189].

6. Acknowledgements

This document is dedicated to our friend and colleague Francois Audet who is greatly missed in our community. His work on improving security in SIP and RTP provided the foundation for this work.

Thanks to Eric Rescorla, Martin Thomson, and Richard Barnes for their comments.

7. References

7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC3264] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", RFC 3264, DOI 10.17487/RFC3264, June 2002, <<http://www.rfc-editor.org/info/rfc3264>>.
- [RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, DOI 10.17487/RFC3550, July 2003, <<http://www.rfc-editor.org/info/rfc3550>>.
- [RFC3551] Schulzrinne, H. and S. Casner, "RTP Profile for Audio and Video Conferences with Minimal Control", STD 65, RFC 3551, DOI 10.17487/RFC3551, July 2003, <<http://www.rfc-editor.org/info/rfc3551>>.
- [RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, DOI 10.17487/RFC3711, March 2004, <<http://www.rfc-editor.org/info/rfc3711>>.
- [RFC4568] Andreasen, F., Baugher, M., and D. Wing, "Session Description Protocol (SDP) Security Descriptions for Media Streams", RFC 4568, DOI 10.17487/RFC4568, July 2006, <<http://www.rfc-editor.org/info/rfc4568>>.
- [RFC5124] Ott, J. and E. Carrara, "Extended Secure RTP Profile for Real-time Transport Control Protocol (RTCP)-Based Feedback (RTP/SAVPF)", RFC 5124, DOI 10.17487/RFC5124, February 2008, <<http://www.rfc-editor.org/info/rfc5124>>.
- [RFC5763] Fischl, J., Tschofenig, H., and E. Rescorla, "Framework for Establishing a Secure Real-time Transport Protocol (SRTP) Security Context Using Datagram Transport Layer Security (DTLS)", RFC 5763, DOI 10.17487/RFC5763, May 2010, <<http://www.rfc-editor.org/info/rfc5763>>.
- [RFC6189] Zimmermann, P., Johnston, A., Ed., and J. Callas, "ZRTP: Media Path Key Agreement for Unicast Secure RTP", RFC 6189, DOI 10.17487/RFC6189, April 2011, <<http://www.rfc-editor.org/info/rfc6189>>.

- [RFC7435] Dukhovni, V., "Opportunistic Security: Some Protection Most of the Time", RFC 7435, DOI 10.17487/RFC7435, December 2014, <<http://www.rfc-editor.org/info/rfc7435>>.

7.2. Informative References

- [I-D.kaplan-mmusic-best-effort-srtp]
Audet, F. and H. Kaplan, "Session Description Protocol (SDP) Offer/Answer Negotiation For Best-Effort Secure Real-Time Transport Protocol", draft-kaplan-mmusic-best-effort-srtp-01 (work in progress), October 2006.
- [IMTC-SIP]
"Best Practices for SIP Security", IMTC SIP Parity Group <http://www.imtc.org/uc/sip-parity-activity-group/>, 2011, <<http://www.imtc.org>>.
- [RFC5939] Andreasen, F., "Session Description Protocol (SDP) Capability Negotiation", RFC 5939, DOI 10.17487/RFC5939, September 2010, <<http://www.rfc-editor.org/info/rfc5939>>.
- [RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, DOI 10.17487/RFC6982, July 2013, <<http://www.rfc-editor.org/info/rfc6982>>.
- [SIPCONNECT]
"SIP-PBX / Service Provider Interoperability SIPconnect 2.0 - DRAFT Technical Recommendation", SIP Forum <http://www.sipforum.org/content/view/179/213/>, 2015, <<http://www.sipforum.org>>.

Authors' Addresses

Alan Johnston
Unaffiliated
Bellevue, WA
USA

Email: alan.b.johnston@gmail.com

Bernard Aboba
Microsoft
One Microsoft Way
Redmond, WA 98052
USA

Email: bernard.aboba@gmail.com

Andy Hutton
Unify
Technology Drive
Nottingham NG9 1LA
UK

Email: andrew.hutton@unify.com

Laura Liess
Deutsche Telekom
Heinrich-Hertz-Strasse 3-7
Darmstadt 64295
Germany

Email: laura.liess.dt@googlemail.com

Thomas Stach
Unaffiliated

Email: thomass.stach@gmail.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 18, 2016

C. Lilley
W3C
October 16, 2015

The font Primary Content Type
draft-lilley-font-toplevel-00

Abstract

This memo serves to register and document the "font" Primary Content Type, under which the Internet Media subtypes for representation formats for fonts may be registered. This document also serves as a registration application for a set of intended subtypes, which are representative of some existing subtypes already registered under the "application" tree by their separate registrations.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 18, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of

the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

1. Introduction

The process of setting type in computer systems and other forms of text presentation systems uses fonts in order to provide visual representations of the glyphs. Just as with images, for example, there are a number of ways to represent the visual information of the glyphs. Early font formats often used bitmaps, as these could have been carefully tuned for maximum readability at a given size on low-resolution displays. More recently, scalable vector outline fonts have come into widespread use: in these fonts, the outlines of the glyphs are described, and the presentation system renders the outline in the desired position and size.

This document defines a top-level Internet Media Type type "font" under which different representation formats of fonts may be registered (e.g. a bitmap or outline formats). It should be emphasized that, just as under the "image" top-level type one does not find registration for a specific image, for example, "The Night-watch" (by Rembrandt) but instead "JPEG" (a compressed image data representation format), so, under "font" one will not find "Courier" (the name of a popular font) but perhaps "TTF", "OTF" or "SFNT" (the names of commonly used TrueType and OpenType font formats as well as their higher-level wrapper format).

2. Background and Justification

Historically there has not been a registration of formats for fonts. Most recently, there have been several representation formats registered as MIME subtypes under the "application" top-level type. However, with the rapid adoption of web fonts (based on the data from HTTP Archive [1] showing a huge increase in web font usage from 1% in the end of 2010 to 50% across all sites in the beginning of 2015) custom fonts on the web have become a core web resource. As the in-depth analysis [2] shows, the lack of the intuitive top-level font type is causing significant confusion among developers - while currently defined font subtypes are severely under-utilized there are many more sites that already use non-existent (but highly intuitive) media types such as "font/woff", "font/ttf" and "font/truetype". At the same time, the majority of sites resort to using generic types such as "application/octet-stream", "text/plain" and "text/html"; or use unregistrable types such as "application/x-font-ttf".

Contrary to our expectations, the officially defined IANA subtypes such as "application/font-woff" and "application/font-sfnt" see a very limited use - their adoption rates trail far behind as the

actual use of web fonts continues to increase. The members of the W3C WebFonts WG believe the use of "application" top-level type is not ideal. First, the "application" sub-tree is treated (correctly) with great caution with respect to viruses and other active code. Secondly, the lack of a top-level type means that there is no opportunity to have a common set of optional attributes, such as are specified here. Third, fonts have a unique set of licensing and usage restrictions, which makes it worthwhile to identify this general category with a unique top-level type.

The W3C WebFonts WG believes that the situation can be significantly improved if a set of font media types is registered using "font" as a dedicated top-level type. Based on the data analysis presented above, we believe that it is the presence of simple and highly intuitive media types for images that caused the widespread adoption of IANA's recommendations, where the correct usage of existing media types reaches over 97% for all subtypes in the "image" tree. The WG believes that, considering a rapid adoption of fonts on the web, the registration of the top-level media type for fonts along with the intuitive set of subtypes that reflect popular and widely used data formats would further stimulate the adoption of web fonts, significantly simplify web server configuration process and facilitate the proper use of IANA media type recommendations.

3. Security considerations

Fonts are interpreted data structures that represent collections of different tables containing data that represent different types of information, including glyph outlines in various formats, hinting instructions, metrics and layout information for multiple languages and writing systems, rules for glyph substitution and positioning, etc. Depending on the format used to represent the glyph data the font may contain TrueType, PostScript or SVG outlines and their respective hint instructions, where applicable. There are many existing, already standardized font table tags and formats that allow an unspecified number of entries containing predefined data fields for storage of variable length binary data. Many existing (TrueType, OpenType and OFF, SIL Graphite, WOFF, etc.) font formats are based on the table-based SFNT (scalable font) format which is extremely flexible, highly extensible and offers an opportunity to introduce additional table structures when needed, in a way that would not affect existing font rendering engines and text layout implementations. However, this very extensibility may present specific security concerns - the flexibility and ease of adding new data structures makes it easy for any arbitrary data to be hidden inside a font file. There is a significant risk that the flexibility of font data structures may be exploited to hide malicious binary content disguised as a font data component.

Fonts may contain 'hints', which are programmatic instructions that are executed by the font engine for the alignment of graphical elements of glyph outlines with the target display pixel grid. Depending on the font technology utilized in the creation of a font these hints may represent active code interpreted and executed by the font rasterizer. Even though hints operate within the confines of the glyph outline conversion system and have no access outside the font rendering engine, hint instructions can be, however, quite complex, and a maliciously designed complex font could cause undue resource consumption (e.g. memory or CPU cycles) on a machine interpreting it. Indeed, fonts are sufficiently complex, and most (if not all) interpreters cannot be completely protected from malicious fonts without undue performance penalties.

Widespread use of fonts as necessary component of visual content presentation warrants that a careful attention should be given to security considerations whenever a font is either embedded into an electronic document or transmitted alongside media content as a linked resource. While many existing font formats provide certain levels of protection of data integrity (such mechanisms include e.g. checksums and digital signatures), font data formats provide neither privacy nor confidentiality protection internally; if needed, such protection should be provided externally.

4. Definition and encoding

The "font" as the primary media content type indicates that the content identified by it requires certain graphic subsystem such as font rendering engine (and, in some cases, text layout and shaping engine) to process font data, which in turn may require certain level of hardware capabilities such as certain levels of CPU performance and available memory. The "font" media type does not provide any specific information about the underlying data format and how the font information should be interpreted - the subtypes defined within a "font" tree will name the specific font formats. Unrecognized subtypes of "font" should be treated as "application/octet-stream". Implementations may pass unrecognized subtypes to a common font-handling system, if such system is available.

5. Defined subtypes

In this section the initial entries under the top-level 'font' MIME type are documented. They also serve as examples for future registrations.

5.1. Generic SFNT font type

Type name: font

Subtype name: sfnt

Required parameters: None.

Optional parameters:

- 1) Name: Outlines Value: TTF, CFF, SVG

This parameter can be used to specify the type of outlines supported by the font. Value "TTF" shall be used when a font resource contains glyph outlines in TrueType format, value "CFF" shall be used to identify fonts containing PostScript/CFF outlines, and value SVG shall be used to identify fonts that include SVG outlines. TTF, CFF or SVG outlines can be present in various combinations in the same font file, therefore, multiple values for the same optional parameter may be defined.

- 2) Name: Layout

Value: OTF, AAT, SIL

This parameter identifies the type of implemented support for advanced text layout features. The predefined values "OTF", "AAT" and "SIL" respectively indicate support for OpenType text layout, Apple Advanced Typography or Graphite SIL. More than one shaping and layout mechanism may be supported by the same font file, therefore, multiple values for the same optional parameter may be defined.

Encoding considerations: Binary.

Interoperability considerations: As it was noted in the first paragraph of the "Security considerations" section, the same font format wrapper can be used to encode fonts with different types of glyph data represented as either TrueType or PostScript (CFF) outlines. Existing font rendering engines may not be able to process some of the particular outline formats, and downloading a font resource that contains unsupported glyph data format would result in inability of application to render and display text. Therefore, it would be extremely useful to clearly identify the format of the glyph outline data within a font using an optional parameter, and allow applications to make decisions about downloading a particular font resource sooner. Similar, another optional parameter is suggested to identify the type of text

shaping and layout mechanism that is supported by a font. Please note that as new outline formats and text shaping mechanisms may be defined in the future, the set of allowed values for two optional parameters defined by this section may be extended.

Published specification: ISO/IEC 14496-22 "Open Font Format" (OFF) specification being developed by ISO/IEC SC29/WG11.

Applications that use this media type: Any and all applications that are able to create, edit or display textual media content.

Additional information:

Magic number(s): The TrueType fonts and OFF / OpenType fonts containing TrueType outlines should use 0x00010000 as the 'sfnt' version number.

The OFF / OpenType fonts containing CFF data should use the tag 'OTTO' as 'sfnt' version number.

File extension(s): Font file extensions used for OFF / OpenType fonts: .ttf, .otf

Typically, .ttf extension is only used for fonts containing TrueType outlines, while .otf extension can be used for any OpenType/OFF font, either with TrueType or CFF outlines.

Macintosh file type code(s): (no code specified)

@font-face Format: none.

Fragment Identifiers none.

Person & email address to contact for further information: Vladimir Levantovsky (vladimir.levantovsky@monotype.com).

Intended usage: COMMON

Restrictions on usage: None

Author: The ISO/IEC 14496-22 "Open Font Format" specification is a product of the ISO/IEC JTC1 SC29/WG11.

Change controller: The ISO/IEC has change control over this specification.

5.2. TTF font type

Type name: font

Subtype name: ttf

Required parameters: None.

Optional parameters:

Name: Layout Value: OTF, AAT, SIL

This parameter identifies the type of support mechanism for advanced text layout features. The predefined values "OTF", "AAT" and "SIL" respectively indicate support for OpenType text layout, Apple Advanced Typography or Graphite SIL. More than one shaping and layout mechanism may be supported by the same font file, therefore, multiple values for the same optional parameter may be defined.

Encoding considerations: Binary.

Interoperability considerations: As was noted in the first paragraph of the "Security considerations" section, the same font format can be used to encode fonts supporting different types of outlines and/or text shaping and layout mechanisms. Existing font rendering engine implementations may not be able to process some of the particular layout table formats, and downloading a font resource that contains unsupported text shaping mechanism would result in inability of applications to display text properly. Therefore, it would be extremely useful to clearly identify the supported text shaping and layout data within a font using an optional parameter, and allow applications to make decisions about downloading a particular font resource sooner. Please note that as new text shaping mechanisms may be defined in the future, the set of allowed values for the optional parameter defined by this section may be extended.

Published specification: ISO/IEC 14496-22 "Open Font Format" (OFF) specification being developed by ISO/IEC SC29/WG11.

Applications that use this media type: Any and all applications that are able to create, edit or display textual media content.

Additional information:

Magic number(s): The TrueType fonts and OFF / OpenType fonts containing TrueType outlines should use 0x00010000 as the 'sfnt' version number.

File extension(s): Font file extensions used for TrueType / OFF / OpenType fonts: .ttf, .otf

Typically, .ttf extension is only used for fonts containing TrueType outlines, while .otf extension may be used for any OpenType/OFF font, either with TrueType or CFF outlines.

Macintosh file type code(s): (no code specified)

@font-face Format: truetype

Fragment Identifiers none.

Person & email address to contact for further information: Vladimir Levantovsky (vladimir.levantovsky@monotype.com).

Intended usage: COMMON

Restrictions on usage: None

Author: The ISO/IEC 14496-22 "Open Font Format" specification is a product of the ISO/IEC JTC1 SC29/WG11.

Change controller: The ISO/IEC has change control over this specification.

5.3. OTF font type

Type name: font

Subtype name: otf

Required parameters: None.

Optional parameters

Name: Outlines Value: TTF, CFF, SVG

This parameter can be used to specify the type of outlines supported by the font. Value "TTF" shall be used when a font resource contains glyph outlines in TrueType format, value "CFF" shall be used to identify fonts containing PostScript/CFF outlines, and value SVG shall be used to identify fonts that include SVG outlines. TTF, CFF or SVG outlines can be present

in various combinations in the same font file, therefore, multiple values for the same optional parameter may be defined.

Encoding considerations: Binary.

Interoperability considerations: As it was noted in the first paragraph of the "Security considerations" section, the same font format can be used to encode fonts with different types of glyph data represented as either TrueType, PostScript (CFF) or SVG outlines. Existing font rendering engines may not be able to process some of the particular outline formats, and downloading a font resource that contains unsupported glyph data format would result in inability of application to render and display text. Therefore, it would be extremely useful to clearly identify the format of the glyph outline data within a font using an optional parameter, and allow applications to make decisions about downloading a particular font resource sooner. Please note that as new outline formats may be defined in the future, the set of allowed values for the optional parameter defined in this section may be extended.

Published specification: ISO/IEC 14496-22 "Open Font Format" (OFF) specification being developed by ISO/IEC SC29/WG11.

Applications that use this media type: Any and all applications that are able to create, edit or display textual media content.

Additional information:

Magic number(s): The TrueType fonts and OFF / OpenType fonts containing TrueType outlines should use 0x00010000 as the 'sfnt' version number.

The OFF / OpenType fonts containing CFF data should use the tag 'OTTO' as 'sfnt' version number.

File extension(s): Font file extensions used for OFF / OpenType fonts: .ttf, .otf

Typically, .ttf extension is only used for fonts containing TrueType outlines, while .otf extension can be used for any OpenType/OFF font, either with TrueType, CFF or SVG outlines.

Macintosh file type code(s): (no code specified)

@font-face Format: opentype

Fragment Identifiers none.

Person & email address to contact for further information: Vladimir Levantovsky (vladimir.levantovsky@monotype.com).

Intended usage: COMMON

Restrictions on usage: None

Author: The ISO/IEC 14496-22 "Open Font Format" specification is a product of the ISO/IEC JTC1 SC29/WG11.

Change controller: The ISO/IEC has change control over this specification.

5.4. WOFF 1.0

Type name: font

Subtype name: woff

Required parameters: None.

Optional parameters: None.

Encoding considerations: Binary.

Interoperability considerations: None.

Published specification: This media type registration is extracted from the WOFF specification [3] at W3C.

Applications that use this media type: WOFF is used by Web browsers, often in conjunction with HTML and CSS.

Additional information:

Magic number(s): The signature field in the WOFF header MUST contain the "magic number" 0x774F4646

File extension(s): woff

Macintosh file type code(s): (no code specified)

Macintosh Universal Type Identifier code: "org.w3c.woff"

@font-face Format: woff

Fragment Identifiers: none.

Person & email address to contact for further information: Chris Lilley (www-font@w3.org).

Intended usage: COMMON

Restrictions on usage: None

Author: The WOFF specification is a work product of the World Wide Web Consortium's WebFonts Working Group.

Change controller: The W3C has change control over this specification.

5.5. WOFF 2.0

Type name: font

Subtype name: woff2

Required parameters: None.

Optional parameters: None.

Encoding considerations: Binary.

Interoperability considerations: WOFF 2.0 is an improvement on WOFF 1.0. The two formats have different Internet Media Types, different @font-face formats, and may be used in parallel.

Published specification: This media type registration is extracted from the WOFF 2.0 specification [4] at W3C.

Applications that use this media type: WOFF 2.0 is used by Web browsers, often in conjunction with HTML and CSS.

Additional information:

Magic number(s): The signature field in the WOFF header MUST contain the "magic number" 0x774F4632 ('WOFF2')

File extension(s): woff2

Macintosh file type code(s): (no code specified)

Macintosh Universal Type Identifier code: "org.w3c.woff2"

@font-face Format: woff2

Fragment Identifiers none.

Person & email address to contact for further information: Chris Lilley (www-font@w3.org).

Intended usage: COMMON

Restrictions on usage: None

Author: The WOFF2 specification is a work product of the World Wide Web Consortium's WebFonts Working Group.

Change controller: The W3C has change control over this specification.

6. References

6.1. URIs

[1] <http://httparchive.org/trends.php?s=All&minlabel=Nov+15+2010&maxlabel=Feb+15+2015>

[2] <http://goo.gl/zbDhUN>

[3] <http://www.w3.org/TR/WOFF>

[4] <http://www.w3.org/TR/WOFF2>

Author's Address

Chris Lilley
W3C
2004 Route des Lucioles
Sophia Antipolis 06902
France

Email: chris@w3.org