

The font Top Level Type
draft-ietf-justfont-toplevel-06

Abstract

This memo serves to register and document the "font" Top Level 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 in use, and currently 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 June 16, 2017.

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.

1. Specification Development

[Note to the RFC Editor: Please remove this section upon publication.]

This section is non-normative. The source for this specification is maintained on GitHub [\[1\]](#). The issues list [\[2\]](#) is also on GitHub. Discussion should be on the mailing list justfont@ietf.org [\[3\]](#).

2. 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.

Over time, a number of standard formats for recording font descriptions have evolved. This document defines a new top-level Internet Media Type "font" according to [Section 4.2.7 of \[RFC6838\]](#). This top-level type indicates that the content specifies font data. Under this top-level type, different representation formats of fonts may be registered.

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

3. Background and Justification

Historically there has not been a registration of formats for fonts. More recently, there have been several representation formats registered as media subtypes under the "application" top-level type (for example, application/font-woff). However, with the rapid adoption of web fonts (based on the data from HTTP Archive [\[HTTP-Archive-Trends\]](#) 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 [\[Font-Media-Type-Analysis\]](#) shows, the lack of the intuitive top-level font type is causing significant confusion

Lilley

Expires June 16, 2017

[Page 2]

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 the expectations of the W3C WebFonts Working Group which developed WOFF, the officially defined media types 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 concluded that 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 parameters, 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 decided [[WG-tlt](#)] 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 conclude that it is the presence of simple and highly intuitive media types for images that caused their widespread adoption, where the correct usage of existing media types reaches over 97% for all subtypes in the "image" tree. The WG considers 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 media types for fonts.

4. 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. In particular, the hinting instructions for TrueType glyphs represent executable code which has the potential to be maliciously constructed (for example, intended to hang the interpreter). There are many existing, already standardized font table tags and formats that allow an unspecified number of entries containing predefined

Lilley

Expires June 16, 2017

[Page 3]

data fields for storage of variable length binary data. Many existing font formats (TrueType [[truetype-wiki](#)], OpenType and OFF [[opentype-wiki](#)], SIL Graphite, WOFF, etc.) 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 an upward-compatible 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 that most (if not all) interpreters cannot be completely protected from malicious fonts without undue performance penalties.

Widespread use of fonts as necessary components of visual content presentation warrants that 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.

5. IANA Considerations

This specification registers a new top-level type, "font", in the standards tree; adds it as an alternative value of "Type Name" in the media types registration form [[Media-Type-Registration](#)]; and registers several subtypes for it.

5.1. 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 it as 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 name the specific font formats. Unrecognized sub-types of "font" should be treated as "application/octet-stream". Implementations may pass unrecognized subtypes to a common font-handling system, if such a system is available.

5.2. Fragment Identifiers for Font Collections

Fragment identifiers for font collections identify one font in the collection by the PostScript name (name ID=6) [[ISO.14496-22.2015](#)]. This is a string, no longer than 63 characters and restricted to the printable ASCII subset, codes 33 - 126, except for the 10 characters '[', ']', '(', ')', '{', '}', '<', '>', '/', '%' which are forbidden by [[ISO.14496-22.2015](#)].

In addition, the following 6 characters could occur in the PostScript name but are forbidden in fragments by [[RFC3986](#)] and thus must be escaped: '"', '#', '\\', '^', '`', '|'.

If (following un-escaping) this string matches one of the PostScript names in the name table, that font is selected. For example, "#Foo-Bold" refers to the font with PostScript name "Foo-Bold" and "#Caret%5Estick" refers to the font with PostScript name "Caret^stick". If the name does not match, or if a fragment is not specified, the first font in the collection is matched. Note that the order of fonts in collections may change as the font is revised, so relying on a particular font in a collection always being first is unwise.

5.3. Registration Procedure

New font formats should be registered using the online form [[Media-Type-Registration](#)]. [RFC 6838](#) [[RFC6838](#)] should be consulted on registration procedures. In particular the font specification should preferably be freely available. If the font format can contain multiple fonts, a fragment identifier syntax should also be defined.

Note that new parameter sub-values may be defined in the future. If an implementation does not recognize a sub-value in the comma-

separated list, it should ignore the sub-value and continue processing the other sub-values in the list.

5.4. Subtype Registrations

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

For each subtype, an @font-face format identifier is listed. This is for use with the @font-face src descriptor, defined by the CSS3 Fonts specification [[W3C.CR-css-fonts-3-20131003](#)]. That specification is normative; the identifiers here are informative.

5.4.1. Generic SFNT Font Type

Type name: font

Subtype name: sfnt

Required parameters: None.

Optional parameters:

- 1) Name: outlines Values: a comma-separated subset of: TTF, CFF, SVG

This parameter can be used to specify the type of outlines provided 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 [[cff-wiki](#)], and value SVG [[svg-wiki](#)] 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, this optional parameter is a list containing one or more items, separated by commas. Order in the list is not significant.

- 2) Name: layout

Values: a comma-separated subset of: OTL, AAT, SIL

This parameter identifies the type of implemented support for advanced text layout features. The predefined values "OTL", "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 provided by the same font file, therefore, this optional parameter is a list

containing one or more items, separated by commas. Order in the list is not significant.

Encoding considerations: Binary.

Interoperability considerations: As it was noted in the first paragraph of the "Security considerations" section, a single font file can contain encoding of the same glyphs using several different representations, e.g., both TrueType and 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 only unsupported glyph data format would be futile. Therefore, it is 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. Similarly, another optional parameter identifies the type of text shaping and layout mechanism that is provided by a font.

Published specification: ISO/IEC 14496-22 "Open Font Format" (OFF) specification [[ISO.14496-22.2015](#)] being developed by ISO/IEC SC29/WG11.

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

Note that font/sfnt is an abstract type, from which the (widely used in practice) font/ttf and font/otf types are conceptually derived. Use of font/sfnt is likely to be rare in practice, and might be confined to

uncommon combinations such as font/sfnt; outlines=sil which do not have a shorter type

cases where a new parameter value is registered

test cases, experimentation, etc

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)

Macintosh Universal Type Identifier code: "public.font"

@font-face Format: none.

Fragment Identifiers none.

Deprecated Alias: The existing registration "application/font-sfnt" is deprecated in favor of "font/sfnt".

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.

5.4.2. TTF Font Type

Type name: font

Subtype name: ttf

Required parameters: None.

Optional parameters:

Name: layout Values: a comma-separated subset of: OTL, AAT, SIL

This parameter identifies the type of support mechanism for advanced text layout features. The predefined values "OTL", "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 provided by the same

font file, therefore, this optional parameter is a list containing one or more items, separated by commas. Order in the list is not significant.

Encoding considerations: Binary.

Interoperability considerations: As it was noted in the first paragraph of the "Security considerations" section, a single font file can contain encoding of the same glyphs using several different representations, e.g., both TrueType and 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 only unsupported glyph data format would be futile. Therefore, it is 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. Similarly, another optional parameter identifies the type of text shaping and layout mechanism that is provided by a font.

Published specification: ISO/IEC 14496-22 "Open Font Format" (OFF) specification [[ISO.14496-22.2015](https://www.iso.org/standard/68811.html)] being developed by ISO/IEC SC29/WG11.

Applications that use this media type: 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)

Macintosh Universal Type Identifier code: "public.truetype-font"

@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.

5.4.3. OTF Font Type

Type name: font

Subtype name: otf

Required parameters: None.

Optional parameters

Name: outlines Values: a comma-separated subset of: TTF, CFF, SVG

This parameter can be used to specify the type of outlines provided 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, this optional parameter is a list containing one or more items, separated by commas. Order in the list is not significant.

Encoding considerations: Binary.

Interoperability considerations: As it was noted in the first paragraph of the "Security considerations" section, a single font file can contain encoding of the same glyphs using several different representations, e.g., both TrueType and 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 only unsupported glyph data format would be futile. Therefore, it is 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. Similarly, another

optional parameter identifies the type of text shaping and layout mechanism that is provided by a font.

Published specification: ISO/IEC 14496-22 "Open Font Format" (OFF) specification [[ISO.14496-22.2015](https://www.iso.org/standard/68811.html)] being developed by ISO/IEC SC29/WG11.

Applications that use this media type: 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 outlines should use the tag 'OTTO' as 'sfnt' version number. There is no magic number for SVG outlines; these are always accompanied by either TrueType or CFF outlines and thus use the corresponding magic 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)

Macintosh Universal Type Identifier code: "public.opentype-font"

@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.

5.4.4. Collection Font Type

Type name: font

Subtype name: collection

Required parameters: None.

Optional parameters

Name: outlines Values: a comma-separated subset of: TTF, CFF, SVG

This parameter can be used to specify the type of outlines provided 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, this optional parameter is a list containing one or more items, separated by commas. Order in the list is not significant.

Encoding considerations: Binary.

Interoperability considerations: As it was noted in the first paragraph of the "Security considerations" section, a single font file can contain encoding of the same glyphs using several different representations, e.g., both TrueType and 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 only unsupported glyph data format would be futile. Therefore, it is 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. Similarly, another optional parameter identifies the type of text shaping and layout mechanism that is provided by a font.

Published specification: ISO/IEC 14496-22 "Open Font Format" (OFF) specification [[ISO.14496-22.2015](#)] being developed by ISO/IEC SC29/WG11.

Applications that use this media type: 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 outlines should use the tag 'OTTO' as 'sfnt' version number. There is no magic number for SVG outlines; these are always accompanied by either TrueType or CFF outlines and thus use the corresponding magic number.

File extension(s): Font file extensions used for OFF / TrueType and OpenType fonts: .ttc

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

Macintosh Universal Type Identifier code: "public.truetype-collection-font"

@font-face Format: collection

Fragment Identifiers: See [Section 5.2](#).

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.

[5.4.5](#). 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 updates the WOFF specification [[W3C.REC-WOFF-20121213](#)] 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 ('woff')

File extension(s): woff

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

Macintosh Universal Type Identifier code: "org.w3.woff"

@font-face Format: woff

Fragment Identifiers: none.

Deprecated Alias: The existing registration "application/font-woff" is deprecated in favor of "font/woff".

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.

[5.4.6.](#) **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 [[W3C.CR-WOFF2-20150414](#)] 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 ('wOF2')

File extension(s): woff2

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

Macintosh Universal Type Identifier code: "org.w3.woff2"

@font-face Format: woff2

Fragment Identifiers: See [Section 5.2](#).

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. 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>>.

- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, [RFC 3986](http://www.rfc-editor.org/info/rfc3986), DOI 10.17487/RFC3986, January 2005, <<http://www.rfc-editor.org/info/rfc3986>>.
- [RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", [BCP 13](http://www.rfc-editor.org/info/rfc6838), [RFC 6838](http://www.rfc-editor.org/info/rfc6838), DOI 10.17487/RFC6838, January 2013, <<http://www.rfc-editor.org/info/rfc6838>>.
- [W3C.CR-css-fonts-3-20131003]
Daggett, J., "CSS Fonts Module Level 3", World Wide Web Consortium CR CR-css-fonts-3-20131003, October 2013, <<http://www.w3.org/TR/2013/CR-css-fonts-3-20131003>>.
- [ISO.14496-22.2015]
International Organization for Standardization, "Coding of audio-visual objects Part 22: Open Font Format", ISO Standard 14496-22, 10 2015, <http://standards.iso.org/ittf/PubliclyAvailableStandards/c066391_ISO_IEC_14496-22_2015.zip>.
- [W3C.REC-WOFF-20121213]
Kew, J., Leming, T., and E. Blokland, "WOFF File Format 1.0", World Wide Web Consortium Recommendation REC-WOFF-20121213, December 2012, <<http://www.w3.org/TR/2012/REC-WOFF-20121213>>.
- [W3C.CR-WOFF2-20150414]
Levantovsky, V. and R. Levien, "WOFF File Format 2.0", World Wide Web Consortium WD CR-WOFF2-20150414, March 2016, <<https://www.w3.org/TR/2016/CR-WOFF2-20160315/>>.

6.2. Informative References

- [cff-wiki]
"CFF", <https://en.wikipedia.org/wiki/PostScript_fonts#Compact_Font_Format>.
- [opentype-wiki]
"OpenType", <<https://en.wikipedia.org/wiki/OpenType>>.
- [truetype-wiki]
"TrueType", <<https://en.wikipedia.org/wiki/TrueType>>.
- [svg-wiki]
"SVG", <https://en.wikipedia.org/wiki/Scalable_Vector_Graphics>.

[HTTP-Archive-Trends]

Kuetell, D., "HTTP Archive trend analysis", March 2015, <<http://httparchive.org/trends.php?s=All&minlabel=Nov+15+2015&maxlabel=Feb+15+2015#perFonts>>.

[Font-Media-Type-Analysis]

Kuetell, D., "Web Font Media Type (mime type) Analysis 2015", 2015, <<http://goo.gl/zbDhUN>>.

[WG-tlt] W3C, "ACTION-164: Bring widely used top-level-type to w3c-ietf liaison", 2015, <<https://www.w3.org/Fonts/WG/track/actions/164>>.

[Media-Type-Registration]

IANA, "Application for a Media Type", <<http://www.iana.org/form/media-types>>.

6.3. URIs

[1] <https://github.com/svgeesus/ietf-justfont>

[2] <https://github.com/svgeesus/ietf-justfont/issues>

[3] <mailto:justfont@ietf.org>

Author's Address

Chris Lilley
W3C
2004 Route des Lucioles
Sophia Antipolis 06902
France

Email: chris@w3.org

