

Independent  
Internet-Draft  
Updates: [7231](#) (if approved)  
Intended status: Experimental  
Expires: May 14, 2015

M. Karcz  
UKLO Tczew  
November 10, 2014

**Unified User-Agent String  
draft-karcz-uuas-01**

Abstract

User-Agent is a HTTP request-header field. It contains information about the user agent originating the request, which is often used by servers to help identify the scope of reported interoperability problems, to work around or tailor responses to avoid particular user agent limitations, and for analytics regarding browser or operating system use. Over the years contents of this field got complicated and ambiguous. That was the reaction for sending altered version of websites to web browsers other than popular ones. During the development of the WWW, authors of the new web browsers used to construct User-Agent strings similar to Netscape's one. Nowadays contents of the User-Agent field are much longer than 15 years ago. This Memo proposes the Uniform User-Agent String as a way to simplify the User-Agent field contents, while maintaining the previous possibility of their use.

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 May 14, 2015.

## Copyright Notice

Copyright (c) 2014 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.

This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">1.1.</a>	Conformance . . . . .	<a href="#">3</a>
<a href="#">1.2.</a>	Syntax Notation . . . . .	<a href="#">3</a>
<a href="#">1.2.1.</a>	Whitespaces . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Use of the User-Agent strings . . . . .	<a href="#">3</a>
<a href="#">3.</a>	Definition of Proposed Format . . . . .	<a href="#">3</a>
<a href="#">3.1.</a>	Standard String . . . . .	<a href="#">4</a>
<a href="#">3.2.</a>	Regular String . . . . .	<a href="#">4</a>
<a href="#">3.3.</a>	Web Browser String . . . . .	<a href="#">5</a>
<a href="#">4.</a>	ABNF Definition of UUAS . . . . .	<a href="#">7</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">8</a>
<a href="#">6.</a>	IANA Considerations . . . . .	<a href="#">8</a>
<a href="#">7.</a>	Acknowledgments . . . . .	<a href="#">8</a>
<a href="#">8.</a>	References . . . . .	<a href="#">8</a>
	Author's Address . . . . .	<a href="#">9</a>

## [1.](#) Introduction

Nowadays User-Agent strings are long, complicated and often ambiguous. (e.g. "Mozilla/4.0 (compatible; MSIE 6.0; X11; Linux i686; en) Opera 8.01" - it is Opera Browser, but it can be read as Internet Explorer or Netscape Navigator.) This document specifies a new, easy and clear format of Unified User-Agent String (UUAS), which allows simple distinction between user agents, maintaining most of the features of the existing solutions.



### **1.1. Conformance**

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 [[RFC2119](#)].

### **1.2. Syntax Notation**

This specification uses the Augmented Backus-Naur Form (ABNF) notation of [[RFC5234](#)]. [Section 4](#) contains a full syntax definition of the Unified User-Agent String.

#### **1.2.1. Whitespaces**

This specification uses two rules to denote the use of linear whitespace: OWS (optional whitespace) and RWS (required whitespace). They are defined in [Section 3.2.3 of \[RFC7230\]](#).

## **2. Use of the User-Agent strings**

Generally, the User-Agent header field was intended for statistical purposes. However, in mid-90. during the "browser wars" data provided by this field became used to alter the content of the resources before sending them to the user, or even to prevent users of particular browser the access to resources. To avoid these protections, software vendors started to change their identifiers in a way resembling User-Agent strings of the most popular browsers. During the years it has made these identifiers much more complicated, ambiguous and difficult to parse.

Nowadays User-Agent strings are still used for statistical purposes, but also for avoiding limitations of particular implementations. However, in modern browsers these limitations greatly decreased and "user agent spoofing" is now unnecessary. Unfortunately, there are a lot of websites still discriminating particular web browsers.

Unified User-Agent String is intended to propose a way for simplifying, clarifying and standarizing the content of User-Agent HTTP header field. Furthermore, if it becomes widespread, it will be able to reduce the practice of "user agent spoofing" and discrimination of particular groups of the Internet users.

## **3. Definition of Proposed Format**

This document proposes a formal definition of three types of User-Agent string: standard string, regular string and web browser string.



```
User-Agent = uuas  
uuas = standard-string / regular-string / browser-string
```

Standard string is intended to maintain backward compatibility with existing implementations and it is the same simple format as defined in [\[RFC7230\]](#).

Regular string introduces a degree of standardization making every theoretical UUAS parser able to obtain information from it.

Web browser string is designed for modern graphical web browsers and proposes a set of signatures, which should form together a clear and unequivocal application identifier.

### **[3.1.](#) Standard String**

The standard User-Agent string MUST be generated in conformance with [Section 5.5.3 of \[RFC7231\]](#). The standard User-Agent string consists of one or more product identifiers, each followed by zero or more comments ([Section 3.2 of \[RFC7230\]](#)), which together identify the user agent software.

Standard string syntax definition:

```
standard-string = product *( RWS ( product / comment ) )
```

The product identifiers and comments SHOULD be listed in decreasing order of their significance. Each of them consists of a name and OPTIONAL version number.

In the standard string a sender SHOULD limit generated product identifiers to what is necessary to identify the product; a sender MUST NOT generate advertising or other nonessential information within the product identifier. A sender SHOULD NOT place non-version-related information in version number part of product identifier. In the standard string successive versions of the same product SHOULD differ only in the version part of the identifier.

Example:

```
CERN-LineMode/2.15 libwww/2.17b3
```

### **[3.2.](#) Regular String**

Regular Unified User-Agent String is intended for request senders other than graphical web browsers and general web crawlers. It MUST provide a signature of the operating system or platform (eg. in case of runtime environments) used to generate the request at the first



position in the comment after the first product identifier. After this signature the regular string MAY contain any comments and next product identifiers. Only this information MUST be provided, because this format is designed for cases, when the server does not need to know the exact parameters of the application originating the request. In such cases this string can be applicable in statistical purposes or in adapting the server's response to capabilities of particular software platforms (eg. for indicating the need for adding carriage returns before the newlines).

Regular string syntax definition:

```
regular-string = product RWS "(" os [ sc 1*ctext ] ")"
                *( RWS ( product / comment ) )
```

Regular Unified User-Agent Strings are syntactically compliant with the standard definition.

Example:

```
Wget/1.11.1 (Red Hat modified)
```

### **3.3. Web Browser String**

Web Browser User-Agent String is a format of this field-value intended for identifying modern graphical web browsers, which are compatible with HTML5, CSS3 or other modern web technologies. Web browser string MUST contain "Mozilla/5.0" tag at the beginning for historical reasons. This helps avoid the recognition of browsers as very old ones. Web Browser UUAS MUST also contain "Gecko" tag. This can avoid delivering impaired versions of websites to modern but not Gecko-based client applications. It is also in conformance with Section 6.6.1.1 of [[W3C.REC-html5-20141028](#)].

Web browser string syntax definition:

```
browser-string = Mozilla-tag
                RWS "(" *( signature sc ) os
                *( sc signature ) [ sc language ]
                *( sc signature ) [ sc rvtag ] ")"
                RWS Gecko-string
                *( RWS ( product / comment ) )
```

Like regular string, Web Browser Unified User-Agent String MUST provide information about software platform. Fields contained between brackets (comments) SHOULD be separated by semicolons with optional space. Application MAY also include language tag in its





User-Agent string. Then it MUST be a Language-Tag in accordance with [\[RFC5646\]](#).

Due to the fact that the application originating the request cannot provide its version info in the first product identifier, it SHOULD place its version number in the separate revision tag.

Of course, a sender can add to the string any valid product identifiers and comments, but this Memo is intended to simplify and clarify this element of the protocol. In the web browser string there MUST be at least one signature allowing to identify particular client application product. Also the order of platform, language and revision signatures MUST NOT be changed.

This type of UUAS SHOULD be also used by general web crawlers. It can help avoid certain unfair practices relying on delivering other resources to web browsers, other to web crawlers.

Example:

Mozilla/5.0 (Windows NT 6.3; Trident/7.0; rv:11.0) like Gecko



#### 4. ABNF Definition of UUAS

```
; Unified User-Agent String general definition
User-Agent = uuas
uuas = standard-string / regular-string / browser-string

; Standard string, as described in [RFC7231]
standard-string = product *( RWS ( product / comment ) )

; Regular string, recommended for non-browsers
regular-string = product RWS "(" os [ sc 1*ctext ] ")"
                  *( RWS ( product / comment ) )

; String recommended for web browsers and crawlers
browser-string = Mozilla-tag
                  RWS "(" *( signature sc ) os
                  *( sc signature ) [ sc language ]
                  *( sc signature ) [ sc rvtag ] ")"
                  RWS Gecko-string
                  *( RWS ( product / comment ) )

; Tags and signatures definitions
signature = product / 1*schar
os = 1*schar
language = <Language-Tag, see [RFC5646], Section 2.1>
rvtag = "rv:" OWS token
Mozilla-tag = "Mozilla/5.0"
Gecko-string = Gecko-tag
               / ( product RWS "(" *ctext
                   RWS Gecko-tag [ RWS 1*ctext ] ")" )
Gecko-tag = ["like "] "Gecko" ["/20100101"]

; Additional definitions
product = <product, see [RFC7231], Section 5.5.3>
comment = <comment, see [RFC7230], Section 3.2.6>
ctext = <ctext, see [RFC7230], Section 3.2.6>
schar = tchar / HTAB / SP / obs-text
token = <token, see [RFC7230], Section 3.2.6>
tchar = <tchar, see [RFC7230], Section 3.2.6>
obs-text = <obs-text, see [RFC7230], Section 3.2.6>
sc = ";" OWS
OWS = <OWS, see [RFC7230], Section 3.2.3>
RWS = <RWS, see [RFC7230], Section 3.2.3>
```



## **5. Security Considerations**

Implementations are encouraged not to use the product tokens of other implementations in order to declare compatibility or identity with them beyond the scope prescribed in this document, as this circumvents the purpose of the User-Agent field.

A user agent SHOULD NOT generate a User-Agent field containing needlessly fine-grained detail and SHOULD limit the addition of subproducts by third parties. Overly detailed User-Agent strings increase request latency and the risk of a user being identified against their wishes. In theory, this can make it easier for an attacker to exploit known security holes; in practice, attackers tend to try all potential holes regardless of the software being used. But when User-Agent string is combined with other characteristics of the application, particularly if the client application sends excessive details about the user's system or extensions, the risk of successful attack gets higher.

As User-Agent strings are text data, they can be used to carry out attacks by causing buffer overflows or changing formatting strings. Implementers should secure their applications against such practices.

Data provided by User-Agent header field can be used to discriminate the users of particular client applications by preventing them accessing the requested resources or replacing them with false ones.

## **6. IANA Considerations**

This document has no actions for IANA.

## **7. Acknowledgments**

I would like to thank my English teacher, who devoted her time to conduct a linguistic revision of this Memo.

## **8. References**

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), [BCP 14](#), March 1997.
- [RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", [RFC 5646](#), September 2009.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", [RFC 5234](#), STD 68, January 2008.



- [RFC7230] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", [RFC 7230](#), June 2014.
- [RFC7231] Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", [RFC 7231](#), June 2014.
- [W3C.REC-html5-20141028]  
Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Doyle Navara, E., O'Connor, E., and S. Pfeiffer, "HTML5", World Wide Web Consortium Recommendation REC-html5-20141028, October 2014,  
<<http://www.w3.org/TR/2014/REC-html5-20141028/>>.

## Author's Address

Mateusz Karcz  
Uniwersyteckie Katolickie Liceum Ogólnokształcące w Tczewie  
6 Wodna Street  
Tczew, PM 83-100  
PL

Email: [mateusz.karcz\(at\)interia.eu](mailto:mateusz.karcz@interia.eu)



