Universal Acceptance

Universal Acceptance of Popular Browsers

17 August 2017

TABLE OF CONTENTS

Report UASG016

Executive Summary

Universal Acceptance (UA) helps create a more inclusive Internet as it ensures that all domain names and email addresses can be used by all Internet-enabled applications, devices and systems. The top-level domains (TLDs) in the Domain Name System (DNS) expanded dramatically in 2010, which now includes more than 1,500 TLDs. Many of those domains are longer than the legacy two or three-character domain name (e.g. .com, .edu and .org) or are in non-Latin based scripts (such as Chinese, Arabic or Cyrillic).

UA addresses an issue that prevents some Internet users from successfully interacting online. The problem can arise when applications reject or don't treat all parts of the domain name correctly, which can occur if those domain names are longer than three characters (e.g., .photography) or in different languages (e.g., .рф for "Russia Federation"). While UA issues are not new, developers frequently make over-simplified assumptions when processing domain names, which are based on Internet conventions from decades ago and are specific to two or three-letter domains. UA is essential for the continued expansion of the Internet and provides a gateway to the next billion Internet users.

Organizations worldwide have made significant progress toward UA, although there is still more work to be done. The Universal Acceptance Steering Group (UASG), an Internet community initiative that was founded in February 2015 and tasked with undertaking activities that will effectively promote the Universal Acceptance of all valid domain names and email addresses, commissioned a report on the UA-readiness of popular browsers.

Results showed that of the major browsers tested, only one – Internet Explorer – was fully UA compliant. Four others performed well but had minor issues. Mobile platforms fared worse – which should be an area of focus for organizations seeking to become UA-ready. Much of the developing world accesses the Internet primarily through mobile devices.

The report demonstrates that there is still work to be done by even the major browser developers. The goal of the report is to educate the Internet community on the state of Universal Acceptance, and spotlight specific areas where companies can make improvements to become UA-ready.

Results Overview

This document contains an overview of the results of the UASG browser evaluation that was performed in Q2 of 2017.

After performing individual tests of 17 URLs in eight browsers on six different operating systems (four desktop, two mobile), only Internet Explorer on the desktop performed completely as expected – meaning the expected webpage loaded and was displayed properly.

Most browsers running on a desktop platform (Windows 10, macOS 10.12, Ubuntu 17.04) performed very well, Vivaldi being the exception of those tested. Of the others, Chrome, Opera, Safari and Edge failed to correctly render mixed RTL1.ASCII URLs in the tab title bar. Neither Firefox or Safari handle the open dot " ｡ " as a label delimiter, which is recommended by UASG (see UASG004, also Appendix A). This leads to search results being displayed, instead of the browser loading the expected webpage.

The results were more varied on the two mobile platforms tested (iOS 10.3 and Android 7.0). Firefox and Opera had very poor results because the location bar displayed URLs in Punycode instead of in Unicode in almost all cases (Firefox on Android renders URLs with no path component in Unicode, see Table 2). There were no obvious settings in either browser to change this behaviour. Additionally, all the browsers tested had at least one of the minor issues encountered in the desktop testing.

It is also noted that in several test cases where a site is secured with HTTPS the certificate name is displayed only in Punycode.

The testing criteria are described in detail here: UASG Browser Evaluation Criteria

The test results are provided in detail here: UASG Browser Evaluation Results Details

[1] RTL-Right to Left (Arabic, Hebrew)

Specific Results

The following tables provide an overview of the URLs used in the evaluation and the results for browsers on desktop and mobile platforms. The URLs are taken from UASG 0004 (the numeric identifiers here are used in the document for brevity).

Desktop platforms

Testing was performed on Windows 10, macOS 10.12, Ubuntu 17.04 and no variations were found in the results for a particular browsers when running on different desktop platforms, hence no platforms are shown in Table 1 below.

Table 1: Desktop platform results

Result key:
Y - Passes all tests
A - Fails to load the correct page
B - Fails to display URL correctly in location bar
C - Fails to display URL correctly in title bar

Test ID	Test Data	Chrome	Firefox	Opera	Safari	Edge	IE	Vivaldi
1	ua-test.link	Y	Y	Y	Y	Y	Y	Y
2	ua-test.technology	Y	Y	Y	Y	Y	Y	Y
3	普遍接受-测试.top	Y	Y	Y	Y	Y	Y	B
4	ua-test.世界	Y	Y	Y	Y	Y	Y	B
5	普遍接受-测试.世界	Y	Y	Y	Y	Y	Y	B
6	普遍接受-测试。世界	Y	A	Y	A	Y	Y	A
7	ua-test.xn--rhqv96g	Y	Y	Y	Y	Y	Y	B
8	xn----f38am99bqvcd5liy1cxsg.top	Y	Y	Y	Y	Y	Y	B
9	xn----f38am99bqvcd5liy1cxsg.xn--rhqv96g	Y	Y	Y	Y	Y	Y	B
10	اختبار-القبولالعالمي.top	C	Y	C	C	C	Y	B
11	اختبار-القبولالعالمي.شبكة	Y	Y	Y	Y	Y	Y	B
12	ua-test.link/我的页面	Y	Y	Y	Y	Y	Y	Y
13	ua-test.technology/我的页面	Y	Y	Y	Y	Y	Y	Y
14	普遍接受-测试.top/我的页面	Y	Y	Y	Y	Y	Y	B
15	ua-test.世界/我的页面	Y	Y	Y	Y	Y	Y	B
16	普遍接受-测试.世界/我的页面	Y	Y	Y	Y	Y	Y	B
17	普遍接受-测试。世界/我的页面	Y	A	Y	A	Y	Y	B

Observations

- On the desktop, Internet Explorer had the best results (passing all the test cases), Vivaldi has the worst displaying Punycode for all relevant test cases (with no obvious setting to change this). The remaining browsers has issues with only 1 or 2 test cases (typically mixed those involving RTL labels (10, 11) and the Chinese 'open dot' character (6,17).

- On the desktop, both Edge and Internet Explorer will (by design) display Punycode in the location bar if the relevant language packs are not installed. With the language pack installed they display Unicode correctly in all but one case (see below).

Mobile platforms

Testing was done on iOS 10.3 and Android 7.0

Table 2: Mobile platform results

Result key:
Y - Passes all tests
A - Fails to load the correct page
B - Fails to display URL correctly in location bar
C - Fails to display URL correctly in title bar
D,E - Other non-failing issues seen

		Chrome		Firefox		Opera		Safari	Samsung Browser
Test ID	Test Data	Android	iOS	Android	iOS	Android	iOS	iOS	Android
1	ua-test.link	Y	Y	Y	Y	Y	Y	Y	Y
2	ua-test.technology	Y	Y	Y	Y	Y	Y	Y	Y
3	普遍接受-测试.top	Y	Y	Y	B	B	B	Y	Y
4	ua-test.世界	Y	Y	Y	B	B	B	Y	Y
5	普遍接受-测试.世界	Y	Y	Y	B	B	B	Y	Y
6	普遍接受-测试。世界	Y	Y	A	A	B	A	A	Y
7	ua-test.xn--rhqv96g	Y	Y	Y	B	B	B	Y	Y
8	xn----f38am99bqvcd5liy1cxsg.top	Y	Y	Y	B	B	B	Y	Y
9	xn----f38am99bqvcd5liy1cxsg.xn--rhqv96g	Y	Y	Y	B	B	B	Y	Y
10	اختبار-القبولالعالمي.top	C	C	C	B + C	B + C	B + C	C	B
11	اختبار-القبولالعالمي.شبكة	Y	Y	Y	B + C	B + C	B + C	Y	Y
12	ua-test.link/我的页面	Y	Y	B	B	Y	B + E	Y	E
13	ua-test.technology/我的页面	Y	Y	B	B	Y	B + E	Y	E
14	普遍接受-测试.top/我的页面	Y	Y	B	B	B	B + E	Y	E
15	ua-test.世界/我的页面	Y	Y	B	B	B	B + E	Y	E
16	普遍接受-测试.世界/我的页面	Y	Y	B	B	B	B +E	Y	E
17	普遍接受-测试。世界/我的页面	Y	Y	A	A	B	A	A	E

Observations

- There was a significant variation for browsers running on different mobile platforms and between different browsers when running on mobile platforms (see later sections for details).
- On mobile platforms both Firefox and Opera display Punycode for Unicode URLs for many or all cases, with no obvious setting to change this.

Browser Results

Chrome

Desktop and mobile platforms

The only issue seen for Chrome was a failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar: The labels were displayed in the incorrect order

- HTTPS Handling

(Minor) When displaying a Unicode URL secured with HTTPS only the Punycode for the URL is visible in the dialogue displayed when clicking on the 'Green lock' icon.

Firefox

Desktop platforms

The only issue seen for Firefox on desktop was a mishandling of the Chinese 'open dot' character (URLs 6 and 17):
A Google search was performed instead of correctly processing the URL

Mobile platforms

Several issues were observed:

- Android
  - A mishandling of the Chinese 'open dot' character (URLs 6 and 17): A Google search was performed instead of correctly processing the URL
  - A failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar: The labels were displayed in the incorrect order
  - If the location bar is touched to edit the URL whilst the page for any Unicode URL (URLs 3-19) is loaded the URL path is displayed as code points not Unicode: Code points displayed in location bar
- iOS
  - Punycode was displayed for all Unicode URLs (URLs 3-19): Example - chinese unicode displayed as Punycode
  - A mishandling of the Chinese 'open dot' character (URLs 6 and17):
  - A Google search was performed instead of correctly processing the URL

- A failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar:
- The labels were displayed in the incorrect order

Opera

Desktop platforms

The only issue seen for Opera was a failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar:
The labels were displayed in the incorrect order

Mobile platforms

Several issues were observed:

- Android
  - Punycode was displayed for all Unicode URLs (URLs 3-19):
    Example - chinese unicode displayed as Punycode
  - A failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar:
    The labels were displayed in the incorrect order

- iOS
  - Punycode was displayed for all Unicode URLs (URLs 3-19):
    Example - chinese unicode displayed as Punycode
  - A mishandling of the Chinese 'open dot' character (URLs 6 and17):
    A Google search was performed instead of correctly processing the URL
  - A failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar:
    The labels were displayed in the incorrect order
  - If the location bar is touched to edit the URL whilst the page for any Unicode URL (URLs 3-19) is loaded the URL path is displayed as code points not Unicode:
    Code points displayed in location bar

- HTTPS Handling

(Minor) When displaying a Unicode URL secured with HTTPS only the Punycode for the URL is visible in the dialogue displayed when clicking on the 'Green lock' icon.

Safari

Desktop and mobile platforms

- A mishandling of the Chinese 'open dot' character (URLs 6 and17):
  A Google search was performed instead of correctly processing the URL
- A failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar:
  The labels were displayed in the incorrect order

- HTTPS Handling

(Minor) Desktop only: When displaying a Unicode URL secured with HTTPS only the Punycode for the URL is visible in the dialogue displayed when clicking on the 'Green lock' icon.

Edge

These results assume the relevant language packs are installed correctly.

Desktop platform only

- The only issue seen for Edge was a failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar:
  The labels were displayed in the incorrect order

Internet Explorer

These results assume the relevant language packs are installed correctly.

Desktop platform only

No issues were seen.

Samsung Browser

Native Android browser for a Samsung phone

Mobile platform only

- Failure to correctly display the mixed RTL.ASCII (URL 10) in the tab title bar:
  The labels were displayed in the incorrect order
- If the location bar is touched to edit the URL whilst the page for any Unicode URL (URLs 3-19) is loaded the URL path is displayed as code points not Unicode:
  Code points displayed in location bar

Vivaldi

Desktop platform only

- Punycode was displayed for all Unicode URLs (URLs 3-19):
  Example - chinese unicode displayed as Punycode
- A mishandling of the Chinese 'open dot' character (URLs 6 and17):
  A Google search was performed instead of correctly processing the URL

Conclusion

As the results indicated, all browsers – with the exception of Internet Explorer on desktop – showed certain issues resolving searches and displaying results properly. The findings indicate that while browser developers are making progress toward becoming UA-ready, there is still more work to do.

The UASG is sharing the results of this report with the companies measured, so that they have the opportunity to gauge their UA-readiness and connect with resources that can assist them in updating their systems.

Organizations or individuals that would like to learn more about Universal Acceptance or the UASG can find more information at uasg.tech.

https://drive.google.com/open?id=0B5gNT4RRJ0xPdl8zSkpha0tNX28

Appendix A: Discussion of Ideographic Full Stop

The ideographic full stop (U+3002 [。]) is used in languages such as Chinese or Japanese to mark the end of a sentence. UASG004 states "We expect software to transform the 'open dot' to a standard ASCII dot ".", thus making use of the already registered domain name."

However, the latest standards guidance is not crystal clear:

- The obsoleted RFC3490 (IDNA 2003) spec was clear:

  "Whenever dots are used as label separators, the following characters MUST be recognized as dots: U+002E (full stop), U+3002 (ideographic full stop), U+FF0E (fullwidth full stop), U+FF61 (halfwidth ideographic full stop)."

- The IDNA 2008 specs say nothing explicitly about this however RFC5895 says in Section 2.4:

  "[IDNA2008protocol] is specified such that the protocol acts on the individual labels of the domain name. If an implementation of this mapping is also performing the step of separation of the parts of a domain name into labels by using the FULL STOP character (U+002E), the IDEOGRAPHIC FULL STOP character (U+3002) can be mapped to the FULL STOP before label separation occurs. There are other characters that are used as "full stops" that one could consider mapping as label separators, but their use as such has not been investigated thoroughly. This step was chosen because some input mechanisms do not allow the user to easily enter proper label separators. Only the IDEOGRAPHIC FULL STOP character (U+3002) is added in this mapping because the authors have not fully investigated the applicability of other characters and the environments where they should and should not be considered domain name label separators."

  Note the use of 'can' instead of a RFC2119 keyword.

- The Unicode Technical Standard #46 (UTF#46) states:

  In this document, a label is a substring of a domain name. That substring is bounded on both sides by either the start or the end of the string, or any of the following characters, called label-separators:
  - U+002E (.) FULL STOP
  - U+FF0E (．) FULLWIDTH FULL STOP
  - U+3002 (。) IDEOGRAPHIC FULL STOP
  - U+FF61 (。) HALFWIDTH IDEOGRAPHIC FULL STOP

- No W3C document could be found that explicitly described the use of full stop characters in domain names (please update this if incorrect). Some documents mention its use in text and for layout (e.g. WD-clreq-20170220)

- It also appears that historically there was a decision by Mozilla to include the ideographic full stop in a list of blacklisted characters.