

Subject: Re: [UA-Measurement] [EXTERNAL] Re: Background on "Characterize how much Android platform limits acceptance of IDNs in web browsing" (FY23 M4)
Date: 4 August 2022 Thursday 23:12:32 GMT+03:00
From: UA-Measurement on behalf of Julien Bernard via UA-Measurement
To: Mark Svancarek (CELA), Mark Datysgeld, Jim DeLaHunt, ua-measurement@icann.org
Attachments: ATT00001.txt

PC browsers based on Chromium are also impacted.
In okhttp issue #7008 [1], author provided a link with relevant testing results [2].

[1] <https://github.com/square/okhttp/issues/7008> [github.com]
[2] <https://wpt.fyi/results/url/toascii.window.html?label=master&label=experimental&aligned> [wpt.fyi]

Regards,
Julien

On 19/07/2022 13:55, Mark Svancarek (CELA) via UA-Measurement wrote:

Chrome, Edge, opera and perhaps others are all built on the Chromium codebase. If there is an IDNA defect in Chrome for Android, I'd expect it to show up on the PC browsers as well. Do we have any info whether that is the case or not?

From: UA-Measurement <ua-measurement-bounces@icann.org> **On Behalf Of** Mark Datysgeld via UA-Measurement
Sent: Thursday, July 14, 2022 12:17 PM
To: Jim DeLaHunt <uag@jdlh.com>; ua-measurement@icann.org
Subject: [EXTERNAL] Re: [UA-Measurement] Background on "Characterize how much Android platform limits acceptance of IDNs in web browsing" (FY23 M4)

Very interesting points. This is maybe the sort of thing that should be coordinated with ICANN org. itself for outreach?

On 14/07/2022 04:45, Jim DeLaHunt via UA-Measurement wrote:

UA Measurement WG colleagues:

The agenda for our 14 July 2022 meeting includes a topic, "Characterize how much Android platform limits acceptance of IDNs in web browsing". This is FY23 goal M4. Here is some background on that topic. There are links which people can follow to find out more.

M4 is inspired by findings documented in UASG 037 *UA-Readiness of Some Programming Language Libraries and Frameworks*[1]. They report that the developer of a popular Android URL-handling library, **okHttp**, refused to fix a problem with handling IDNs, in part because better UA would make them behave differently than the leading Android browser, Chrome. Chrome follows the outdated IDNA2003 spec, which causes problems with some IDNs.

M4 reads, "Characterise how much Android platform limits acceptance of IDNs in web browsing, because they stay compatible with the leader, Chrome on Android, which uses an outdated IDN spec (IDNA2003)".

M4 studies the Android platform, its leading browser, Chrome, and leading

Android libraries and frameworks. It does not study browsers in general. It should answer the questions: how many Android app projects will make decisions like `okHttp`, and reject UA because it conflicts with Chrome on Android? Of the obstacles to UA on Android, how significant is the obstacle of developers rejecting UA in favour of Chrome? Is it one of the few greatest obstacles, or less significant? Regarding Chrome and the other builtin Android platform services, how many other significant UA obstacles exist due to choices by the Android project? How many kinds of IDNs are obstructed? Additionally, what does it take to change the behaviour of Chrome and the Android platform to handle IDNs in a way that provides the best Universal Acceptance? How difficult will it be to persuade the Android platform to make this change? What kind of evidence do we need to provide? And, when we succeed, what disruption will that change cause across the Android ecosystem?

I will link to some of the evidence in UASG037, and pointed to from UASG037.

UAS037, section **Android - Kotlin - okHttp (IDNA2008)**, p. 17:

`okHttp` is a very popular HTTP client in the Java and Android environments. However, it is only compatible with IDNA2003 as it relies on `java.net.IDN`. A bug report (`okhttp` #1615, [2]) was closed in 2020 showing they are not willing to support IDNA2008....

We created another bug report for IDNA2008 (`okhttp` #6910, [3]), providing a more appropriate way to solve the issue than the old one. The maintainer was responsive and willing to fix it, at least for Android that provides packages for IDNA directly in the SDK, but the fix broke some of their tests and the fact that IDNA2008 is not implemented[4] in the most used web browser (Chrome) made them stop implementing that support.

It is therefore highly recommended to make Google implement IDNA2008 instead of IDNA2003 in their products if we want to encourage the community to follow.

In `okhttp` #1615, **Consider upgrading `HttpUrl` to IDNA 2008** [2], the `okhttp` developer rejects the request which would promote UA, saying, "IDN hasn't taken over, we don't see enough usage to justify owning custom implementation, so closing. Following the JVM and Android platform. Won't fix." (2020-03-09).

In `okhttp` #6910, **Add support for IDNA 2008 (RFC 5891)** [3], the `okhttp` developer rejects the fix which would promote UA, asking, "How common are these domains? What are some examples that are currently broken?", and saying, "Going to close this out as the strictness of IDNA 2008 is likely to cause more visible issues than this solves, particularly as this isn't uniformly supported or implemented by clients and servers." (2021-11-15).

The Android platform's issue tracker has an issue #206015971, **Add support for IDNA 2008 (RFC 5891)** [5]. I believe this was submitted by the UASG vendor who wrote UASG037. The Google response is, "We've shared this with our product and engineering teams and will continue to provide updates as more information becomes available." (2021-11-14). There is no further response. The issue is neither accepted nor rejected.

Another `okhttp` issue, #7008, **Upgrade from IDN2003 to UTS #46 Non-transitional** [6], appears to be another look at the change. The writer's comment about IDNA2008 is, "a spec that never gained widespread adoption". I am concerned that this dismissive attitude is bad news for our attempts to persuade developers to support.

This is a good point to look at the Unicode Technical Standard #46, **Unicode IDNA Compatibility Processing** [7]. I would summarise it as the Unicode community's good-faith effort to help developers migrate from IDNA2003 to IDNA2008, and solve some technical problems caused by the transition.

In recent meetings, the Google paper **Internationalized Domain Names (IDN) in Google Chrome** [8] has come up. This strikes me as an interesting and worthwhile strategy about confusable IDNs, fraud, and protecting Chrome users. However, it does not seem to me to directly address the choice between staying with IDNA 2003 or moving to IDNA 2008. Thus it seems to me that it is a separate issue to M4.

- [1] <<https://uasg.tech/download/uasg-037-ua-readiness-of-some-programming-language-libraries-and-frameworks-en/>> [can01.safelinks.protection.outlook.com]
- [2] <<https://github.com/square/okhttp/issues/1615>> [can01.safelinks.protection.outlook.com]
- [3] <<https://github.com/square/okhttp/issues/6910>> [can01.safelinks.protection.outlook.com]
- [4] <https://en.wikipedia.org/wiki/Usage_share_of_web_browsers#Summary_tables> [can01.safelinks.protection.outlook.com]
- [5] <<https://issuetracker.google.com/issues/206015971?pli=1>> [can01.safelinks.protection.outlook.com]
- [6] <<https://github.com/square/okhttp/issues/7008>> [can01.safelinks.protection.outlook.com]
- [7] <<http://unicode.org/reports/tr46/>> [can01.safelinks.protection.outlook.com]
- [8] <[https://chromium.googlesource.com/chromium/src/+main/docs/idn.md](https://chromium.googlesource.com/chromium/src/+/main/docs/idn.md)> [can01.safelinks.protection.outlook.com]

M4 is valuable because it is an opportunity to engage with a specific, concrete example of the general supply-demand paradox which obstructs Universal Acceptance adoption. I believe these developers are good candidates to support universal acceptance, but they have technical and business objections. We have not yet succeeded in making good responses to those objections. I think we should. M4 gives us that opportunity.

I think it would be valuable to have an expert in UTS 46 and IDNA 2008 look at the technical issues, and provide advice on how solve the issues and improve universal acceptance.

Best regards,
—Jim

--

. --Jim DeLaHunt, jdlh@jdlh.com <http://blog.jdlh.com/> [can01.safelinks.protection.outlook.com]
multilingual websites consultant

2201-1000 Beach Ave, Vancouver BC V6E 4M2, Canada
Canada mobile +1-604-376-8953

UA-Measurement mailing list

UA-Measurement@icann.org

<https://mm.icann.org/mailman/listinfo/ua-measurement> [can01.safelinks.protection.outlook.com]

By submitting your personal data, you consent to the processing of your personal data for the purposes of this mailing list.

--

Mark W. Datysgeld [markwd.website@can01.safelinks.protection.outlook.com]

Director at Governance Primer [governanceprimer.com]

[can01.safelinks.protection.outlook.com]

ICANN GNSO Councilor



UA-Measurement mailing list

UA-Measurement@icann.org

<https://mm.icann.org/mailman/listinfo/ua-measurement>

By submitting your personal data, you consent to the processing of your personal data for purposes of subscribing to this mailing list in accordance with the ICANN Privacy Policy (<https://www.icann.org/privacy/policy>) and the website Terms of Service (<https://www.icann.org/privacy/tos>). You can visit the Mailman link above to change your membership status or configuration, including unsubscribing, setting digest-style delivery or disabling delivery altogether (e.g., for a vacation), and so on.