UA Measurement WG Meeting

20 October 2022

Attendees

Nabil Benamar
Harsha Wijayawardhana
Jim DeLaHunt
Bibek S, Kathmandu
Gopal Tadepalli
Yin May Oo
Seda Akbulut

Meeting Agenda:

1. Welcome and Roll Call
2. (M4) Characterize how much Android platform limits acceptance of IDNs in web browsing (UASG037)
   (Identify the UA related constraints on applications running on Android platform)
   a. How does UTS #46 differ from IDNA 2008?
3. AOB

Meeting recording:

https://icann.zoom.us/rec/share/nB4WIJBSjUVyP1NWq6Qbkb5icz02YfydTOn2pazDWF9EnMTO-wzaCmxGyjbMArgR.1E2m-mS0rE9Vg55V
Passcode: !pNve7U1Fh

Meeting Notes

In this meeting, only the M4 action item was partially discussed, and it will be continued in the next meeting.

The action plan from the previous meeting was to characterize how much Android platform limits acceptance of IDN in web browsing. It has been discussed on the UASG037 document where the feedback from the third-party library called okHttp (Kotlin-Android-okhtttp) rejected the UA-compliance to be in line with Android platform. This is the most important item to work on, and we need to explore the impacts. Jim said there was some discussion of this topic for the past months, but we have not picked it up for a while. Nabil invites Jim to give the idea of the current problems.

> https://mm.icann.org/pipermail/ua-measurement/2022-July/000600.html
>
> *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.

Jim shared the working group's mail-list-archive #M4 which will link to other messages to give the background. On the Android platform, various bits of Android generated software which were written by Google and the most important of those is the Chrome Browser. There are also third-party offered pieces of software and one of those is a library called "okHttp". We did an assessment of universal acceptance of some components on Android and we measured UA Http, and discovered that it had weaknesses that were related to it, treating internationalized domain names according to an obsolete specification IDNA2003 instead of IDNA2008.

Cofomo, our contractor, did a good job posting a bug report to okHttp's bug list to improve their software to work with IDNA2008, and got back a very interesting reply back from the developer saying "Sorry, we don't want to do that because Chrome on Android uses IDNA2003 and we want to be compatible with Chrome."

That leads to two questions:

1) **What is the difference between IDNA2003 and IDNA2008?**

We need to know whether there is a gap between IDNA2003 and IDNA2008, and which has worse universal acceptance. The challenge for the first question is to measure which URLs are treated differently depending on whether the software supports IDNA2008 or IDNA2003, and assess how bad the difference is.

2) **What do we do about it?**

We need to find a way to persuade Chrome to come up with the best standard which is IDNA2008.

The specification from the Unicode Consortium called UTS#46 is intended for any software that has started off using IDNA2003 moving to IDNA2008 with smoother transition for their users. It says tweak certain edge cases in a special way that will make the transition easier.

Jim thinks it is going to be helpful to **understand the difference between conforming to UTS#46 vs going directly to IDNA2008**. Maybe it **is worthwhile categorizing the UA-weaknesses of complying with UTS#46.**

There are three levels: Android, Chrome browser on Android, and various other software conforming to IDNA2003. Some software conforms to UTS#46 which is a transaction to IDNA2008. And some software conforms to IDNA2008, which we presently believe is what everybody should be using.

Jim explained the essence of the question in the agenda item: “What are the differences between the two lesser ones and the standard we desire?”

Nabil asked if this is the issue raised only to Chrome Browser or any other software or browser or other environment. Jim answered the focus according to the agenda is Chrome on Android, since Chrome is the official browser on Android environment and software like okHttp conforms to what it does, and it limits the entire Android platform.

How do these specifications differ, how does the software differ in terms of universal acceptance if they support wrong specification, instead of right spec? That applies to any platform but only Android where a major piece of software is holding the wrong specs and saying that is what we want to do and causing other pieces of software to also hold onto the wrong specs.

Android is a platform that we can put a boundary around and say the effects are going to be on this platform. That is the reason to focus on Android platform, but the technical questions about universal acceptance and those specifications do in fact reach to any platform.

Nabil asked about the Chrome browser on the iOS platform. Jim answered that he did not know. There are Chrome browser versions on Mac OS, Windows OS and so on to look at. Nabil understood that this is not limited to mobile platforms only, but this needs to be focused first.

Nabil asked where to start from. Jim said there are Technology WG tasks and Measurement WG tasks. As for a measurement task, think of the full range of URLs, which URLs are treated differently depending on whether you use IDNA2003 or UTS#46 or IDNA2008. And how many of these software on Android conforms to IDNA2003, or IDNA2008, etc. We get the idea of impact on all the Android platform users. By looking at the URLs we know, how big is the difference and how much should we be concerned by looking at the different pieces of software for those particular URLs, how widespread is the problem across the software.

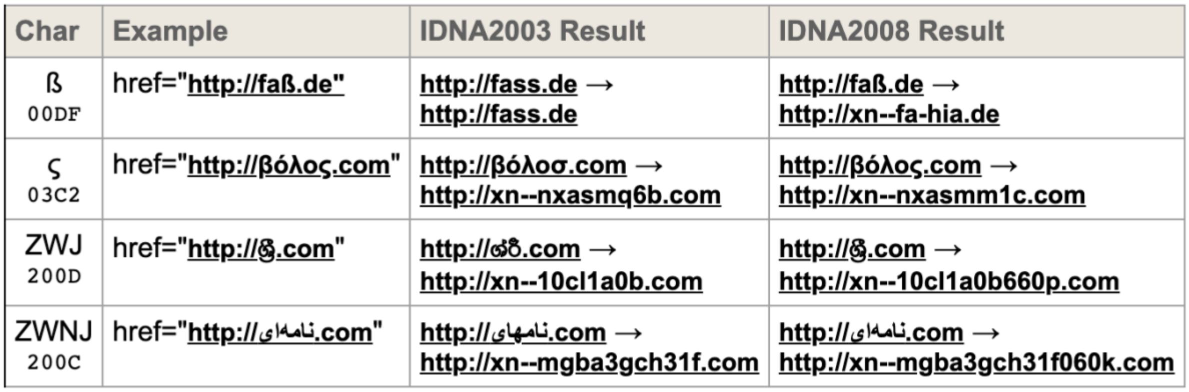
Jim suggested looking at Gopal’s contribution in the chat. Quote “Google Android has the URL-Character limit of 8192 characters for most of the apps including Google maps. This does not map linearly to the IDNs.”

Gopal said the upper bound of character count is not a serious concern right now, the question is more about transition, how smooth the transition be from Unicode compatibility to IDNA compatibility. Based on UTS#46, Gopal said the mapper gets stuck because of the deviation character Latin small letter Eszett (ß), that must be checked manually. Gopal’s approach is writing a lookup-table for some of the Indian languages where there is still a need to be added more later.

The concern is, even then, there is not zero-error. The lookup-table is for the characters which do not get automatically mapped. Jim explained pointing to the Table of Deviation in UTS#46 that the whole debate about IDNA2008 and UTS#46 comes down to these four characters:

- U+00DF (German esszett ß),
- U+03C2 (Greek sigma ς),
- U+200D (Zero Width Joiner, ZWJ),
- U+200C (Zero Width Nonjoiner, ZWNJ).

Table 1. Deviation Characters



Char	Example	IDNA2003 Result	IDNA2008 Result
ß 00DF	href="http://faß.de"	http://fass.de → http://fass.de	http://faß.de → http://xn--fa-hia.de
ς 03C2	href="http://βόλος.com"	http://βόλοσ.com → http://xn--nxasmq6b.com	http://βόλος.com → http://xn--nxasmm1c.com
ZWJ 200D	href="http://ශ්‍රී.com"	http://ශ්රී.com → http://xn--10cl1a0b.com	http://ශ්‍රී.com → http://xn--10cl1a0b660p.com
ZWNJ 200C	href="http://نامه‌ای.com"	http://نامهای.com → http://xn--mgba3gch31f.com	http://نامه‌ای.com → http://xn--mgba3gch31f060k.com

When Jim had conversations with North Americans and Europeans about these characters, they suggested the **first two may be common characters but do not appear in domain names very often.**

Harsha chimed in and said some characters were not included in the IDNA2003. IDNA is important for complex scripts. He emphasized that the ZWJ (U+200D) is important for Sri Lanka script, and it should not be ignored.

Jim said this is a valuable matter for people from North America and Europe to note for the scripts such as Sinhala, for which the ZWJ is very important. For some scripts ZWJ or ZWNJ is very important and for some is not. What would be useful for the Measurement WG is to make examples of how ZWJ and ZWNJ would appear in TLDs, SLDs.

Harsha pointed out that ZWJ and ZWNJ are not allowed for TLD to prevent phishing attacks. Jim asked if it is still debatable whether to use ZWJ. Harsha said although IDNA2008 allows ZW characters, Sinhala Generation Panel decided not to allow the ZWJ and some characters in the second level domain level as well. However, Harsha would like to know if this could be still added, especially for the email address local part. There is a possibility that in future there may be a need to use the labels with ZWJ for Sinhala. Harsha also added that the Facebook desktop application removes the ZWJ, which makes Sinhalese word “Sri” (ශ්‍රී , U+0DC1 U+0DCA U+200D U+0DBB U+0DD3) rendered in broken form (ශ්රී , U+0DC1 U+0DCA U+0DBB U+0DD3).

Jim suggested making a list of how this policy affects domain names.

Harsha explained where the problem lies, creating email addresses, rendering the labels on different social-media platforms, applications which remove ZW characters completely and so on.

Jim suggested **writing down the evaluative information in a way which reflects the concerns of each language and that is technically understandable** so that people who do not have knowledge of the scripts also can participate in problem solving.

Jim asked to confirm since ZWJ is disallowed by the Sinhala GP (although it is required to write the Sinhalese word ශ්‍රී correctly) for both TLD and SLD, there is no technical conflict of displaying Sinhalese labels whether by IDNA2003 or IDNA2008 or UTS#46. Harsha said that the GP had to reluctantly block ZWJ because they thought there was no other way to solve the problem of misusing the ZWJ in the labels. Currently, the IDN-ccTLD is just “.ලංකා” (.Lenka) without the “Sri”. It is confirmed that ZWJ is disallowed for Sinhalese TLD, however, it might still have adjustments for Sinhalese SLD. There is a radio station name with “Sri” in it, so it might be needed in the future.

Jim suggested tracing back to the agenda item of how much impact will there be if Android platform continues imposing IDNA2003. He thinks it has zero impact on current Sinhalese TLDs. He asked if the conclusion could be regardless of IDNA2008 or UTS#46 or IDNA2003, Sinhalese labels would be displayed the same since technical precision would be very important as a product from this WG.

Regarding the unavailable TLD labels in IDNA2003, Jim clarified by checking the specification on RFC3490, 5890, and 5891. Jim suggested stating all the conditions of how labels are affected on different environments which follows each standard, since the technical precision of this WG's output would be very helpful.

Jim said this was a good conversation where he was allowed to state what he thinks and learn more about problems in other scripts as well.

Nabil said if this conversation is about just checking with one language for the moment.

Harsha asked about German Eszett and Greek Sigma, Jim said they are fine as these are visible characters. Seda pointed out that there is a Latin Small Letter a with Diaeresis (ä, U+00A4) and also (small letter a+U0308). Jim clarified that both forms are normalized into one predicting so there is no issue about this in IDNA2003 or IDNA2008. But **when it comes to email IDs, that can be an issue.** So, the only problem is with the complex script group where there is a ZW character issue for IDN and email ID. **However, the email address issue may be out of scope for the measurement WG, but may be related to EAI WG scope.**

Harsha will see if those blocked characters can be used in email ID's to represent the names. Harsha will look into this and get back to the WG with more information.

Jim expressed concern with three different things that have three different sets of rules:

1) The top level and the second level domain names where ICANN or registries make the rule.
2) The lower-level domain names where an individual system operator makes their own rule (they could feel free to ignore the LGR that ICANN proposed. In this area, users might be able to use labels with even restricted characters to create a domain name.)
3) The mailbox part of the email addresses before the '@' sign

If we are going to report the issues on IDNA2003 vs IDNA2008, we should consider all the three concerns, the mailbox name part of the email addresses might be more related to the EAI WG.

It was agreed that the topic needs to be discussed more in detail to find out the objective of this work, and what our product should be.

Yin May asked about the combination of “Sri” characters. Jim responded with the following details in the chat:

Jim DeLaHunt, Vancouver, Canada to Everyone (19:58)
Sri: ශ්‍රී -> Codepoints
ශ්‍රී ලංකාව " =
0DC1 SINHALA LETTER TAALUJA SAYANNA
0DCA SINHALA SIGN AL-LAKUNA
200D ZERO WIDTH JOINER
0DBB SINHALA LETTER RAYANNA
0DD3 SINHALA VOWEL SIGN DIGA IS-PILLA
0020 SPACE
0DBD SINHALA LETTER DANTAJA LAYANNA
0D82 SINHALA SIGN ANUSVARAYA
0D9A SINHALA LETTER ALPAPRAANA KAYANNA
0DCF SINHALA VOWEL SIGN AELA-PILLA
0DC0 SINHALA LETTER VAYANNA

Jim DeLaHunt, Vancouver, Canada to Everyone (20:00)
I used this app to break characters by Unicode code point and name:
https://r12a.github.io/uniview/?charlist=%C3%A2%E2%82%AC%C2%A6

Other ref: https://unicode.org/reports/tr46/

Other comments in the chat:
Gopal Tadepalli to Everyone (19:48)
Language experts must "understand" and "speak" both the language been translated to and verse-visa. This traditional approach used for solving the problem of language differences has not been productive and favourable. Suggestion: Get closer to the machine with UNICODE & Generator Rules. Like in real life there are restrictions on certain words @ Registration. This may have to be given a thought.

Next meeting: Thursday 03 November 2022 UTC 1600-1700

Action items

No.	Action Item	Owner
1	To confirm if Sinhala GP can allow ZWJ for SLD, and email IDs	Harsha
2	Continue the discussion to have consensus on the objective of this work, and what our product should be.	Measurement WG