

# UA Technology WG Meeting

20 December 2021

## Attendees

Satish Babu

Julian Bernard

Jim DeLaHunt

Marc Blanchet

Guillaume Blanchet

Vadim Mikhaylov

Sarmad Hussain

## Agenda

1. Welcome and Roll Call
2. Overview of the Phase 3 Report on Programming Languages
3. AOB

## Meeting Notes

The meeting focused on the ongoing work about the 3<sup>rd</sup> phase of UA readiness evaluation of the programming languages and development frameworks. This work was initiated as a result of [SOW](#) that the Technology WG developed.

Julien shared a short summary of Phase 1 and 2, that covered the testing of 22 libraries in C, C#, Go, Java, Javascript, Python3 and Rust only on the Linux platform. Phase 3 is to add 18 libraries in PHP, Swift and Kotlin for Linux, Windows, iOS and Android platforms. Test framework was inherited from Phase 1. It covers 4 datasets updated with all instances in UASG004A. He also presented a short summary of code development. Some programs are command line apps and others are mobile apps.

Guillaume presented results of benchmarks for all libraries. For IOS Swift, URLSession and Alamofire are not UA ready. MessageUI and IDNA-Cocoa are UA ready, but developers need to be careful.

In PHP results for Windows 10 and Linux, Native mail for windows, Windows cURL PHP extension and PHP mailer are not UA ready. Linux cURL PHP extension, Email Validator, Guzzle, Symphony (Polyfill-intl-idn) and Intl are UA ready. Symphony

(http-client) is UA ready, but developers need to be careful as it does not use all flags for IDN conversion to A-Label.

As a result of the test on Android Kotlin, okhttp, HttpURLConnection, Retrofit, Fuel, Volley, Apache HttpClient are found as not UA ready. Jakarta Mail and Email Intent are UA ready but considers some scripts invalid. So, developers need to be careful.

Julian shared that for 15 bug reports, 4 code fixes have been submitted. Three out of four have been merged by the maintainers. He shared the format for future bug reports that it should be clear, precise and must suggest ways to resolve it as well. If possible, also provide code for solution in the form of a patch. He shared a Bug Reports table.

### Bug reports

Language	Platform	Library	Resolution
Swift	iOS	MessageUI	No answer yet
Swift	iOS	URLSession & Alamofire	Being discussed & maintainer assigned to the issue
PHP	Windows	mail	This does not claim to support RFC6531, this would be a new feature
PHP	Windows	cURL	This is a Windows issue
PHP	Windows & Linux	intl	This is a ICU issue
PHP	Windows & Linux	PHP Mailer	<b>Fix submitted and merged upstream</b>
PHP	Windows & Linux	Symfony HttpClient	<b>Fix submitted and merged upstream</b>
PHP	Windows & Linux	Symfony Mailer	3 different bug reports: <ul style="list-style-type: none"> <li>• <b>Fix submitted and merged upstream</b></li> <li>• Being discussed</li> <li>• No answer yet</li> </ul>
Kotlin	Android	Jakarta Mail	No answer yet
Kotlin	Android	HttpURLConnection	Transmitted to the engineering teams
Kotlin	Android	OkHttp	Closed after Q&A as IDNA 2008 is not uniformly supported or implemented by clients and servers
Kotlin	Android	Fuel	<b>Fix submitted and pending to be merged upstream</b>
Kotlin	Android	Apache HttpClient	No answer yet

Conclusion was provided for UA not ready and ready with precaution cases for Windows, Android, IOS and bug reports. Providing code patches while reporting bugs can speed up the process. Resources for the whole project progress and bug reports for each platform are at the following link:

<https://cofomo.github.io/universal-acceptance/>

One maintainer commented that “...as the strictness of IDNA2008 is likely to cause more visible issues than this solves, particularly as it is not uniformly supported or implemented by clients and servers”. Google Chrome is compliant with IDNA2003

and not with IDNA2008. So, some domains won't give the same page with Chrome (IDNA2003) and Firefox (IDNA2008).

Jim mentioned that Apple also did half the way by enabling to send and receive EAI, but they didn't follow through to make the corresponding programmatic interface work. As UASG we should inform Apple about this remaining work.

Sarmad mentioned that the efforts go beyond testing, and now we start to get bug reports for the first time. We no longer do just the gap analysis, but also we will have bug reports as a countermeasure. He shared [UTS #46 Unicode IDNA Compatibility Processing](#) report. Marc mentioned that IETF and Unicode have a conflict over using IDNA2008.

Julian mentioned that there is a test case in "PHP – intl", where the maintainer responded as "[IDNA2008 is implemented according to UTS#46](#)". It is not compatible with IDNA2008 and breaks. So, they are not advised to use it. People who work in internationalization are closer to UTS46 than they are to IDNA2008.

The work has been appreciated by the working group. It revealed many new things. Some constructive feedbacks are provided as follows:

- The "Bug Report" table should explicitly state all the actions taken and the results. Extra bug reports and discoveries must be mentioned in the report to make it more visible. (e.g Swift bug database does not have a link because it is private.) In the summary table, these findings should be included.
- The discovery about the contradiction between IDNA2008 and IDNA2003 should be made more visible in the report.
- Julian mentioned a mistake in the presentation that Symphony (mailer) should be red instead of green as it is not UA ready currently. He will update the presentation.

Sarmad shared the report with Technology Working Group via mailing list and requested them to review it and provide feedback.

**The agenda of the next week** was identified as reviewing the [T5 – UA challenges and survey\[docs.google.com\]](#).

**Next meeting:** (may change) Monday 03 January 2022 UTC 1600-1700

## Action items

<b>No.</b>	<b>Action Item</b>	<b>Owner</b>
1	Review Report and share feedback	All
2	Deciding on the next meeting date (3 Jan or 17 Jan)	Satish / All