Universal Acceptance (UA) Readiness Evaluation of Programming Languages and Development Frameworks – Phase 3

Cofomo / December 20, 2021

Universal Acceptance

First & second phases

- 22 libraries
- Only on Linux

Language	Platform	Framework/Library
C	Linux	libcurl
C	Linux	libidn2
C#	Linux	Mailkit
C#	Linux	Microsoft
Go	Linux	Idna
Go	Linux	Mail
Go	Linux	Smtp
Java	Linux	Commons-Validator
Java	Linux	Guava
Java	Linux	ICU
Java	Linux	Javamail/JakartaMail
Java	Linux	JRE
Javascript	Linux	Idna-uts46
Javascript	Linux	Nodemailer
Javascript	Linux	Validator
Python3	Linux	Django_auth
Python3	Linux	Email_Validator
Python3	Linux	Encodings_Idna
Python3	Linux	Idna
Python3	Linux	Smtplib
Rust	Linux	Idna
Rust	Linux	Lettre

This third phase

- This third phase add 18 libraries to the set;
- Target 4 platforms:
  - Linux
  - Windows
  - IOS
  - Android
- Test framework inherited from the first phase

Language	Platform	Framework/Library
PHP	Linux	cURL
PHP	Linux	emailValidator
PHP	Linux	Guzzle
PHP	Linux	intl
PHP	Linux	mailer
PHP	Linux	symfony
PHP	Windows	cURL
PHP	Windows	emailValidator
PHP	Windows	Guzzle
PHP	Windows	intl
PHP	Windows	mailer
PHP	Windows	symfony
PHP	Windows	mail (native)
Swift	iOS	AlamoFire
Swift	iOS	URLSession
Swift	iOS	MessageUI
Kotlin	Android	Apache
Kotlin	Android	EmailIntent
Kotlin	Android	Fuel
Kotlin	Android	HttpUrlConnection
Kotlin	Android	JakartaMail
Kotlin	Android	okHttp
Kotlin	Android	retrofit
Kotlin	Android	volley

Methodology

4 Datasets (Updated with all instances in [UASG004A](#))

* H_DNS: syntactic check on a domain name (IDNA2008)
* H_ES: syntactic check on an email address (EAI)
* L_A2U: Conversion from A-LABEL to U-LABEL (IDNA2008)
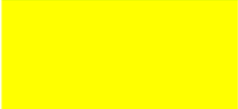* L_U2A: Conversion from U-LABEL to A-LABEL (IDNA2008)

Code

* Each library is tested upon its correspondent dataset with a pytest runner triggering a program written in the tested language;

* Some programs are command line apps others are mobile apps

* For mobile apps, a ui-crawler is sometimes necessary to simulate a real user tapping the interface

Two result formats

General / Brief:

Detailed:

▲ Result	▼ Test	Input	Outcome	
Failed *(show details)*	HEs - HESG1	info@ua-test.भारत	The ['info@ua-test.भारत'] is marked by the tested library as invalid even if it is valid	
Failed *(show details)*	HEs - HESS2	"i@fo"@ua-test.link	The ['"i@fo"@ua-test.link'] is marked by the tested library as invalid even if it is valid	
Passed *(show details)*	HEs - HESG1	info@ua-test.link	The ['info@ua-test.link'] is marked as valid and sendable	

IOS Swift Results

* **MessageUI** ▮ : MFMailComposeViewController not compliant. The developer needs to send the user through the iOS native Mail app by creating a mailto: url. (EAI)

* **URLSession**▮ : Not compliant (IDNA2008)

* **Alamofire** ▮ : Not compliant (IDNA2008)

* **IDNA-Cocoa** ▮ : Compliant. Ensure to take master branch (sha=870ba3e) for IDNA2008 (released versions are only IDNA2003 compatible) (IDNA2008)

Windows 10 & Linux PHP Results

* Native **mail** (Windows only) 🟥 : Not compliant. Does not send SMTPUTF8 (EAI);

* **Windows cURL PHP extension** 🟥: Use windows.h IdnToUnicode/IdnToAscii method which are based on IDNA2003.

* **Linux cURL PHP extension** 🟩: Use libidn2 which is fully IDNA2008 compliant (see our second phase report).

* **EmailValidator** 🟩 : Email validation compliant (EAI);

* **Guzzle** 🟩 : Http client fully compliant (IDNA2008);

* **Intl** 🟩: Disallowed characters sometimes not detected; compliant (IDNA2008). Ensure you use the right flag when calling it:

```
idn_to_utf8($url, IDNA_DEFAULT
        | IDNA_USE_STD3_RULES
        | IDNA_CHECK_BIDI
        | IDNA_CHECK_CONTEXTJ
        | IDNA_NONTRANSITIONAL_TO_UNICODE,
    INTL_IDNA_VARIANT_UTS46, $idnaInfo);
idn_to_ascii ($url, IDNA_DEFAULT
        | IDNA_USE_STD3_RULES
        | IDNA_CHECK_BIDI
        | IDNA_CHECK_CONTEXTJ
        | IDNA_NONTRANSITIONAL_TO_ASCII,
    INTL_IDNA_VARIANT_UTS46, $idnaInfo);
```

Windows 10 & Linux PHP Results (continue)

* **PHPMailer** 🟥 : Not compliant: rejects unicode (EAI);

* **Symphony** (http-client) 🟨 : Compliant but not using all flags for IDN conversion to A-Label (IDNA2008);

* **Symphony** (Polyfill-intl-idn) 🟩 : Compliant (IDNA2008);

* **Symphony** (mailer) 🟥 : Not compliant: does not send SMTPUTF8 flag (EAI);

Android Kotlin results

* **okHttp** 🟥 : Not compliant (IDNA2008). Use java.net.IDN that use ICU4J but with the deprecated methods, thus supporting IDNA2003;

* **HttpUrlConnection** 🟥 Not compliant (IDNA2008). Implemented using okHttp;

* **Retrofit** 🟥 : Not compliant (IDNA2008). Based on okHttp;

* **Fuel** 🟥 : Not compliant (IDNA2008). Uses HttpUrlConnection;

* **Volley** 🟥 : Not compliant (IDNA2008). Uses HttpUrlConnection;

* **Apache HttpClient** 🟥 : Not compliant (IDNA2008);

* **Jakarta Mail** 🟨 : Compliant but consider some scripts invalid (EAI).

* **Email Intent** 🟨 : Provide data as-is to a mail client (EAI)

Bug reports

* 15 bug reports
* 4 code fixes have been submitted
    * <u>3 merged upstream</u> by the maintainers
    * 1 pending to be merged upstream
* Key elements to write a bug report:
    * Write clear and precise report,
    * Suggest ways to resolve the problem whenever possible,
    * Follow the rules of the project.
* Watch and provide answers in the bug report
* If possible, provide new code to solve the issue in the form of a patch in a pull request

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	**Fix submitted and merged upstream**
PHP	Windows & Linux	Symfony HttpClient	**Fix submitted and merged upstream**
PHP	Windows & Linux	Symfony Mailer	3 different bug reports: • **Fix submitted and merged upstream** • Being discussed • No answer yet
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	**Fix submitted and pending to be merged upstream**
Kotlin	Android	Apache HttpClient	No answer yet

Windows – Note

- During testing, all our Windows batch files were failing to pass idn domains to the PHP cmdline apps. We realized an obscure setting should be used to enable the UTF-8 support:
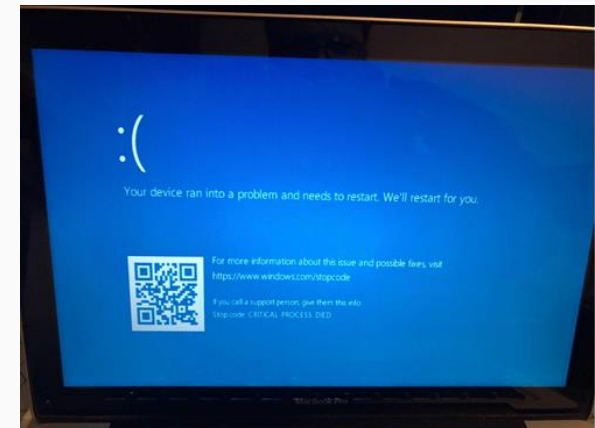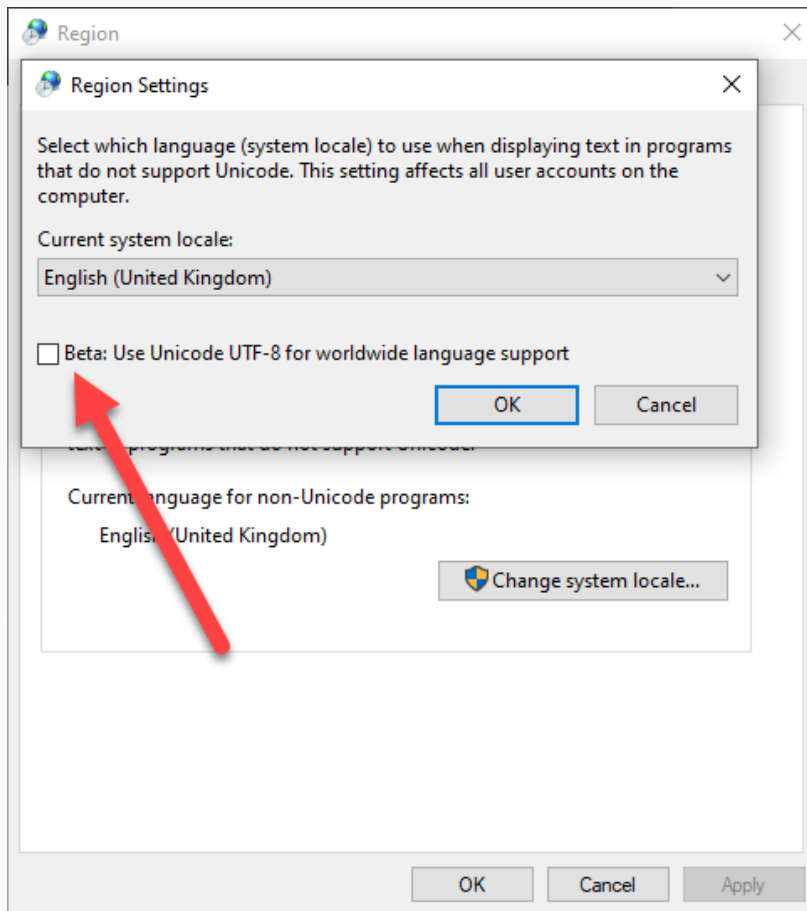
IOS - Conclusion

- Developer tools (MessageUI-MFMailComposeViewController, URLComponents, URL) provided by Apple has a poorer compliance than their end-user application (Safari, Mail). When possible, sending the user through an Apple native app is better than opening modals inside the application.
- Alamofire & URLSession are based on the same bogus Foundation framework

Windows - Conclusion

- Despite, the effort made in the Microsoft .NET Framework to support IDNA2008, many lib like cURL rely on the C windows.h file which provide a "IdnToUnicode/IdnToAscii" which is IDNA2003 compliant only;
- This cause cURL to be only compliant with IDNA2003 on windows and all extension of cURL, like the PHP ones or all the C lib using windows.h;

Android– Conclusion

- Most http libs are based on the Android stack. We provided simple suggestions to the community that could fix most of the problems we saw: https://github.com/square/okhttp/issues/6910 and https://issuetracker.google.com/issues/206015971
- Jakarta is widely used and EAI compliant, but Android developers will probably use Email Intent instead and delegate the responsibility of sending and validating the email to another end-user app.

Bug reports – Conclusion

* Some successful fixes have been approved and included in the libraries

* Providing code patches significantly increases the likelihood of a fix to be included in the source code

* Demonstration of both technical competency on the issue (i.e. UA) and on the software being patched gives more chances to be answered

Resources

- All resources, bug reports, detailed results and the likes are available at this url:

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