



# Universal Acceptance- Readiness Report FY23

26 September 2023



## TABLE OF CONTENTS

<b>EXECUTIVE SUMMARY</b>	<b>4</b>
<b>1. UNIVERSAL ACCEPTANCE FOR DIGITAL INCLUSION</b>	<b>4</b>
<b>2. OUR COMMITMENT</b>	<b>5</b>
<b>3. UNIVERSAL ACCEPTANCE STEERING GROUP (UASG)</b>	<b>6</b>
3.1 OVERVIEW	6
<b>4. THE PROCESS FOR ACHIEVING UA</b>	<b>8</b>
<b>5. UA-READINESS SCOPE AND EVALUATION FRAMEWORK</b>	<b>9</b>
5.1 LAYERS OF TECHNOLOGY RELEVANT FOR UA TESTING	9
5.2 UA TESTING FRAMEWORK	11
5.3 EAI SELF-CERTIFICATION PROGRAM	13
5.4 MAKE SYSTEMS UA-READY WITH FREE CODE SAMPLES	14
<b>6. UA-READINESS GAP ANALYSIS</b>	<b>16</b>
6.1 UA-READINESS OF PROGRAMMING LANGUAGES AND FRAMEWORKS	16
6.2 UA-READINESS EVALUATION OF STANDARDS AND BEST PRACTICES	19
6.3 UA-READINESS OF NETWORKING COMMAND LINE TOOLS	20
6.4 UA-READINESS OF CONTENT MANAGEMENT SYSTEMS	21
6.5 EAI-READINESS OF POPULAR WEB HOSTING TOOLS	22
6.6 UA-READINESS OF POPULAR WEB BROWSERS	23
6.7 UNIVERSAL ACCEPTANCE OF SOCIAL MEDIA PLATFORMS	25
6.8 EMAIL ACCEPTANCE BY WEBSITES	27
6.9 UA-READINESS OF OPEN-SOURCE SYSTEMS	28
6.10 UA-READINESS OF POPULAR IDENTITY PLATFORMS	29
6.11 EAI SUPPORT OF MAJOR EMAIL SOFTWARE AND SERVICES	30
6.12 EAI-READINESS OF EMAIL SERVERS	31
6.13 UA-READINESS OF REGISTRY AND REGISTRAR SYSTEMS	33
6.13.1 UA-READINESS SURVEY REPORT – ccTLDs	34
6.13.2 UA-READINESS OF CONTRACTED PARTIES AND REGISTRY SERVICE PROVIDERS	35
<b>7. GUIDANCE, TRAINING MATERIALS, AND REMEDIATION FOR UA</b>	<b>36</b>
7.1 UA-READINESS FRAMEWORK	36
7.2 NAMING INTERNATIONALIZED EMAIL MAILBOXES	37
7.3 ROLE OF CCTLDs IN ACHIEVING UA-READINESS	37
7.4 ACHIEVING UA-READINESS IN REGISTRY AND REGISTRAR SYSTEMS	38
7.5 CASE STUDIES ON UA ADOPTION	39
7.6 UA TRAINING MATERIALS	40
7.6.1 UA TECHNICAL TRAINING	40
7.6.2 UA COURSE ON ICANN LEARN	41
7.7 UA REMEDIATION EFFORTS	41
7.7.1 INTERACTING WITH TOOL AND FRAMEWORK PROVIDERS	41



7.7.2 INTERACTING WITH END-USER APPLICATION SERVICE PROVIDERS	44
7.7.3 INTERACTING ON DEVELOPER NETWORKS	45
<b>8. UA LOCAL INITIATIVES</b>	<b>48</b>
<b>9. LOOKING AHEAD</b>	<b>50</b>
<b>10. ENGAGE WITH THE UASG</b>	<b>51</b>



## EXECUTIVE SUMMARY

Since 2015, the Internet Corporation for Assigned Names and Numbers ([ICANN](#)) and volunteer-led Universal Acceptance Steering Group ([UASG](#)) have been working together to promote the Universal Acceptance (UA) of all valid domain names and email addresses. A large part of this work includes measuring the UA-readiness of various Internet applications, technology, and platforms to determine an overall state of UA-readiness that can help direct efforts. The Universal Acceptance-Readiness Report FY23 highlights:

- A summary of the scope of UA.
- Documentation of current gaps in technology based on UA-readiness testing of programming languages, email tools, identity platforms, web hosting tools, websites, and more.
- An update on the remediation activities undertaken in FY23.

The information provided in the report can help determine whether your systems are UA-ready and provide the current resources and training materials to address and fix UA-related issues. This report is intended for the entire software development community, public sector, Domain Name System (DNS) industry, and anyone looking to make their systems UA-ready.

### 1. UNIVERSAL ACCEPTANCE FOR DIGITAL INCLUSION

Domain names, like `example.com`, provide Internet users with the opportunity to create a unique online identity of their choice. The Internet Corporation for Assigned Names and Numbers (ICANN) has supported the work of the ICANN community to increase the options for such identifiers through the expansion of the DNS.

In addition to the earlier generic top-level domains (gTLDs) such as `.com` and `.org`, more than 1,200 new gTLDs are now available for consumers through the New gTLD Program, which is managed by ICANN. Some of these gTLD are longer than three characters (long TLDs). Individuals, businesses, and organizations are now able to choose gTLDs that represent a specific geography, profession, interest, community, and more, including `.cooking`, `.dentist`, `.fashion`, `.international`, `.istanbul`, just to name a few. These gTLDs provide global consumers and end users more choice for forming domain names while also increasing competition in the domain name industry.

The expansion of the DNS has also made the Internet more multilingual and inclusive of communities around the world, as domain names are also available in several different languages and scripts. These are called Internationalized Domain Names (IDNs). More than 150 IDN TLDs, including both generic and country code TLDs, have been delegated, including `.موقع` (`.site` in Arabic), `.онлайн` (`.online` in Russian), `.みんな` (`.everyone` in Japanese), `.ලංකා` (`.lanka` in Sinhala), `.ລາວ` (`.lao` in Lao), and `.中国` (`.china` in Chinese), and more.

Despite the availability of domain names and their corresponding email addresses in local languages and scripts, their acceptance (i.e., usability) is not well supported by current software and applications. These include many programming languages, email tools, online applications, social media applications, content management systems, and others. This presents a significant gap in current technology to allow global users to meaningfully access online resources and communicate effectively. It is also an opportunity for the public and private sectors to upgrade their applications and systems to accept these domain names and email addresses to give Internet users real choice and representation online, while enabling a multilingual Internet.



UA is the state in which all valid domain names and email addresses, including IDNs and long TLDs, are accepted and treated consistently by all software and applications. Achieving UA ensures that Internet users can effectively navigate and communicate online using a chosen domain name and email address that best aligns with their interests, business, culture, language, and script. Therefore, UA is a foundational requirement for achieving meaningful online access by enabling consumer choice and digital inclusion.

## 2. OUR COMMITMENT

Anil Kumar Jain  
Chair, Universal Acceptance Steering Group (UASG)



“Thanks in part to the first annual UA Day held in 2023, the UASG has seen increased growth and interest from various stakeholders worldwide. The UASG will build on this momentum as our work shifts from awareness toward UA remediation and adoption. The UASG is developing a strategic plan for the next five years that will guide and focus our efforts, which include helping organizations tackle underlying technical issues and gaps, hosting UA trainings for software developers and system administrators, and providing resources to help address barriers to adoption of UA.”

Edmon Chung  
Chair, ICANN Board IDN-UA Working Group



“Universal Acceptance is the foundation to a digitally inclusive and multilingual Internet. As ICANN works towards opening the next round of the New gTLD Program, ensuring Universal Acceptance of valid domains and email addresses by Internet applications and systems is critical to maintaining one unfragmented Internet. The ICANN Board is committed to supporting the UASG in its efforts and together we can make the Internet more accessible for global users and the next billion to come online.”

Sally Costerton  
ICANN, Sr. Advisor to President and SVP, Global Stakeholder Engagement & Interim President and CEO



“The UASG has made great strides in spreading UA awareness around the world, and the success of the first UA Day in 2023 is a tremendous testament to that. ICANN org now looks forward to strengthening our collaboration with the UASG and focusing on providing the tools necessary for organizations to become UA-ready so that we can achieve concrete UA adoption worldwide.”



### 3. UNIVERSAL ACCEPTANCE STEERING GROUP (UASG)

In 2015, the ICANN community formed the [Universal Acceptance Steering Group](#) (UASG), a volunteer-led organization supported by ICANN org, that aims to promote UA awareness and increase technical knowledge to address UA remediation and promote UA adoption. To address the multifaceted challenges of UA, the UASG has identified multiple stakeholders that are critical to UA adoption: technology developers, email tool and service providers, the public sector, TLD registries and registrars, and academia.

The UASG has organized itself into seven working groups to help engage with various stakeholders and tackle the different stages in the UA remediation process. The UA Measurement Working Group (WG) focuses on identifying relevant technology and undertaking gap analyses to identify the extent and nature of the UA issues in them. The UA Technology and Email Address Internationalization (EAI) WGs develop technical recommendations, determine remediation measures, and develop training materials. The UA Communications WG plans and drafts messaging that is used in outreach to relevant stakeholders. The UA Local Initiatives and UA Ambassadors raise awareness, train, and motivate local stakeholders to update their technology to be UA-ready. The UA Day Working Group was formed in the first half of FY23 to launch UA Day. This working group aims to strategize, plan, and conduct annual UA Day events in collaboration with ICANN org and key stakeholders.

The vision of the UASG is:

**All valid domain names and all email addresses work in all software applications.**

The mission of the UASG is:

**To mobilize software application developers to get their products UA-ready by providing encouragement, documentation, case studies, tools, and measures to deliver the right user experience for the end user.**

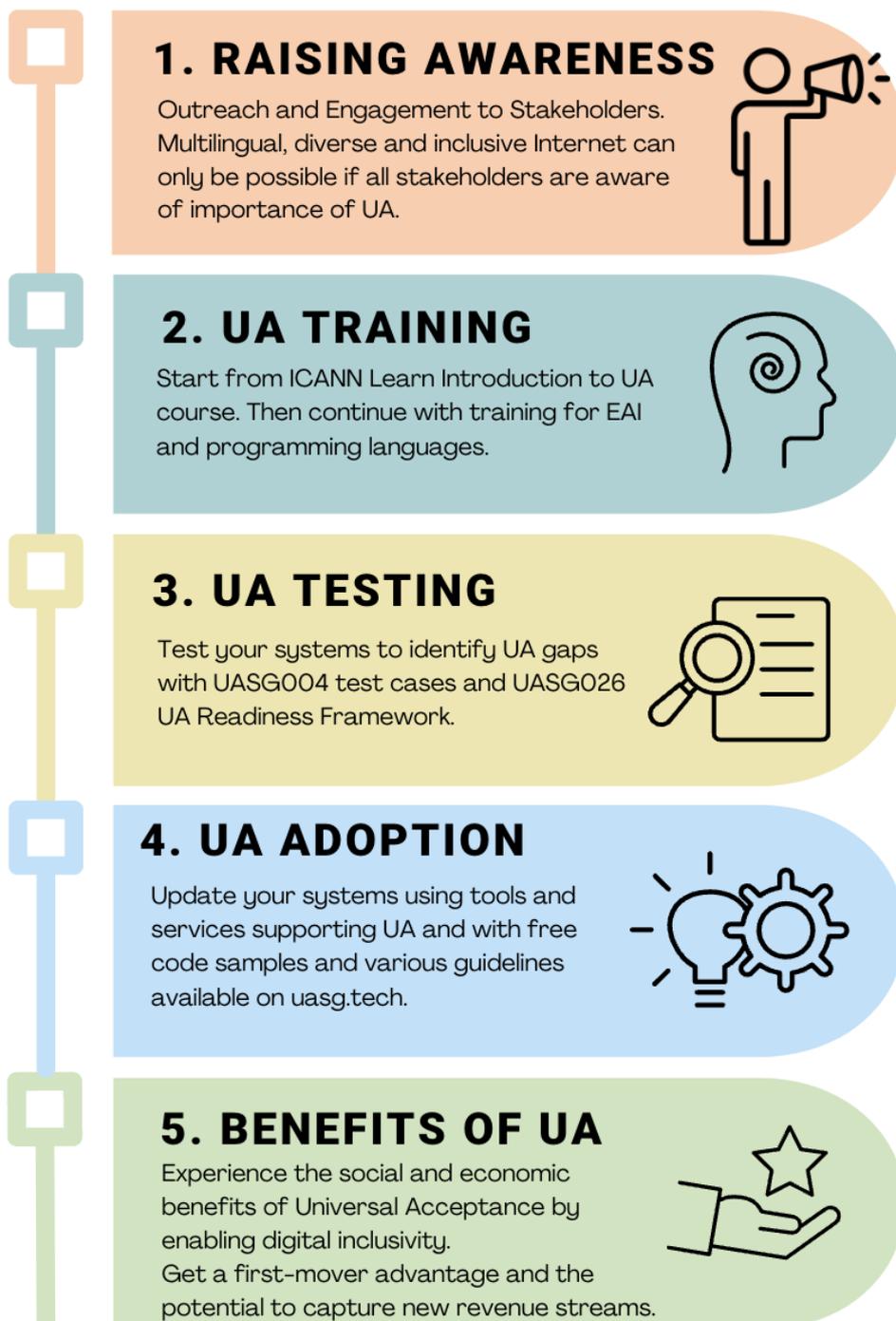
#### 3.1 OVERVIEW

This report, which builds on the [UA-Readiness Report for FY22](#), summarizes the scope of UA, documents current gaps in technology related to UA-readiness, and provides an update on the remediation activities undertaken by the UASG in FY23 (July 2022 – June 2023). It also aims to identify materials to achieve UA-readiness.

To help reach a global audience, the UASG has published several UA-related reports and resources in different languages available at [uasg.tech](#). The UASG also publishes [blogs and announcements](#) showcasing UASG efforts, events, and achievements. More detailed information on how the UASG plans and organizes its work for the future is available in its [FY24 Action Plan](#).



# UA IMPLEMENTATION PROCESS



**UASG.TECH**



## 4. THE PROCESS FOR ACHIEVING UA

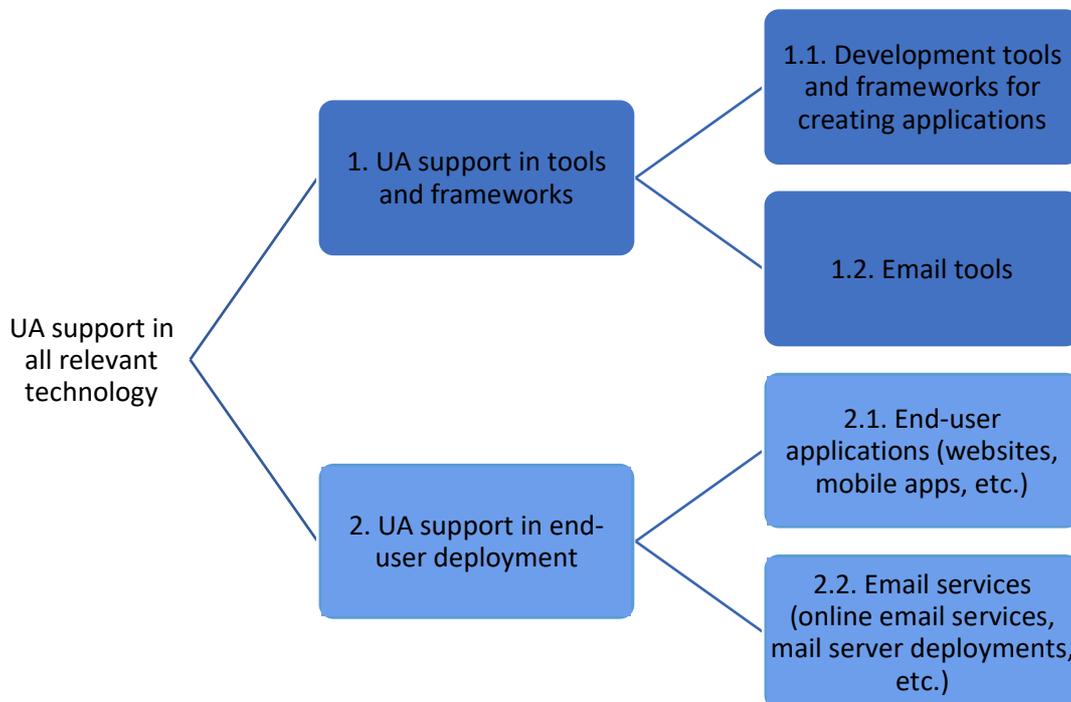
Work on Universal Acceptance can be categorized by determining gaps and fixing them at two layers of software:

1. The underlying software development tools, frameworks, etc., which are used to create the software solutions for end-users.
2. The actual software applications deployed and used by the end-users, e.g., mobile apps, websites, email services.

End-user applications, websites, and email services that are deployed are dependent on the underlying development tools and frameworks. If the underlying technology is not UA-ready, the end-user software application development and its deployment will be much harder to make UA-ready.

On the other hand, even if the underlying technology is UA-ready, the end-user software application development and deployment does not automatically become UA-ready. Software engineers, programmers, and system administrators still need to make sure they are using the UA-ready tools and frameworks properly and configuring them appropriately during the software application development and deployment process. Only proper software design, development, and deployment would make mobile apps, websites, and email services UA-ready for end-users.

At each layer, UA-readiness means supporting both domain names, including IDNs, and email addresses, including internationalized email addresses (EAI). This creates a four-way categorization of work for addressing UA-readiness, as illustrated in Figure 1.



**Figure 1: Steps for Updating Technology to Support Universal Acceptance**

UA-readiness translates into the requirement that software applications should be able to accept, validate, process, store, and display all valid domain names and email addresses regardless of script, language, or character length.



**Figure 2: Software Requirements for Universal Acceptance of Domain Names and Email Addresses**

The underlying technology is developed by the stakeholder groups identified by the UASG as technology enablers and email tool providers. For end-user development and deployment, the UASG identifies the relevant stakeholder groups as technology developers, email service providers, and email administrators.

For software applications to work for end-users, the UASG engages with all stakeholders. For technology enablers and email tool providers, the UASG conducts gap analysis of various tools, identifies and prioritizes issues, and then reaches out to encourage them to address these issues. For technology developers, email service providers and email administrators, the UASG also conducts gap analysis and outreach to encourage them to address these issues. In addition, the UASG develops materials and conducts training.

## 5. UA-READINESS SCOPE AND EVALUATION FRAMEWORK

The UASG has compiled documentation outlining the scope of UA-readiness, accompanied by a framework and a range of supplementary resources aimed at facilitating the testing and updating of software and systems to be UA-ready.

1. [Layers of Technology Relevant for UA Testing](#)
2. [UA Testing Framework](#)
3. [EAI Self-Certification Program](#)
4. [Make Systems UA-Ready With Free Code Samples](#)

### 5.1 LAYERS OF TECHNOLOGY RELEVANT FOR UA TESTING

Multiple layers of technology may need to be fixed to achieve UA-readiness. Figure 3 shows how such technology can be categorized at a high level with examples for each layer. The layers highlighted in gray are examples of end-user applications and websites, whereas the other layers represent the underlying technology and associated standards.

The UA-readiness of an end-user application has dependency on the UA-readiness of the underlying technology. Therefore, the whole technology stack would need to be reviewed and upgraded, where needed, to be able to accept, validate, process, store, and display all valid domain names and email addresses. The figure only provides a limited list that may be extended to include database management systems, identity platforms, additional system tools, and more.



---

### **Applications and Websites**

- Amazon.com, ICANN.org, Wikipedia.org, custom websites globally
- Acrobat, Google-Docs, PowerPoint, Safari, custom apps

---

### **Social Media and Search Engines**

- Bing, Chrome, Firefox, Safari, local (e.g., Chinese) browsers
- Facebook, Instagram, Skype, Twitter, Viber, WeChat, WhatsApp

---

### **Programming Languages and Frameworks**

- C#, Java, JavaScript, PHP, Python, Swift
- Angular, J2EE, .NET core, Oracle, SAP, Spring, WordPress

---

### **Platforms, Operating Systems and System Tools**

- Android, App Stores, iOS, Linux, Windows
- Active Directory, OpenLDAP, OpenSSL, Ping, Telnet

---

### **Standards and Best Practices**

- IETF RFCs, Unicode CLDR, W3C HTML, WHATWG
- Industry-based standards (health, aviation, ...)

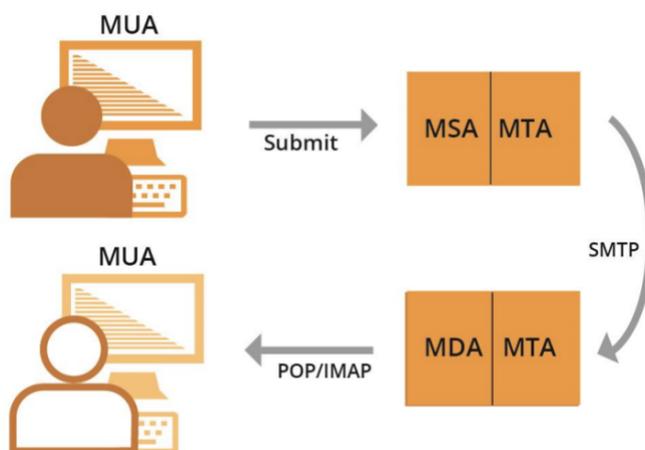
=

## **Figure 3: Layers of Technology Relevant for Universal Acceptance Testing**

Similarly, email systems and services also need to be updated to adopt the recent changes in the standard for Email Address Internationalization (EAI). Figure 4 shows the different email components (explained below), all of which need to be upgraded and configured to support EAI.

- MUA – Mail User Agent: A client program that a person uses to send, receive, and manage mail.
- MSA – Mail Submission Agent: A server program that receives mail from a MUA and prepares it for transmission and delivery.
- MTA – Mail Transfer Agent: A server program that sends and receives mail to and from other Internet hosts.
- MDA – Mail Delivery Agent: A server program that handles incoming mail and typically stores it in a mailbox or folder.

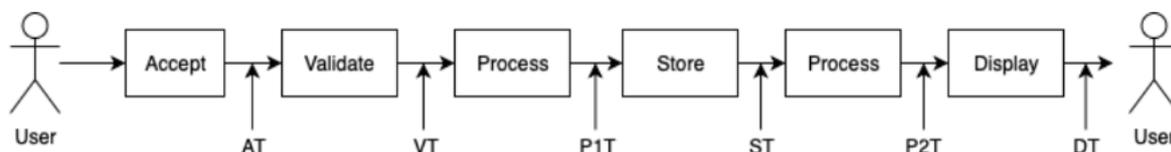
In addition, the email ecosystem also contains additional tools, including spam filtering, calendars, and others related to emailing systems.



**Figure 4: Layers of Technology Relevant for Universal Acceptance Testing**

## 5.2 UA TESTING FRAMEWORK

The UA of all domain names and email addresses requires that all software applications accept, validate, process, store, and display them correctly. The [Universal Acceptance Readiness Framework](#) (UASG026) lays out details on how to check for UA-readiness using a gating approach to verify UA conformance of an application (shown in Figure 5). This gating approach is based on applying tests at the various steps, named gates, and on the various components. Accept Tests (AT), Validate Tests (VT), Process Tests on the Input and Output (P1T, P2T), Store Tests (ST), and Display Tests (DT) are identified. The following figure shows the proposed gating approach.



**Figure 5: Universal Acceptance Testing Framework**

Details of these tests and how they are applicable to the different categories of applications are provided in the framework document. Developers can use this framework to plan specific test cases and check the UA-readiness of their applications.

UA-readiness issues can happen if software applications are not able to handle any one of the following categories of a domain name or email address at any of the stages outlined in the UA-Readiness Framework.

### 1. Domain Names

1. **New short** top-level domain names: example.sky
2. **New long** top-level domain names: example.international
3. Internationalized Domain Names: παράδειγμα.eu

### 2. Email Addresses

1. ASCII@ASCII; new short or long TLD: ekrem@misal.istanbul
2. ASCII@IDN: john@société.org
3. **Unicode@ASCII**: 测试@example.com



4. **Unicode@IDN:** [ईमेल@उदाहरण.भारत](mailto:ईमेल@उदाहरण.भारत)
5. **Unicode@IDN;** right-to-left scripts: [اي-میل@مثال.موقع](mailto:اي-میل@مثال.موقع)

To allow for global UA testing, a set of domain names has been registered covering the various categories and scripts, including Arabic, Armenian, Bengali, Chinese (Simplified, Traditional), Cyrillic, Devanagari, Ethiopic, Georgian, Greek, Gujarati, Gurmukhi, Hangul, Hebrew, Hiragana, Kannada, Katakana, Lao, Latin, Malayalam, Myanmar, Oriya, Sinhala, Tamil, Telugu, Thaana, Thai, and Tibetan. In addition, an email test-bed has also been set up using these domain names. This provides a comprehensive dataset to test applications for UA as captured in [UASG004](#). Some example email categories and addresses are listed below from the companion [UASG004A](#) document with this dataset in UTF8 format:

- Example Domain Names
  - #2, ASCII.ASCII, new-short, Short ASCII  
[universal-acceptance-test.icu](http://universal-acceptance-test.icu)
  - #4, IDN.IDN, Armenian  
[համընդհանուր-ընկալքութեան.հայ](http://համընդհանուր-ընկալքութեան.հայ)
  - #10, IDN.IDN, Gujarati  
[સાર્વત્રિક-સ્વીકૃતિ-પરીક્ષણ.ભારત](http://સાર્વત્રિક-સ્વીકૃતિ-પરીક્ષણ.ભારત)
  - #12, IDN.IDN, Hangul  
[다국어도메인이용환경테스트.한국](http://다국어도메인이용환경테스트.한국)
  - #13, IDN.IDN, RTL, Hebrew  
[מבטח-האוניברסלי.קוד](http://מבטח-האוניברסלי.קוד)
  - #16, IDN.IDN, Katakana  
[ユニバーサルアクセプタンス.クラウド](http://ユニバーサルアクセプタンス.クラウド)
  - #17, IDN.IDN, Lao  
[ສາກົນ-ການຍອມຮັບ-ທົດລອງ.ລາວ](http://ສາກົນ-ການຍອມຮັບ-ທົດລອງ.ລາວ)
  - #18, ASCII.IDN, Latin  
[Universales-Akzeptanz-Test.vermögensberatung](http://Universales-Akzeptanz-Test.vermögensberatung)
- Example Email Addresses
  - #32, ASCII@ASCII.ASCII, new-long, Long ASCII  
[email-test@universal-acceptance-test.international](mailto:email-test@universal-acceptance-test.international)
  - #34, Unicode@IDN.IDN, RTL, Arabic  
[تجربةبريد-الالكتروني@تجربة-القبول-الشامل-موريتانيا](mailto:تجربةبريد-الالكتروني@تجربة-القبول-الشامل-موريتانيا)
  - #37, Unicode@IDN.IDN, Cyrillic  
[почта-тест@универсальное-принятие-тест.москва](mailto:почта-тест@универсальное-принятие-тест.москва)
  - #38, Unicode@IDN.IDN, Devanagari  
[ईमेल-परीक्षण@सर्वभौमिक-स्वीकृति-परीक्षण.संगठन](mailto:ईमेल-परीक्षण@सर्वभौमिक-स्वीकृति-परीक्षण.संगठन)
  - #39, Unicode@IDN.IDN, Georgian  
[ფუნქციონალური-ტესტი@უნივერსალური-თავსაზღვრების-ტესტი.გე](mailto:ფუნქციონალური-ტესტი@უნივერსალური-თავსაზღვრების-ტესტი.გე)
  - #40, Unicode@IDN.IDN, Greek  
[ηλεκτρονικό-μήνυμα-δοκιμή@καθολική-αποδοχή-δοκιμή.ευ](mailto:ηλεκτρονικό-μήνυμα-δοκιμή@καθολική-αποδοχή-δοκιμή.ευ)
  - #52, Unicode@IDN.IDN, Sinhala  
[ඉ-නැපැල්-පිරික්සීම@විශ්ව-සම්මුති-පිරික්සීම.ලංකා](mailto:ඉ-නැපැල්-පිරික්සීම@විශ්ව-සම්මුති-පිරික්සීම.ලංකා)
  - #55, Unicode@IDN.IDN, Thai  
[อีเมลทดสอบ@ยูเอททดสอบ.ไทย](mailto:อีเมลทดสอบ@ยูเอททดสอบ.ไทย)
  - #56, Unicode@IDN.IDN, Simplified Chinese



## 电子邮件测试@普遍适用测试.我爱你

The domain names resolve to a UA test webpage and an email sent to any of the addresses gets an auto-response in multiple languages, shown in Figure 6. To see the messages, click on the links or send an email to one the addresses above.

The figure shows two parts: (a) Test Webpage and (b) Test Email Response. Part (a) displays the UA test webpage in English, Arabic, Spanish, French, Russian, and Chinese. Each language version contains a message explaining the purpose of the test: to ensure that domain names and email addresses are accepted equally by all Internet-enabled applications, devices, and systems. Part (b) shows an email response in the same languages, stating that no reply is required and thanking the sender for their interest in the UA project.

(a) Test Webpage

(b) Test Email Response

### Figure 6: Responses of Test Domains and Emails for Universal Acceptance Testing

## 5.3 EAI SELF-CERTIFICATION PROGRAM

The UASG’s EAI Working Group developed and published the [first version of the EAI Self-Certification Guide](#). The main objective of this document is to guide mail service providers, software vendors, and developers about the functionalities required to self-certify the readiness of email systems, tools, and utilities for EAI.

The guide provides IT and procurement managers the necessary information about EAI-readiness levels that may help them in selecting or purchasing email services and tools. Levels are categorized by the words “Platinum”, “Gold”, and “Silver”. These categories make



the information easier to understand. The table below shows each category and their functionalities.

**Table 1: EAI-Readiness Levels Used in the EAI Self-Certification Guide**

Level	Functionality
Platinum	<ol style="list-style-type: none"><li>1. Meets all Gold requirements listed below.</li><li>2. Supports all UASG best practices guidelines.</li><li>3. Hosting functionality for internationalized mailboxes is enabled by default [to the user and administrator].</li><li>4. May also include “aspirational” extra features which improve user experience, also listed below.</li><li>5. Documentation clearly explains how to set up, use, and administer Platinum-level features.</li><li>6. All integrated tools or utilities should be EAI enabled as well.</li></ol>
Gold	<ol style="list-style-type: none"><li>1. Meets all Silver requirements listed below.</li><li>2. Hosting functionality for internationalized mailboxes is available but not enabled by default.</li><li>3. Marketing materials and public messaging promote globally inclusive email systems and features.</li><li>4. Documentation clearly explains how to set up and use Gold-level features.</li><li>5. Functionality that processes email addresses rather than messages, for example address books, can create and store Unicode email addresses.</li></ol>
Silver	<ol style="list-style-type: none"><li>1. Enables its users to process email messages from [EAI][Unicode] mailboxes but does not host such mailboxes.</li><li>2. Functionality that processes email addresses rather than messages, for example address books, need not create or store Unicode email addresses.</li></ol>

The guide defines EAI support levels so that motivated vendors and service providers have a way to compete based on the quality of their UA-readiness, and customers have a way to compare products, services, and system providers to each other.

Mail service providers, software vendors, and developers who aim to see the level of EAI support and improve their systems can perform self-certification v.01 on <https://uasg.tech/eai-certification/> and share comments at [info@uasg.tech](mailto:info@uasg.tech).

#### 5.4 MAKE SYSTEMS UA-READY WITH FREE CODE SAMPLES

The UASG developed and published [UA-Ready Code Samples in Java, Python, and JavaScript](#) (UASG043) that can be used freely (BSD 3 License included) to make software systems completely UA-ready by covering all applicable categories:

- ASCII domain names with new and longer ASCII TLDs
- IDNs
- Internationalized email addresses with the domain name part in either ASCII format or IDN



- Use of the latest available set of TLDs as they may be added or removed over time

The [UASG043](#) report allows developers to find minimal running prototypes (MVPs) that demonstrate how to program for UA-readiness on the Java, Python and JavaScript technology stacks. The purpose is to demonstrate UA-readiness to developers so that they can be persuaded to adopt the practice.

To make systems UA-ready, it is important to adopt this practice by getting the code samples. In addition, the UASG community is encouraged to star (☆) the project on <https://github.com/icann/ua-code-samples>.

Depending on the libraries, the samples illustrate compliance with Internationalized Domain Names in Application (IDNA 2008) or EAI. The following languages and libraries are nominally used in the code samples.

**Table 2: List of Programming Languages and Libraries Where Sample Codes are Available to Make Systems UA-ready**

Language	Framework/Library	Version	EAI	IDNA
Java	commons-validator	1.7	X	X
	guava	31.0.1-jre		X
	icu	70.1		X
	jakartamail	2.0.1	X	
Python	idna	3.3		X
	email-validator	1.1.3	X	
	smtplib	Python 3.10	X	
Javascript	idna-uts46	1.1.0		X
	validator	13.7.0	X	
	nodemailer	6.7.2	X	

Some libraries use a static list of TLDs that is shipped with the sources, therefore, depending on the version of the library, the TLD list can be obsolete. As the list of TLDs is regularly updated, any static list will quickly become obsolete. To mitigate this, the [UASG043](#) report provides a code snippet to download the TLDs from the IANA website <https://data.iana.org/TLD/tlds-alpha-by-domain.txt>.



## 6. UA-READINESS GAP ANALYSIS

Recent work done by the UASG has focused on understanding the gaps in the technology related to UA-readiness. The UASG has investigated the following technology:

1. Technical Platforms and Applications
  - a. Platforms used to enable and develop applications: standards, frameworks, libraries, and tools.
    - i. [UA-Readiness of Programming Languages and Frameworks](#)
    - ii. [UA-Readiness Evaluation of Standards and Best Practices](#)
    - iii. [UA-Readiness of Networking Command Line Tools](#)
    - iv. [UA of Content Management Systems \(CMS\) Phase 1 - WordPress](#)
    - v. [UA-Readiness of Web Hosting Tools \(cPanel, Plesk, ISPConfig\)](#)  
(published in FY23)
  - b. Actual deployment: applications, including websites.
    - i. [UA-Readiness of Browsers](#)
    - ii. [UA-Readiness of Social Media Platforms](#)
    - iii. [EAI Acceptance Rates of the Top 2,000 Global Websites in 2022](#)
    - iv. [UA-Readiness of Open-Source Code Pilot](#)
    - v. [UA-Readiness of Popular Identity Platforms \(Okta, Auth0, OpenIAM\)](#)  
(published in FY23)
2. Email Software and Services
  - a. Platforms used to enable and develop applications: standards, frameworks, libraries, and tools.
    - i. [Testing Email Tools for EAI Support](#)
    - ii. [Considerations for Naming Internationalized Email Mailboxes](#)
  - b. Actual deployment: email services and servers deployed.
    - i. [EAI-Readiness of Email Servers](#) (measured quarterly)

The following sections present a summary of the findings from these studies. Details are available in the reports linked above. It should be noted that the software tools and applications discussed may become UA-ready over time, so it is advised to always check the documentation of the latest version to see the degree of support for UA.

### 6.1 UA-READINESS OF PROGRAMMING LANGUAGES AND FRAMEWORKS

Programming languages and their specific libraries are used to develop applications. These software applications can provide a host of Internet services. For example, an application can ask for an email address as a username to register a new user, send an email to confirm the request, and then set up the new user using the email address as the user's identity. The software application can run on a variety of operating systems like Linux, Windows, iOS or on mobile platforms like Android. Many applications make use of Internet identifiers which - while historically represented only in characters in American Standard Code for Information Interchange (ASCII) - can now be fully multilingual. These identifiers include:

- Domain names, e.g., example.com or 普遍适用测试.我爱你
- Email addresses, e.g., email@example.com or [อีเมลทดสอบ@ยูเอชทีเอส.ไทย](#)



It is important that all stakeholders involved in the development of a software application are aware of the available libraries for their chosen development environment that can be used for processing Internet identifiers and how to use them to support UA. Additionally, it's important to have a clear basis to assess those libraries for technical and business suitability regarding UA-readiness and compliance.

The UASG conducted a study on [Reviewing Programming Languages and Frameworks for Compliance with Universal Acceptance Good Practice](#) in 2019. This study documented the test cases for evaluating UA support of programming languages and frameworks based on the requirements from different protocols, including those for IDNA2008 and EAI. This work was used to test widely used programming languages and platforms in 2020 and the findings were published in [UASG018A: Universal Acceptance Compliance of Some Programming Language Libraries and Frameworks](#). In 2022, the study was extended to also review PHP and to investigate the app development on mobile platforms, both iOS and Android. The results have been published in [UASG030: UA-Readiness of Some Programming Language Libraries and Frameworks \(Phase 3\)](#).

The findings from the latest report on PHP as well as iOS and Android platforms are summarized in Table 3. Red signifies not being UA-ready; pink signifies being UA-ready, but some details need to be managed; and green signifies being UA-ready.

**Table 3: Level of UA Support in Programming Language Libraries in 2022**

Language	Library Name	Platform	Type of Test
Swift	MessageUI	IOS	Email
Swift	URLSession	IOS	Domain Name
Swift	Alamofire	IOS	Domain Name
Swift	IDNA-Cocoa	IOS	Domain Name
PHP	cURL*	Windows	Domain Name
		Linux	Domain Name
PHP	mail	Windows	Email
PHP	emailValidator	Windows & Linux	Email
PHP	Guzzle	Windows & Linux	Domain Name
PHP	intl	Windows & Linux	Unicode
PHP	PHPMailer	Windows & Linux	Email
PHP	Symfony	Http-client	Windows & Linux
		Polyfill-intl-idn	Unicode, Domain Name
		Mailer	Email
Kotlin	okHttp	Android	Domain Name



Kotlin	URLConnection	Android	Domain Name
Kotlin	Retrofit	Android	Domain Name
Kotlin	fuel	Android	Domain Name
Kotlin	Volley	Android	Domain Name
Kotlin	Apache HttpClient	Android	Domain Name
Kotlin	Jakarta Mail	Android	Email
Kotlin	Email Intent	Android	Email

The results show that despite good compliance with EAI and IDNA2008 from native iOS apps like Mail or Safari, Apple doesn't seem to provide libraries with the same level of acceptance for developers. Perhaps Apple always converts to an A-Label before using their HTTP libraries. For email libraries, a workaround is to go through the mail native app, bypassing the standard way of popping an email composition modal (obsolete as noticed), and everything works fine from there. For Android, most of the tested HTTP libraries are using the same base code, therefore the results are quite similar except for Apache HttpClient, which should not be used. Therefore, no library that uses the base Android network stack would be IDNA2008 compliant. An IDNA2008 compliant solution could be achieved as Android already contains and uses `icu.text.IDNA` that offers that compliance. The most used SMTP library on Android offers good compliance with EAI. A more detailed analysis is presented in the report, along with the test results. PHP has multiple options for libraries which are UA-ready.

The findings from the earlier report on additional languages and frameworks are summarized in Table 4 below. Further details are in the [report](#). The results show that programming languages Java, JavaScript, and Python3 have support for processing IDNs and EAI. Some additional platforms support IDNs but EAI is not supported by the remaining libraries.

**Table 4: Level of UA Support in Programming Language Libraries in 2021**

Language	Library Name	Type of Test
C	libcurl	Email
C	libidn2	Unicode
C#	mailkit	Email
C#	microsoft	Unicode
Go	idna	Domain Name
Go	mail	Email
Go	smtp	Email



Java	commons-validator	Domain Name and Email
Java	guava	Domain Name
Java	icu	Unicode
Java	jakartamail	Email
Java	jre	Unicode
JavaScript	idna-uts46	Unicode, Domain Name
JavaScript	nodemailer	Email
JavaScript	validator	Domain Name and Email
Python3	django_auth	Email
Python3	email_validator	Email
Python3	encodings_idna	Unicode
Python3	idna	Unicode, Domain Name
Python3	smtplib	Email
Rust	idna	Unicode, Domain Name
Rust	lettre	Email

## 6.2 UA-READINESS EVALUATION OF STANDARDS AND BEST PRACTICES

An initial investigation was conducted to identify and prioritize the technical standards bodies and groups that might be relevant for work on UA as well as identify any indexes and indicators that could include UA-readiness as a data point. The end goal of this work was to identify opportunities for engagement and collaboration for the advancement of the UASG's goals.

The work identified key organizations and studied their work streams, eventually generating a comprehensive list. The list is labeled with categories and priority levels as well as descriptions, recommendations, and other relevant information. Some of the organizations and their work identified for conducting a more detailed analysis include, but are not limited to, the following:

- W3C Accessibility Education and Outreach Working Group (EOWG)
- W3C Internationalization Working Group (i18n)
- WHATWG HTML Living Standard
- IETF art-dmarc
- IETF art-regext: Registration Protocols Extensions
- IRTF Hrpc: Human Rights Protocol Considerations Research Group
- Unicode International Components for Unicode (ICU)
- ITU-T Joint Coordination Activity on Accessibility and Human Factors (JCA-AHF)



- Python Software Foundation PEPs
- Cybersecurity organizations engagement

The UASG aims to conduct a more detailed analysis of the suggested standards by these organizations and then determine how to prioritize and engage with them. For the indexes and indicators studied, these require a high degree of commitment from the organization that intends to provide the data, as well as the involvement of governments. Working on enhancing the indexes is more challenging and will be further investigated by the UASG. Additional details are available in [UASG040: UA-Readiness Evaluation of Standards and Best Practices](#).

### 6.3 UA-READINESS OF NETWORKING COMMAND LINE TOOLS

Modern operating systems have a variety of command line tools used in system management and program development. Many of these tools operate on domain names, while a few do so on email addresses. The study [UASG024: UA-Readiness of Command Line Networking Tools](#) was conducted in 2019 and looked at these tools to see how well they support UA.

The UA-related support for the tools reviewed is summarized in Table 5. See the report for further details. **Yes** means that the command at least accepts, validates, and processes. **Yes(D)** means it also displays domain names and shows IDNs as U-labels. **Yes** means that the command accepts IDNs but processes them with IDNA2003 rather than IDNA2008. An empty box means that the system is not distributed with that tool. **No** means IDNs as U-labels are not supported. An empty box means that the system is not distributed with that tool.

**Table 5: Level of UA Support by Some Networking Tools in 2019**

Tool	MacOS 10.14 (BSD/Mach)	FreeBSD 12 (BSD)	Ubuntu 18 (linux)	Centos 7 (linux)	Windows 10
host	No	No	No	Yes*(D)	
ping	Yes*	No	Yes*(D)	Yes*(D)	Yes
ping6	Yes*	No	Yes*(D)	Yes(D)	
tracert	Yes*	No	Yes*(D)	Yes(D)	
tracert6	Yes*	No	Yes*(D)	Yes(D)	
dig	No		No	Yes*(D)	
nslookup	No		No	Yes*(D)	No
telnet	Yes*	No	No		
openssl	Yes*	No	Yes*	No	
gnutls-cli		Yes	Yes		
tracert					Yes



## 6.4 UA-READINESS OF CONTENT MANAGEMENT SYSTEMS

Content Management Systems (CMS) allow for quick development, deployment, and maintenance of websites, and are popular tools to develop websites globally. There are many Content Management Systems available, both open source and proprietary. Examples include WordPress, Joomla, Drupal, and many more. The UASG recently conducted a pilot study with the intention to carry out UA-readiness testing in CMS, specifically WordPress and its associated plugins, for IDNs, path [Internationalized Resource Identifiers \(IRIs\)](#), and EAI. The results are documented with additional details in [UASG032: UA of Content Management Systems \(CMS\) Phase 1 - WordPress](#) published in 2021.

Table 6 summarizes UA support in the WordPress core system and the plugins tested. A-level means that all tests passed; B-level means end-to-end works, but some tests fail; and C-level means that end-to-end testing did not pass, although some individual tests may have passed. The results indicate reasonable (though not complete) support for IDNs and IRIs, but no support for EAI.

**Table 6: Level of UA Support by WordPress and Some Plugins**

	Plugin Name	IDN UA Readiness	Path IRI UA Readiness	EAI UA Readiness
<b>CMS</b>	WordPress Instance	B-Level	A-Level	C-Level
<b>Extensions for Subscription Management</b>	MailChimp	C-Level	B-Level	C-Level
	Mailster	C-Level	C-Level	C-Level
	OptinMonster	B-Level	B-Level	C-Level
	Newsletter	B-Level	A-Level	C-Level
	Sumo List Builder	C-Level	C-Level	C-Level
<b>Extensions for Membership Management</b>	MemberPress	B-Level	A-Level	C-Level
	WooCommerce	B-Level	A-Level	C-Level
	Restrict Content Pro	B-Level	A-Level	C-Level
	Paid Memberships Pro	B-Level	A-Level	C-Level
	S2Member	C-Level	C-Level	C-Level
<b>Extensions for Event Management</b>	Events Manager	B-Level	A-Level	C-Level
	WP Event Manager	C-Level	C-Level	C-Level
	Event Organizer	B-Level	A-Level	C-Level



	All - in - One Event Calendar	B-Level	B-Level	C-Level
	Event Espresso 4 Decaf	B-Level	C-Level	C-Level

## 6.5 EAI-READINESS OF POPULAR WEB HOSTING TOOLS

Web hosting tools are critical to managing and maintaining a functional website. They are the building blocks required to keep a site accessible on the Internet and to provide essential services like email. For anyone who manages a website, whether for personal use, business, or entities like governments, web hosting tools are incredibly important. In FY23, the UASG published a study of the UA-readiness of three of the most widely used web hosting tools: cPanel, Plesk, and ISPConfig. (UASG042 – UA-Readiness of Web Hosting Tools, [Report](#) and [Presentation](#))

The purpose of the study was to determine the extent to which web hosting tools offer the capacity for customers to build websites and host emails in accordance with UA in nine different configurations on different operating systems (Linux and Windows for Plesk, Linux for cPanel and ISPConfig).

The assessment focused on evaluating all the basic functions related to UA for domains and email addresses (input, validation, storage, processing, and display). More specifically, the study tested whether IDNs and email addresses could be entered and encoded directly; email could be sent, received, and read in the appropriate script; the web service could be set up using native scripts; and that all elements could be displayed appropriately.

The assessment revealed that cPanel, Plesk, and ISPConfig have embraced UTF-8 to a certain extent, but that they are not fully UA-ready. This was most often due to the underlying services. Since several underlying services are not ready to support local scripts, the creators of the web hosting tools block any non-ASCII characters throughout most of their interfaces.

Considering the three web hosting tools tested, Plesk had the highest chance of providing a full local script experience because it offers a file access tool. As such, the whole user experience could be driven through a web interface. With Plesk, email, domain name, user, database, and files can be managed using various web services all available from a local native script supporting web browser.

In general, working with the underlying service providers will help enable UA-readiness in the future.

The table below provides the overall results per web hosting tool feature, including the UTF-8 compliance of the environment in default installation. In some cases, additional steps were taken for some UA-ready functionality to pass.

### Legend

Test passed after additional steps (W)	Test passed when using user with ASCII credentials (A)
Test failed (F)	



## Testing Results

Web Hosting Tool	OS	Web Server Version	Email Server	Database	UTF-8 Environment	Access	Domain Name Activation and Management	Email Service	Database	Webhosting
cPanel 104.0.7	Rocky Linux 8.6 (Green Obsidian)	Apache 2.4.54	Exim 4.95	MySQL 8.0.29	W	F	F	F	F	A
	Ubuntu 20.04 LTS	Apache 2.4.54	Exim 4.95	MySQL 8.0.29	W	F	F	F	F	A
	CentOS Linux 7 (Core)	Apache 2.4.54	Exim 4.95	MySQL 5.7.38	W	F	F	F	F	A
Plesk Obsidian Web Host Edition Version 18.0.45	AlmaLinux 8.6 (Sky Tiger)	Apache 2.4.37 nginx 1.20.2	Postfix 3.5.14	MariaDB 10.3-32	W	F	F	F	F	A
					W	F	F	F	F	A
	Debian GNU/Linux 11 (bullseye)	Apache 2.4.37 nginx 1.20.2	Postfix 3.5.13	MariaDB 10.5-15 PostgreSQL 13.7	W	F	F	F	F	A
					W	F	F	F	F	A
Windows 10	Microsoft IIS 10.0	MailEnable Standard 10.34	MariaDB 10.6	W	F	F	F	F	A	
ISPConfig 3.2.2	openSUSE Leap 15.4	Apache 2.4.51	Exim 4.94.2	MySQL 8.0.29	F	W	W	F	W	A
	Oracle Linux Server 8.6	Apache 2.4.37	Postfix 3.5.8	MySQL 8.0.26	W	W	W	F	W	A
	Fedora Linux 36 (Server Edition)	Apache 2.4.54	Postfix 3.6.4	MariaDB 10.5-16	W	W	W	F	W	A

More information about the bug reporting on these platforms is available in the Remediation section of this report.

## 6.6 UA-READINESS OF POPULAR WEB BROWSERS

A recent study was conducted to understand how the Internet can better accommodate user choice of script by analyzing the current usability of IDNs and internationalized email addresses in browsers. The results are described in [UASG036: UA-Readiness of Browsers](#). Following the previous report [UASG016: UA of Popular Browsers](#) done 2017 on the UA of popular browsers, this report intended to not only capture the more recent behavior but also expand the testing to include additional browsers, especially those used locally in different geographic regions, and test browsers on mobile platforms. The following testing was conducted and tabulated:

A	Confirm URL is displayed correctly as pasted into the bar
B	Confirm that the correct page was loaded
C	Confirm that the URL is displayed correctly in the bar
D	Confirm that the page title is displayed correctly in the window/tab
E	Add the URL to browser bookmarks/favorites - operation completes without error
F	Confirm that the URL (and where supported page title) displays correctly in the correct format as added

The detailed results of testing against URLs in different scripts are given in Table 7 below.





(e) iOS

URL/ Browser	360	Anigo Mail	Atom Mail	Chrome	Edge	Epic Privacy Bro	Firefox	Internet Explorer	Opera	Safari	Samsung Brows	Sogou	UC Browser	Yandex
universal-acceptance-test.international	N/A	N/A	N/A	Y	Y	Y	Y	N/A	Y	Y	N/A	N/A	N/A	Y
universal-acceptance-test.icu	N/A	N/A	N/A	Y	Y	Y	Y	N/A	Y	Y	N/A	N/A	N/A	Y
تجربة قبول النطاقات الدولية	N/A	N/A	N/A	F	B, C, D, E, F	F	C, F	N/A	B, C, D, E, F	C, F	N/A	N/A	N/A	Y
универсальное-принятие-тест.москва	N/A	N/A	N/A	F	B, C, D, E, F	C, F	C, F	N/A	C, F	B, C, D, E, F	N/A	N/A	N/A	Y
सर्वोपचारक-प्रतिस्वीकृत-परीक्षण	N/A	N/A	N/A	F	B, C, D, E, F	C, F	C, F	N/A	C, F	F	N/A	N/A	N/A	Y
Universales-Akzeptanz-Test.vermögensberatung	N/A	N/A	N/A	F	B, C, D, E, F	C, F	C, F	N/A	C, F	F	N/A	N/A	N/A	Y
普通适用测试.我爱你	N/A	N/A	N/A	F	B, C, D, E, F	C, F	C, F	N/A	C, F	F	N/A	N/A	N/A	Y
تجربة قبول النطاقات الدولية.com	N/A	N/A	N/A	F	C, F	C, F	C, F	N/A	A, C, F	A, C, F	N/A	N/A	N/A	F
xn-----cldbabochu9c2b9f1acco4c.xn--mgbah1a3h9krd	N/A	N/A	N/A	C, F	B, C, D, E, F	C, F	C, F	N/A	A, B, C, D, E, F	A, C, F	N/A	N/A	N/A	C, F
universal-acceptance-test.icu/测试	N/A	N/A	N/A	C, F	C, F	C, F	F	N/A	B, C, D, E, F	C, F	N/A	N/A	N/A	C, F
ທຸລະກິດ.ໂທ	N/A	N/A	N/A	F	B, C, D, E, F	B, C, F	C, F	N/A	C, F	F	N/A	N/A	N/A	Y
အမျိုးသားအသင်း-အသင်း.အင်း	N/A	N/A	N/A	F	B, C, D, E, F	C, F	F	N/A	C, D, E, F	C, F	N/A	N/A	N/A	Y
համընդհանուր-ընկալում-թեստ.հայ	N/A	N/A	N/A	F	B, C, D, E, F	C, F	F	N/A	C, F	Y	N/A	N/A	N/A	Y
универсальное-принятие-тест.москва.бг	N/A	N/A	N/A	F	B, C, D, E, F	C, F	F	N/A	C, F	C, F	N/A	N/A	N/A	Y
સર્વોપચારક-પ્રતિસ્વીકૃત-પરીક્ષણ.૦૧૦૨	N/A	N/A	N/A	F	B, C, D, E, F	C, F	F	N/A	C, F	Y	N/A	N/A	N/A	Y
どこでもつかえる.みんな	N/A	N/A	N/A	F	B, C, D, E, F	C, F	F	N/A	C, F	Y	N/A	N/A	N/A	Y
épreuve-acceptation-universelle.org	N/A	N/A	N/A	F	C, F	Y	F	N/A	C, F	Y	N/A	N/A	N/A	Y
普通适用测试.我爱你	N/A	N/A	N/A	C, F	B, C, D, E, F	C, D, E, F	B, C, D, E, F	N/A	B, C, D, E, F	B, C, D, E, F	N/A	N/A	N/A	C, F

In considering the results of this study, the general outlook of UA-readiness within modern web browsers is largely positive with almost every tested browser handling the paste and subsequent navigation of all internationalized URLs correctly, resulting in a perfectly usable browsing experience for sites hosted using IDNs.

These findings are perhaps unsurprising due to the strong support for Unicode in modern operating systems and therefore in software platforms used to develop browsers allowing for reliable presentation of internationalized content. However, it was especially encouraging to note that both right-to-left scripts and technically challenging examples such as the Chinese Open Dot use case resulted in a valid resolution of the URL to the correct page in almost every instance.

The one obvious area for improvement for browser vendors is in the handling of bookmarking for IDNs with several vendors listing previously stored bookmarks using Punycode conversions, which have little or no significance to a reader. This was in some ways an unexpected finding due to the strong adherence to Unicode presentation observed in the other tests, including frequent promotion to Unicode for display in URL bars when pasting Punycode URLs.

### 6.7 UNIVERSAL ACCEPTANCE OF SOCIAL MEDIA PLATFORMS

UA-readiness of different social network platforms was also recently measured, published in [UASG035: UA Readiness of Social Media Platforms](#). The following social media applications were tested in this study in different browser and operating systems:

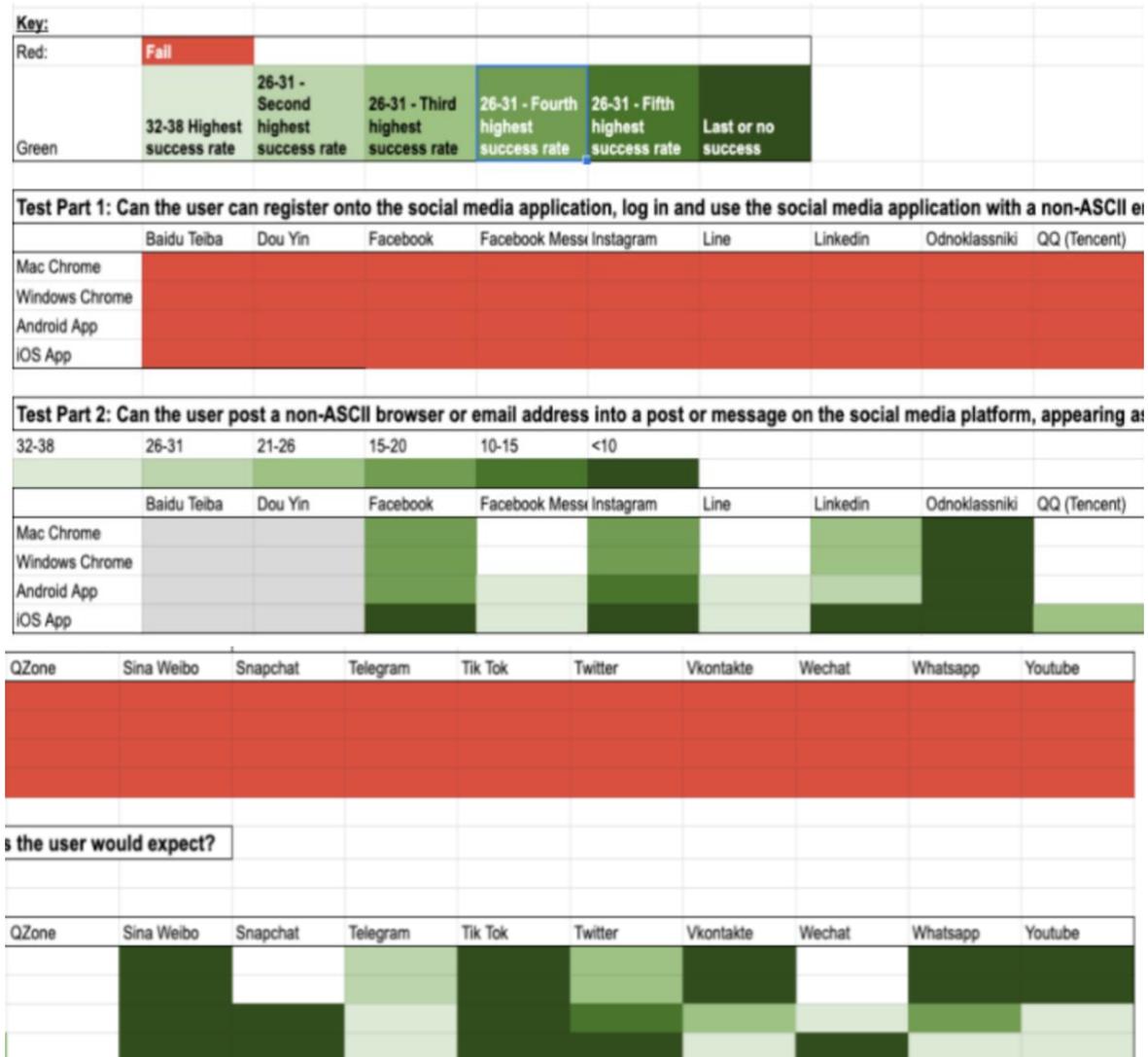
- Facebook
- Facebook Messenger
- Twitter
- Instagram
- WhatsApp
- YouTube
- TikTok
- WeChat
- Baidu Teiba
- Dou Yin (Tik Tok in Chinese)
- QQ (Tencent)
- Sina Weibo

The tests included registering a new account, authenticating a user, logging in, sharing a domain name or email in posts, setting up domain name and email in profile account, sharing domain name in a private message, and displaying a domain name. A varied set of domain names and email addresses in different scripts were tested.



The summarized results of the testing are presented in Table 8. No social media application passed the test of registering with a non-ASCII, internationalized email address. This is shown in the first chart as ‘Test Part 1’ as being red. The second part of the tests were around the posting and messaging of non-ASCII domain names, and testing whether the domain names behaved how the end user would expect (i.e., whether they came up as a hyperlink and connected to the correct URL). This received varying success and is displayed in the second chart in different shades of green, with lighter green representing the highest success. See the report for detailed testing results.

**Table 8: Social Media Application Testing Results Across Various Platforms**



For registration and authentication, findings show very poor support when using non-ASCII, internationalized email addresses. In contrast, the findings show generally very good support for the posting of both internationalized email addresses and IDN-based URLs using built in public posting and private messaging channels including RTL scripts. This is perhaps unsurprising due to the very high level of support for Unicode provided by modern browsers and mobile devices as well as server-side programming languages used to engineer these platforms. Future improvements and enhancements in this area for some social media applications might include the avoidance of displayed Punycode conversions visible in some circumstances, and additional work to support some of the language edge cases such as the



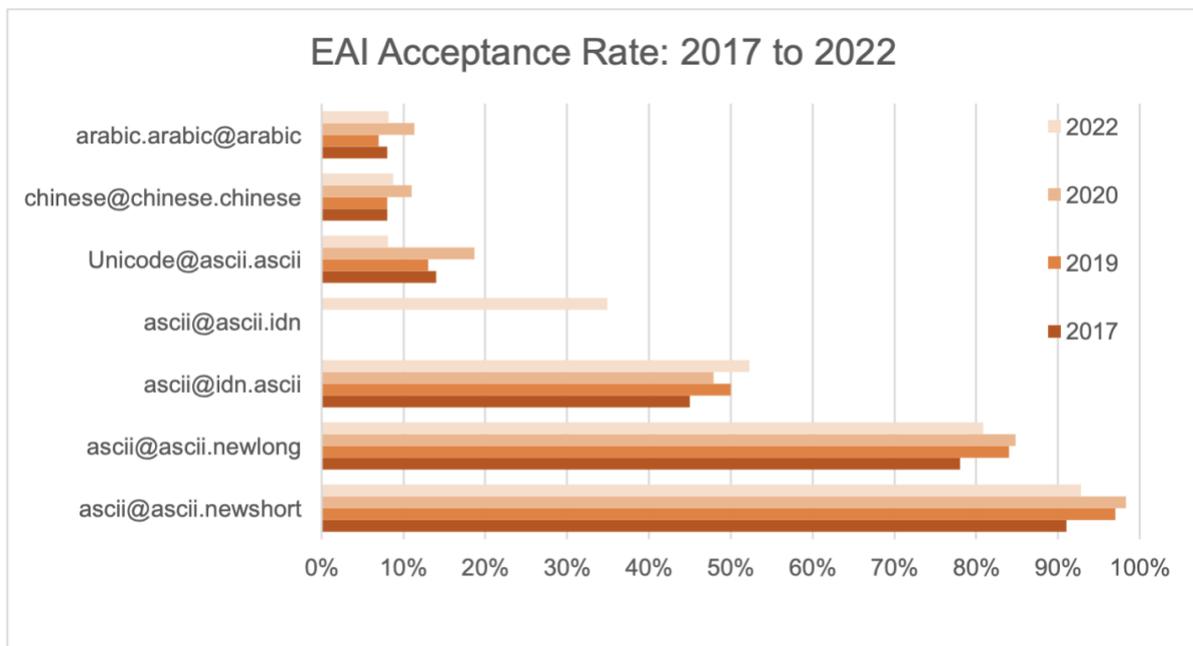
Chinese Open Dot which suffers from lack of support at the device level as well as in software.

## 6.8 EMAIL ACCEPTANCE BY WEBSITES

In 2017 and 2019, the UASG conducted studies to check how many of the top 1,000 global websites could accept email addresses based on a variety of TLDs, including new, long, and IDNs. The study also evaluated EAI with non-ASCII mailbox names represented in Unicode. The results showed that there is more work to be done before the websites are UA-ready. In 2020, the study was repeated with a slightly different design for choosing the 1,000 websites but with the same testing strategy.

The 2020 testing, [UASG027: Country-Based Evaluation of Websites for Accepting Email Addresses](#), did analysis of 50 popular websites in 20 different countries and provided the overall acceptance rates of different types of email addresses. It should be noted that this testing was limited to whether a website accepts a particular email address – it does not cover whether the website can store the email address or respond to it. The countries included in the current phase of the study are Argentina, Bahrain, Benin, Brazil, China, Egypt, Germany, Ghana, India, Israel, Japan, Kenya, Korea, Kuwait, Mexico, Nigeria, Russia, Sweden, Thailand, and Turkey.

In 2022, the study retested the 1,000 global websites as well as the 1,000 additional websites selected for the 20 different countries. Figure 7 compares the 2022 results to the earlier 2017, 2019 and 2020 testing results, noting that different email addresses were tested (but they were of the same category). These results show similar levels of acceptance, suggesting that the website developers are not upgrading their websites to accept UA-related features. The detailed results are reported in [UASG039: EAI Acceptance Rates of the Top 2,000 Global Websites in 2022](#).



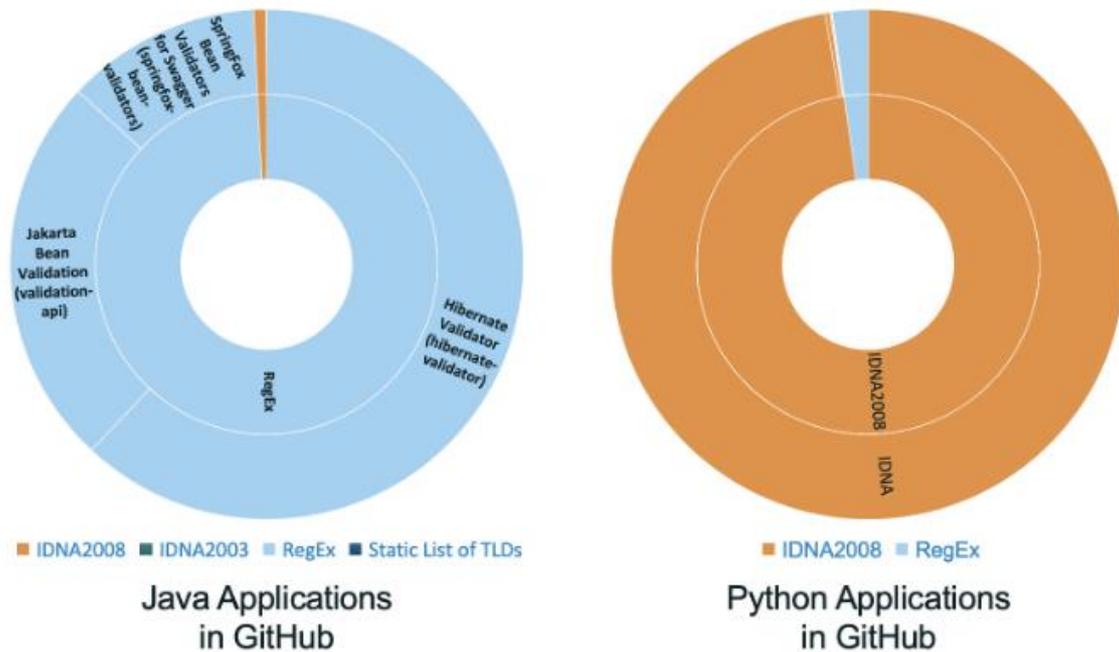
**Figure 7: Email Acceptance of 1,000 Global and 1,000 National Websites**

The process of reaching out to these website developers to inform and encourage them to address UA issues on their websites has been completed. A detailed report, [UASG046](#), was published in FY23. A summary is also provided in the Remediation section of this report.



## 6.9 UA-READINESS OF OPEN-SOURCE SYSTEMS

A survey of 10,000 popular open-source projects in GitHub shows that most of the applications in Java use outdated methods and libraries for domain names and email addresses. However, the applications in Python use libraries that provide UA support as indicated in Figure 7. Many Java-based applications use regular applications (regex) which are generally not UA-compliant and should be checked carefully before using for validating domain names or email addresses. It should be noted that the projects were not compiled and tested individually so it cannot be deduced if they are UA-compliant, though using a UA-compliant library is a step in that direction. A study with this data has been published as [UASG033: UA-Readiness of Open-Source Code](#).



**Figure 7: Usage of UA-Relevant Libraries in Open-Source Software**

Table 9 lists the UA-relevant libraries in popular use by the open-source projects surveyed in the study, along with their occurrence and UA-compliance status based on results in the [UASG018](#) report or their own documentation available online. Focusing remediation efforts on non-UA-compliant libraries, which are more frequently used, can create greater impact on UA-readiness.

**Table 9: Occurrence of UA-Relevant Libraries in Open-Source Projects Surveyed**

Language	Library	Occurrence in Projects	Status (Source)
Java	hibernate-validator	62963	IDNA2003 implied, RegEx via annotations; Hibernate implementation of validation-api. (Documentation)
Java	validation-api	25190	IDNA2003 implied, RegEx via annotations. (Documentation)



Java	Springfox-bean-validators	12501	IDNA2003 implied, RegEx via annotations; SpringFox implementation of validation-api. (Documentation)
Java	commons-validator	4906	Relies on a static list of TLDs from 2017. (UASG018A)
Java	icu4j	886	IDNA2008. (UASG018A)
Java	libidn	29	IDNA2003, deprecated and ported to the Java language as "java.net.IDN". (Documentation)
Python	idna module	70789	IDNA2008. (UASG018A)
Python	email_validator	1178	IDNA2008. (UASG018A)
Python	validators	1660	Email validation based on Django validator, not compliant; URL validation based on regex-weurl.js, which is a RegEx. (Code analysis)
Python	PyICU	243	IDNA2008. (Documentation)
Python	idna_ssl	10	IDNA2008. (Documentation)

## 6.10 UA-READINESS OF POPULAR IDENTITY PLATFORMS

Identity platforms are used to access many applications including accounts for ICANN, Google, Facebook, and GitHub. Usually, they require an email address for identification. The technical impact of the email address used here can be described as follows. Even if website users are trying to access an application that is UA-ready, access to the webpage is blocked if the identification (often known as single sign-on) tool is not UA-ready and cannot support an internationalized email address.

There is also a negative impact on the practical usage of internationalized email addresses with long TLDs in these platforms. If a user's email address is rejected, the user may either give up on using their EAI email address and revert to an ASCII email address with short TLDs, or they may give up on using the service.

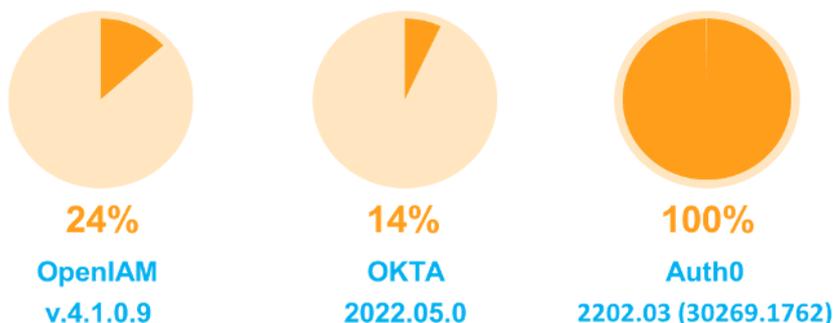
In FY23, the UASG published the study [UASG045 UA-Readiness of Popular Identity Platforms \(Okta, Auth0, OpenIAM\)](#). The objective of the research was to identify UA-related gaps in a few proprietary and open-source identity platforms. The integration of OKTA, Auth0, and OpenIAM identity platforms with the HyeConnect social media platform was implemented. Below are the supported versions.

### Identity Platforms

- Okta: version 2022.05.0
- Auth0: version 2202.03 (30269.1762)
- OpenIAM: version 4.2.1.2



The tests were implemented as automated QA running on the test platform of HyeConnect, written in Java. The following charts summarize the extent of UA support on the target platforms:



Out of 21 email addresses in different categories, UA-readiness testing was run through a workflow and test cases of which the details can be found in the [UASG045](#) report. Here are the results in summary:

1. Authentication platform Auth0 passed all email address validations throughout the given test flow.
2. Authentication platform OKTA passed only 3 email address validations and failed in 18 email addresses.
3. Authentication platform OpenIAM passed only 5 email address validations and failed in 16 email addresses.

## 6.11 EAI SUPPORT OF MAJOR EMAIL SOFTWARE AND SERVICES

Because email constitutes a significant part of online communications, it is important that email software and email service providers meet UA requirements. EAI is the protocol that allows email addresses to use Unicode-based mailbox names in UTF-8 format before the “@” sign and IDNs or ASCII domain names after the “@” sign. A series of studies have been conducted to evaluate the existing EAI capabilities of the email ecosystem, which have been published cumulatively in the most recent report in this series in 2021: [UASG030A: EAI Software Test Results](#).

The summary of EAI support in email tools from the report is presented in Table 10. Blank cells in the table indicate a component that does not exist. A tool or service is considered to have Level 1 (L1) support if it can send to or receive from an internationalized email address but cannot create an internationalized email address. Level 2 (L2) is assigned if the tool or service can also create an internationalized email address in addition to sending and receiving emails from such addresses.

**Table 10: EAI Support by Various Email Tools and Services**

Name	MUA	MSA	MTA	MDA	MSP	Webmail
Apple iOS Mail 14.x	Most L2					
Apple Mail	Few					
Coremail	Few	All L2	Most L2	Few	All L2	Most L2
Courier		All L2	All L2	All L2		
Dovecot				None*		



Exim		Most L2	All L2			
Gmail	All L1	All L1	All L1	Few		
Halon		Most L2	Most L2			
Mozilla Thunderbird	Few					
MS Exchange Server (hosted)		All L1	All L1	Few		
MS Outlook	Most L1					
MS Outlook.com	Most L1	Most L1	Most L1	None	None	Most L1
Postfix		All L2	All L2			
Roundcube	Most L2					
Sendmail 8.17 Alpha		Most L2	Most L2			
Thunderbird 89 beta	Most L1					
XgenPlus		Most L2	Most L2	Most	All L2	Most L2
Yandex Mail	Few	None	None	Few	Part L1	Few

\*Dovecot is a unique case as it does not support the EAI standard but can still provide L1 support due to its ability to handle different legacy character sets allowing it to support Unicode UTF-8 format.

The study shows that there is now good L1 support for EAI in many tools, while some tools are also providing L2 compliance.

This study also investigated the EAI support of a few spam filters, especially those that advertise EAI support. The aim was to develop test cases and do initial testing to finalize them. The results of the pilot testing are promising as two of the three systems tested provide good EAI support, as shown in Table 11.

**Table 11: EAI Support by Various Spam Filters**

Name	Spam
Spamassassin 3.4.5	All L2
Mailchannels	Part L1
Spamjadoo (Xgenplus)	All L2

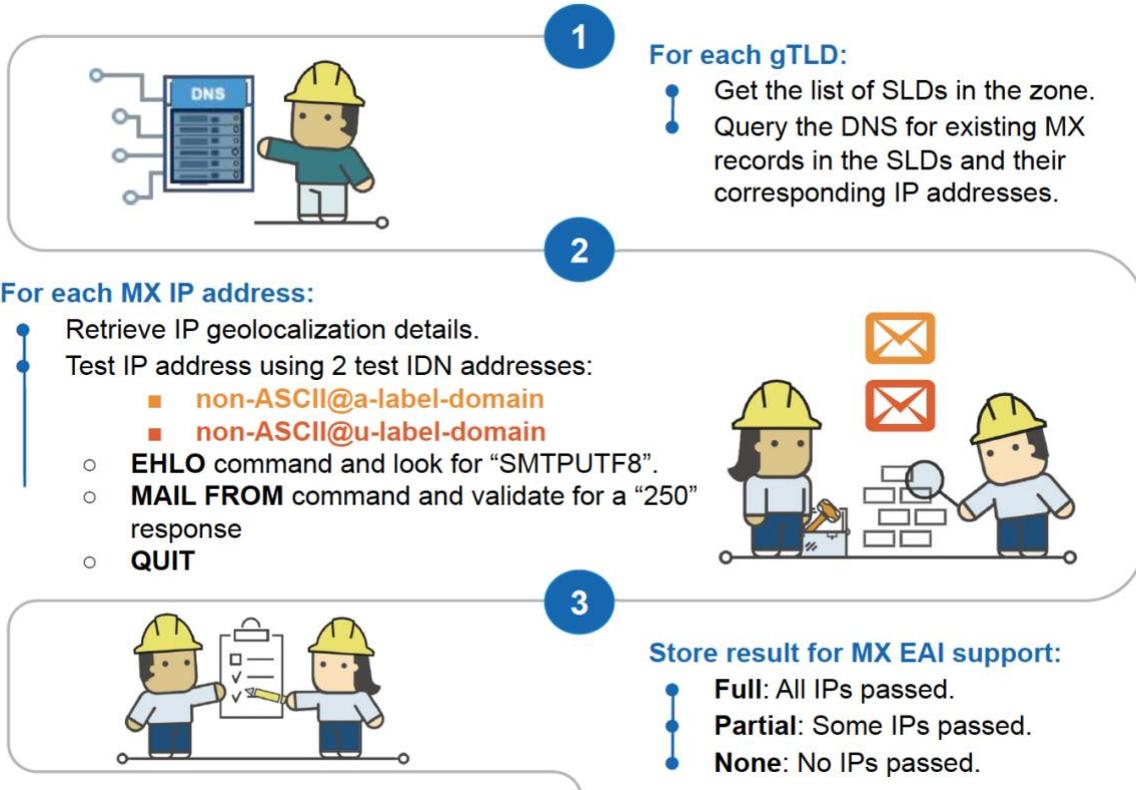
## 6.12 EAI-READINESS OF EMAIL SERVERS

The initial study [UASG021D: EAI-Readiness in TLDs](#), published in 2019, tested mail servers under a TLD that was responding with EAI support flag SMTPUTF8 on a request to connect. Details of the methodology are presented in the report. A small sample of domains were sampled which showed that 9.7% of the domains may be EAI-ready.

The study has been scaled up to cover all domain names registered under the gTLDs to get broader coverage and is repeated quarterly to gauge progress. It is done with a two-step

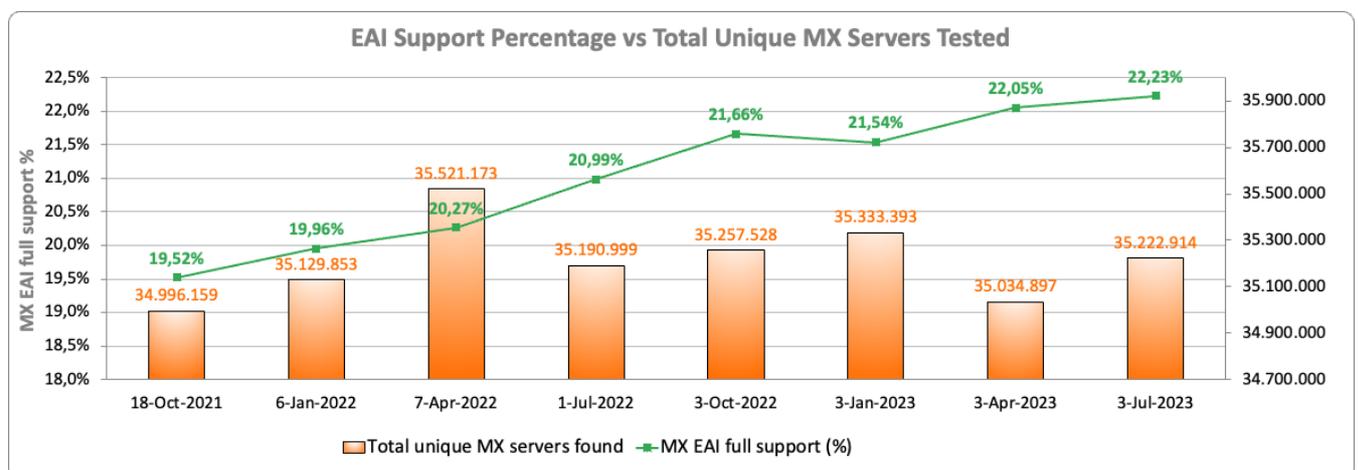


review process: (1) check how many mail servers respond with SMTPUTF8 flag on EHLO request; and (2), how many of these can accept a MAIL FROM request with an internationalized email address (with domain name part in either A-label or U-label format). The process is illustrated in Figure 8.



**Figure 8: Process to Determine EAI Support in Mail Servers**

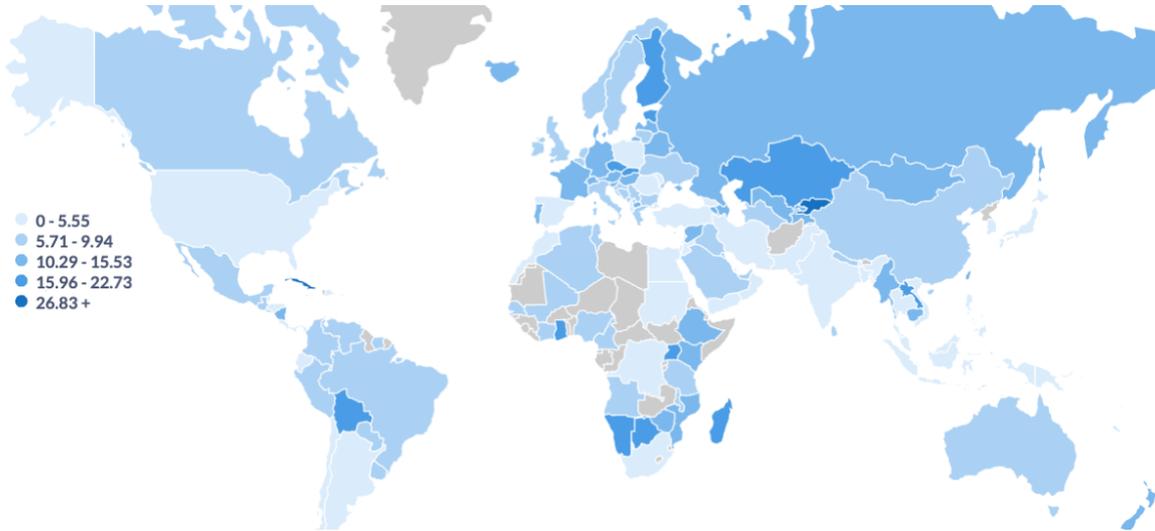
Figure 9 below shows EAI support (by percentage) in around 35 million unique email servers listed in about 1,200 gTLD zone files, as [published by ICANN](#). The acceptance rate has increased from 19.5% to 22.2% since October 2021, with nearly 78% of email servers not supportive of EAI. Email addresses in different scripts (Arabic, Cyrillic, Devanagari, and Han) were used in the tests.



**Figure 9: EAI Support in Mail Servers Listed in gTLD Zones**



Localizing the IPs shows the geographical distribution of email servers that support EAI in Figure 10 (as of 3 July 2023).



**Figure 10: Geographical Distribution of EAI Support in Mail Servers**

Table 12 presents the actual numbers for the top 15 geographies (using ISO 3166 codes) by IP addresses in the gTLD zones pointing to mail servers with the number of these IP addresses supporting EAI as of 3 July 2023.

**Table 12: EAI-Ready Mail Servers Across Geographies**

Country	IP total	IP passed EAI tests	EAI support %
US	906,859	36,300	4
DE	282,387	36,985	13.1
FR	132,066	14,957	11.33
JP	120,588	6,428	5.33
NL	102,777	7,980	7.76
GB	87,319	6,313	7.23
MD	85,556	146	0.17
CA	67,913	4,755	7
SG	49,923	2,337	4.68
ID	47,489	807	1.7
RU	43,847	6,717	15.32
ES	43,554	2,070	4.75
PL	41,463	2,031	4.9
CN	38,445	3,370	8.77
TR	36,690	1,208	3.29

### 6.13 UA-READINESS OF REGISTRY AND REGISTRAR SYSTEMS

To assess the UA-readiness of the systems of ICANN's contracted parties (gTLD registry operators and registrars), registry service providers (RSPs), and country-code top-level domains (ccTLDs), ICANN org conducted two surveys. The objective was to create a



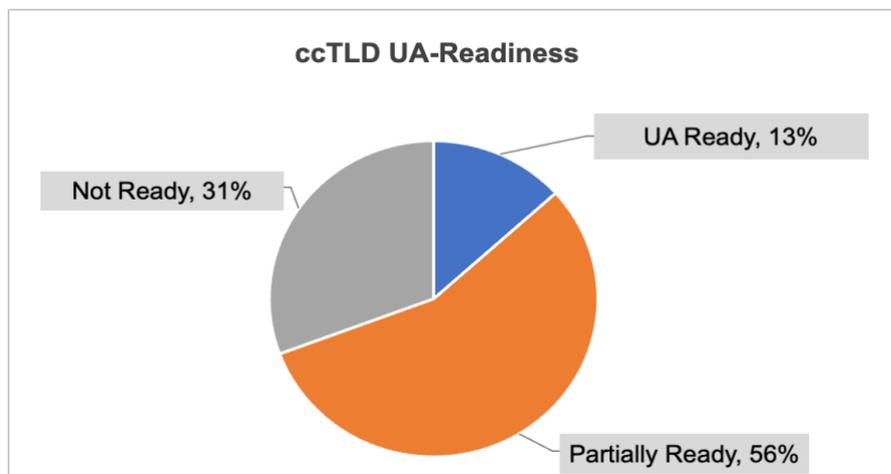
baseline to understand the challenges of becoming UA-ready and to understand how ICANN org can help.

- [UA-Readiness Survey Report – ccTLDs](#)
- [UA-Readiness Survey Report – Contracted Parties and Registry Service Providers](#)

### 6.13.1 UA-READINESS SURVEY REPORT – ccTLDs

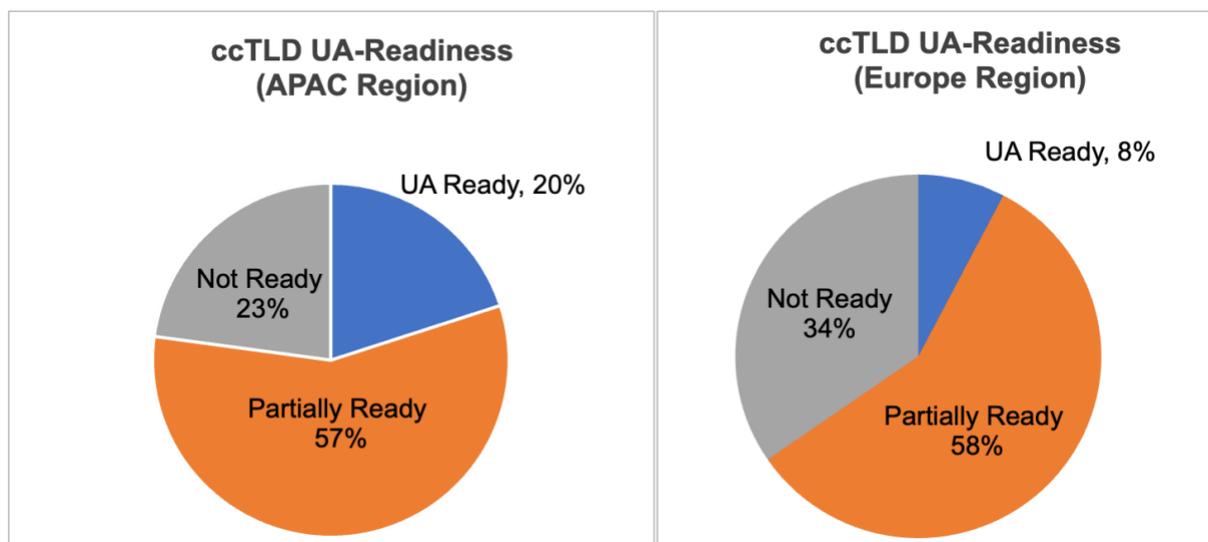
Most of the ccTLD managers who responded to the survey are UA-ready or partially UA-ready (69%). The main motivation to become UA-ready was to gain more customers, respond to customer needs, and support local language communities.

Figure 11 shows the distribution of UA-readiness for the ccTLDs who responded.



**Figure 11: ccTLD UA-Readiness (All)**

Figure 12 shows the distribution of ccTLD UA-readiness for the Asia Pacific (APAC) and Europe regions where there were more than 25% responses from the region.



**Figure 12: ccTLD UA-Readiness in the APAC and European Regions**

Both ASCII ccTLD and IDN ccTLD managers are experiencing similar barriers to becoming UA-ready. The main motivations to become UA-ready for ccTLD managers are to grow the



customer base, respond to customer requests, and to stay competitive. In addition, supporting local language communities is a key motivation for ccTLD managers.

Based on the results of this survey, the general outlook of UA-readiness of ccTLDs is positive, with many ccTLD managers being UA-ready or partially UA-ready (69%). For ccTLD managers who indicated that they are not UA-ready or are only partially UA-ready, 31% include UA-readiness in their roadmap and 50% of them plan to start working on UA-readiness within the next 12 months.

### 6.13.2 UA-READINESS OF CONTRACTED PARTIES AND REGISTRY SERVICE PROVIDERS

On 16 November 2022, invitations to participate in the UA-readiness survey were sent to 1,678 contracted parties and RSPs. Of the 1,678 organizations invited, 83 participated (18 registry operators, 55 registrars, and 8 RSPs), or 5% of the total group.

Figure 13 shows the distribution of UA-readiness for all respondents with 50% of ROs (10), 28% of registrars (15), and 75% (6) of RSPs stating that they are UA-ready, while 40% of ROs (8), 50% of registrars (27), and 25% of RSPs (2) stated they are partially UA-ready.

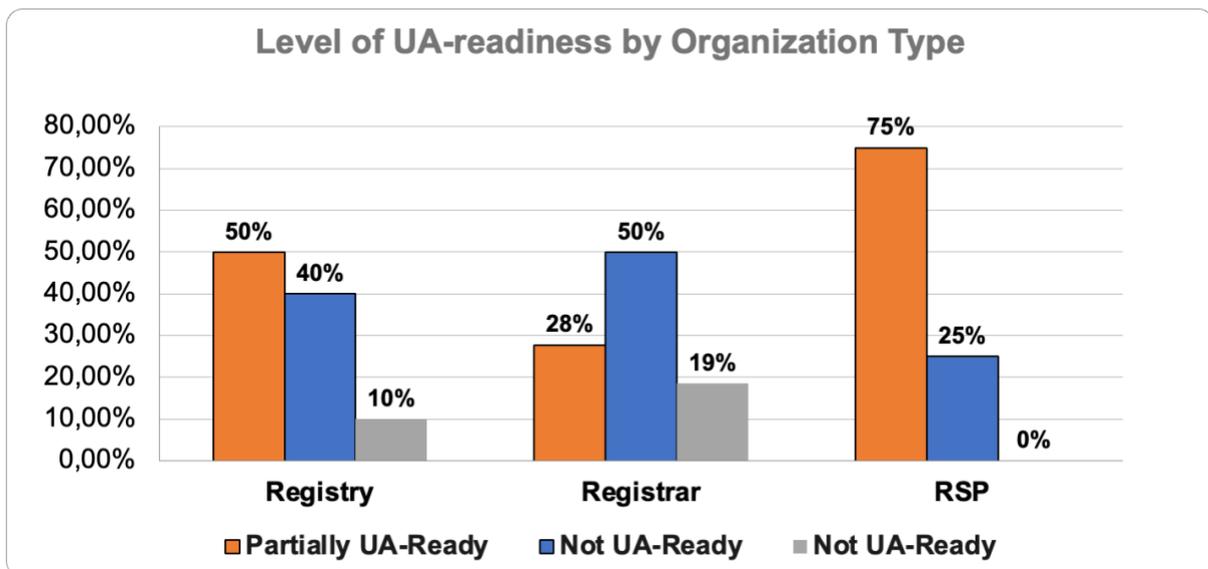
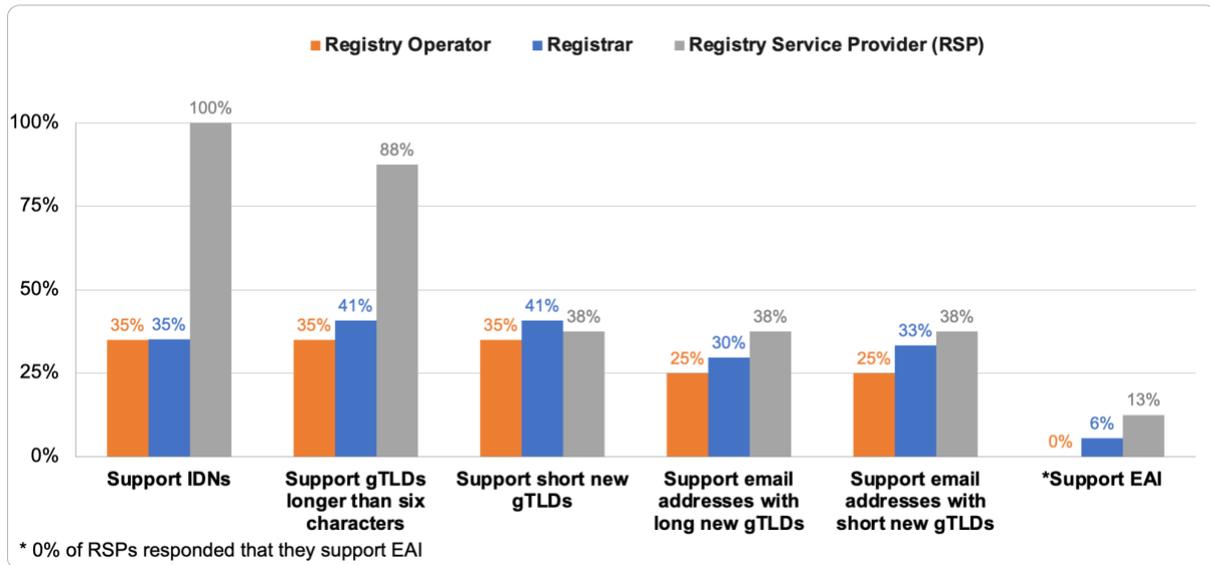


Figure 13: Level of UA-Readiness by Organization Type

Figure 14 represents the 8 registry operators, 27 registrars, and 2 RSPs that responded that they are partially UA-ready. The service supported the most across all entities is IDNs (57%) and the service supported the least is EAI.



**Figure 14: What Partially UA-Ready Organizations Support**

As part of this study, ICANN org wanted to understand what motivated organizations to become UA-ready. The primary motivations include:

- “To grow our customer base” (70%),
- “To be more competitive” (64%),
- “To gain more international customers” (51%),
- “In response to the requests of our customers” (43%)
- “Implement before the next round” (28%)

Note: respondents could select more than one response.

Despite the high percentage of organizations stating they are UA-ready, only a small percentage have created internal policies to support UA (7%) or have made their policies aware to third-party stakeholders.

Overall, it is evident that contracted parties and RSPs have made UA-readiness one of their priorities and have dedicated resources to becoming UA-ready with 84% of registry operators and registrars and 100% of RSP respondents stating they are UA-ready or partially UA-ready.

## 7. GUIDANCE, TRAINING MATERIALS, AND REMEDIATION FOR UA

The UASG working groups have been developing guidelines and training materials to address UA. After conducting gap analysis on many platforms, the UASG set out to fill these gaps in FY22. The focus then shifted to remediation work to fix the technical issues in software tools and applications in FY23. The remediation efforts will become even more significant moving forward as the gap analysis indicates what needs to be fixed. This section provides an overview of these activities undertaken by the UASG.

### 7.1 UA-READINESS FRAMEWORK

The UA of all domain names and email addresses requires that all software can accept, validate, process, store, and display them correctly. The [Universal Acceptance Readiness Framework](#) provides guidance on how to implement UA-readiness and test it using a gating



approach to verify UA conformance of an application (shown in [Figure 5](#)). More details on this are discussed earlier in this report.

## 7.2 NAMING INTERNATIONALIZED EMAIL MAILBOXES

Mailbox names – the part of the email address before the “@” symbol – are important as to how useful, user-friendly, and secure an email system will be. Therefore, it’s important that email administrators adopt good policies for naming mailboxes. Historically, email addresses have been limited to ASCII letters and digits, but now email technology has evolved to allow both mailbox names and domain names to be written in multiple languages and scripts. This flexibility makes policy choices more complex. The UASG has investigated and published guidelines for implementing mailbox names in its report on [Considerations for Naming Internationalized Email Mailboxes](#). This report aims to guide email administrators in adapting mailbox name policy to cover email addresses outside of Latin-script letters and digits.

Ultimately, the business purpose of email systems, along with the language needs of users and their correspondents, will decide policy choices and determine what scripts (writing systems for languages) will be allowed in mailbox names and their length. Since some spellings, character combinations, signs, or symbols are confusing or deceptive, the policy used by administrators should forbid them (details depend on the language used for the mailbox name). Technical issues about how names that look the same to users but might consist of different characters require special attention. Right-to-left (RTL) scripts have unique possibilities for confusing name spellings, so if your system allows these scripts there are additional policy topics that should be addressed.

Users may exchange emails with people who have trouble reading the particular script used in an email address. To reduce these difficulties, implementing a policy that allows users to choose display names (the personal name displayed) and use aliases (alternate addresses for the same user) is recommended. The [UASG028](#) document also includes resources and references that provide more detail on several of these policy topics. It also includes a glossary of terms, some of which may be new to administrators when adopting multiple scripts and languages for mailbox names.

## 7.3 ROLE OF CCTLDS IN ACHIEVING UA-READINESS

The [Asia Pacific Top Level Domain Association](#) (APTLD) partnered with ICANN and conducted a [UA training program](#) for members of the Asia Pacific (APAC) community in 2021. As part of the series of webinars, the final one convened a panel of ccTLD managers to discuss the role of ccTLDs in achieving UA-readiness.

The panel discussion presented a consensus about the significant role ccTLD managers need to play in supporting and promoting UA. First and foremost, ccTLD managers should make their own systems UA-ready, especially in local languages. In addition, ccTLD managers should play an active role in promoting UA in their communities by working with a broad range of stakeholders, planning outreach, and conducting technical training activities for stakeholders. There are many national, regional, and global efforts already underway to promote UA adoption resulting in readily available resources and best practices for both ccTLDs and other stakeholders. Working to achieve UA is an evolving process so ccTLD managers should stay up to date on the latest UA resources and initiatives as they work to make UA-readiness a reality in their local communities.

The panel’s recommendations have been compiled into a [White Paper on The Role of ccTLDs in Achieving Universal Acceptance](#). The white paper can help other ccTLDs managers achieve UA-readiness as well.



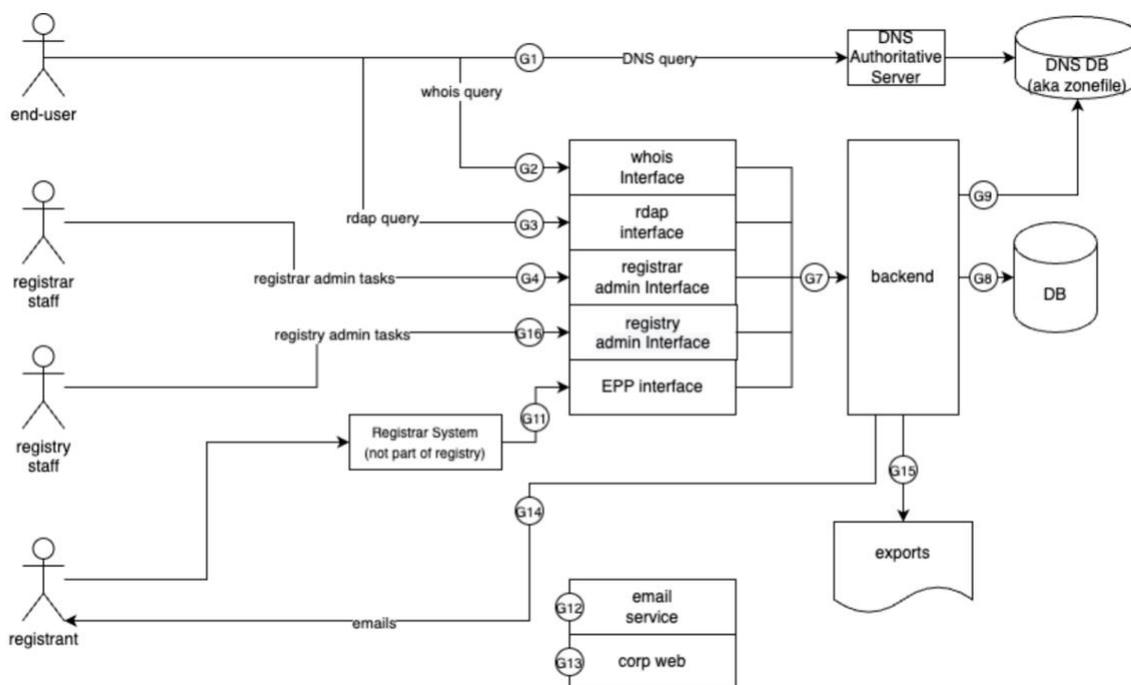
## 7.4 ACHIEVING UA-READINESS IN REGISTRY AND REGISTRAR SYSTEMS

Software applications must be built or updated to support UA capabilities. Among those applications are the registration of domain names, such as domain name registry and registrar software systems. This study is of technical nature and targeted to registry and registrar operators, registry backend providers, developers, and technical managers. The UA roadmap is a technical study serving the purposes of raising awareness of UA and to assist relevant parties with becoming UA-ready.

While this work can apply to both gTLD and ccTLD registry and registrar systems, the specifics also include the requirements of the ICANN gTLD environment, such as required reports and data escrow. Protocols and interfaces to the protocols involved in the registration ecosystem, such as Extensible Provisioning Protocol (EPP), WHOIS, and Registration Data Access protocol (RDAP) are addressed. Finally, considerations for properly testing such systems are outlined with results of such testing for actual registry and registrar systems available in the appendices.

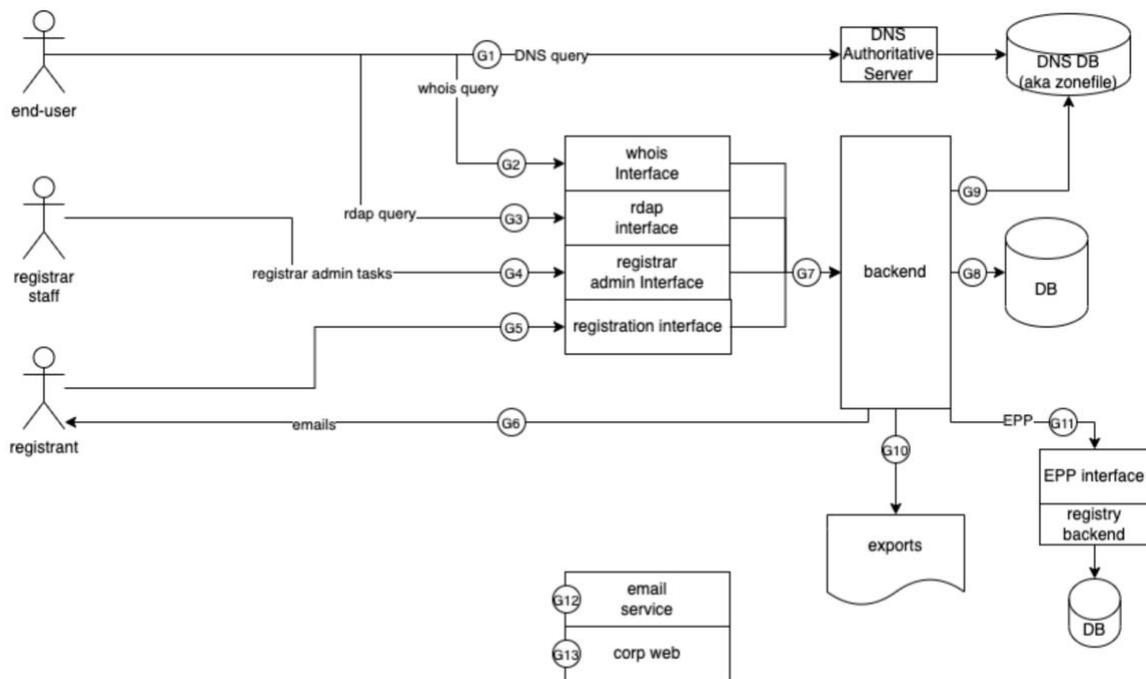
- [Universal Acceptance \(UA\) Roadmap for Domain Name Registry and Registrar Systems](#)
- [Appendix A: Registry Testing](#)
- [Appendix B: Registrar Testing](#)

For a domain name registry, there are primarily four different personas: the Internet end-user, the registrant, the registrar staff, and the registry staff. Figure 15 illustrates an example of a domain name registry high-level system architecture.



**Figure 15: Domain Name Registry High-Level System Architecture**

For a domain registrar, there are primarily three different personas: the Internet end-user, the registrant, and the registrar staff. Figure 16 illustrates an example of a domain registrar high-level system architecture.



**Figure 16: Domain Registrar High-Level System Architecture**

The UA-Readiness gating approach (Gx) is integrated into the UA Roadmap for systematically evaluating UA conformance in registry and registrar systems. See the report and appendices for further details.

## 7.5 CASE STUDIES ON UA ADOPTION

The UASG publishes case studies to document how an organization can promote UA and EAI adoption to provide motivation and guidance for others.

In 2022, the Thai Network Information Center Foundation (THNIC) published a [case study](#) to share how it is working with the community to promote UA training. THNIC organized a GitHub hackathon focused on UA. In this week-long virtual event, developers competed to modify a series of popular, public GitHub repositories to be UA-ready. The event resulted in the successful merging of one of the pull requests into the GitHub origin repository, as well as more widespread awareness among the Thai developer community of the importance of UA and the steps needed to update systems to be UA-ready.

In 2021, a case study was published on Coremail and THNIC's process towards EAI-readiness. While Coremail and THNIC's EAI journeys were unique, there were common key learnings that were captured in the study: [Supporting a Culturally and Linguistically Diverse Internet Through Email Address Internationalization \(EAI\)](#).

ICANN org has also been assessing and updating its systems since 2017 to ensure all its systems become UA-ready. A [case study](#) was published in 2020 on ICANN's journey to UA-readiness, which is divided into three stages:

- Stage 1: Support both new short and long ASCII TLDs
- Stage 2: Support IDNs
- Stage 3: Support non-ASCII email addresses



ICANN org has completed the first stage and is in the process of accomplishing Stage 2. Work has also progressed to achieve Stage 3 goals by working with other relevant organizations to make their own email systems support EAI.

## 7.6 UA TRAINING MATERIALS

### 7.6.1 UA TECHNICAL TRAINING

The UASG has developed technical training materials for relevant UA stakeholders. Three distinct training modules are now available on the UASG's [training wiki page](#). The first training provides an overall technical overview of UA and how to support it for technology managers. The second training provides technical details on how to configure email servers for supporting and deploying email addresses in local languages and scripts (i.e., EAI) for email and system administrators. The third training covers the technical details on how to program websites and applications with UA support aimed at software developers. The training originally used examples from the Java programming language and was enhanced in 2022 to cover the Python programming language as well.

For UA Day 2023, training materials were updated and branded for UA Day [here](#).

Another training on the role of ccTLDs was also developed based on the joint work with the APTLD published in the [White Paper on The Role of ccTLDs in Achieving Universal Acceptance](#).

These training materials are summarized in Table 13.

**Table 13: Summary of UA Training Materials**

Training	Audience	Description	Duration
UA Technical Overview	<ul style="list-style-type: none"><li>- CIOs</li><li>- IT managers</li><li>- System administrators</li><li>- Software developers</li><li>- Email administrators</li></ul>	An overview about UA and EAI readiness issues	1.5 hours
Configuring Email Address Internationalization	<ul style="list-style-type: none"><li>- Email administrators</li><li>- System administrators</li><li>- IT managers</li></ul>	Technical configuration and setup of EAI supported email service	3 hours
UA for Programmers and Software developers	<ul style="list-style-type: none"><li>- Software developers</li><li>- IT managers</li></ul>	Develop current best practices for UA compliant Java and Python applications	3 hours
Role of ccTLDs in Achieving UA	<ul style="list-style-type: none"><li>- ccTLD Managers</li></ul>	Role of ccTLDs in adopting and promoting UA, based on the <a href="#">White Paper on The Role of ccTLDs in Achieving Universal Acceptance</a>	1.5 hours



## 7.6.2 UA COURSE ON ICANN LEARN

An Introduction to Universal Acceptance (UA) course was developed for the [ICANN Learn platform](#) and launched in August 2021. The course is intended for a wide audience including the general Internet community, technology enablers and developers, email service providers, governments, and policymakers. By completing the course, participants will develop a basic understanding of UA and its technical challenges, the current state of UA-readiness globally, and the economic and social benefits of being UA-ready. The course also serves as a great resource for people who want to get involved in regional and global efforts to address UA, and for those who want to make their own systems UA-ready.

## 7.7 UA REMEDIATION EFFORTS

UA gap analysis of existing technology and applications deployed has identified areas that need to be remediated. As reasons for issues become more evident, the focus has started shifting towards fixing them. The remediation efforts can be categorized into three areas:

1. Direct outreach to tool and framework providers, either through the bug-reporting channels available and/or more directly to the organizations or open-source communities maintaining them.
2. Direct outreach to end-user application or service providers, either through the online channels available and/or more directly to the organizations offering them.
3. Interact with the broader developer community on software developer networks and conferences.

ICANN is reaching out to technology companies that have applications with a large online user base to encourage them to become UA-ready. To facilitate these discussions, ICANN prepares examples of UA-related issues in their internet applications, software, and platforms, as identified by UA gap analyses done by UASG. These issues are communicated to the contacts in these organizations to fix and update.

### 7.7.1 INTERACTING WITH TOOL AND FRAMEWORK PROVIDERS

Bugs are generally reported through bug reporting systems online. As part of the testing work on PHP, Swift, and Kotlin platforms in [UASG037](#) and more directly by ICANN org, the issues found were put in as bug reports on the relevant channels. A summary of these reports is given in Table 14 below. As highlighted in **green**, four of these have been resolved.

**Table 14: Summary of Bug Reports Filed**

Language	Platform	Library	Report	Resolution
Swift	iOS	MessageUI	<a href="#">StackOverflow report</a>	No answer yet
Swift	iOS	URLSession & Alamofire	<a href="#">StackOverflow report</a>	Being discussed
Swift	iOS	URLSession & Alamofire	<a href="#">Bug report on Swift bugtracker</a>	Assigned to a maintainer
PHP	Windows	mail	<a href="#">Bug report on RFC6531 compliance</a>	Changed to feature request
PHP	Windows	cURL	<a href="#">Bug report on IDNA 2008 compliance</a>	This is a Windows issue
PHP	Windows	cURL	<a href="#">Bug report on IDN 2008 compliance</a>	No answer yet



PHP	Windows & Linux	intl	<a href="#">Bug report on IDNA 2008 compliance</a>	Suspended as this is an ICU issue
PHP	Windows & Linux	PHP Mailer	<a href="#">Bug report for IDNA 2008 compliance</a>	Pull request submitted and merged upstream
PHP	Windows & Linux	Symfony HttpClient	<a href="#">Bug report on IDNA 2008 compliance</a>	Pull request submitted and merged upstream
PHP	Windows & Linux	Symfony Mailer	<a href="#">Bug report on IDNA 2008 compliance</a>	Pull request submitted and merged upstream
PHP	Windows & Linux	Symfony Mailer	<a href="#">Bug report on RFC5322 compliance</a>	Being discussed
PHP	Windows & Linux	Symfony Mailer	<a href="#">Bug report on RFC6531 compliance</a>	No answer yet
Kotlin	Android	Jakarta Mail	<a href="#">Bug report on RFC6531 compliance</a>	No answer yet
Kotlin	Android	URLConnection	<a href="#">Bug report on IDNA 2008 compliance</a>	Transmitted to the engineering teams
Kotlin	Android	OkHttp	<a href="#">Bug report on IDNA 2008 compliance</a>	Solved
Kotlin	Android	Fuel	<a href="#">Bug report on IDNA 2008 compliance</a>	Pull request submitted and pending to be merged upstream
Kotlin	Android	Apache HttpClient	<a href="#">Apache HttpClient</a>	No answer yet

Earlier, different email software was tested as part of [UASG030A](#). The systems tested did not have ticketing systems to report bugs, so in such cases, their developers were contacted. Minor bugs were found while testing the sendmail MTA and MSA and Mailchannels mail filter. Those bugs were reported to the developers by email and in each case, they responded and fixed the bugs promptly.

Postfix and Exim MTAs, which failed a few of the tests done as part of UASG030A, were also contacted. In those cases, the authors disagreed with the interpretation of the standards and did not change their software (the issues are minor and do not affect EAI compatibility for most users.) A small error in the “Received” header added by Courier was also communicated which the developer has fixed. While developing the Python test scripts, one bug and one limitation were found that were fixed locally. The bugs were then reported, and patches were provided.

During FY23, ICANN increased its focus on UA remediation. As part of this work, ICANN provided patches for several widely-used programming languages, libraries, and email tools. ICANN has found and provided patches for three UA-related bugs in the Ruby programming language, one concerning sending email and two concerning either receiving or acting in email. All three have been accepted by the team and will be available to all Ruby programs in the next version of the language. ICANN has found and provided patches for two UA-related bugs in the Python programming language, one concerning reading email and one concerning passwords. Both are still outstanding. ICANN has also provided patches for a



UA problem in the Java programming language concerning IDNA2008 compliance and is developing a suitable disclaimer. The following table lists the patches provided, with status as of August 2023.

Programming Language	Status	URL
Ruby	Accepted	<a href="https://github.com/ruby/net-imap/pull/114">https://github.com/ruby/net-imap/pull/114</a>
Ruby	Accepted	<a href="https://github.com/ruby/net-imap/pull/111">https://github.com/ruby/net-imap/pull/111</a>
Ruby	Accepted	<a href="https://github.com/ruby/net-smtp/pull/49">https://github.com/ruby/net-smtp/pull/49</a>
Python	Pending	<a href="https://github.com/python/cpython/pull/107290">https://github.com/python/cpython/pull/107290</a>
Python	Pending	<a href="https://github.com/python/cpython/pull/103611">https://github.com/python/cpython/pull/103611</a>
Java	Waiting for ICANN's disclaimer	<a href="https://github.com/openjdk/jdk/pull/14812">https://github.com/openjdk/jdk/pull/14812</a>

ICANN also provided patches for the Dovecot mail server, Roundcube webmail system, and Thunderbird mail client, all of which are widely used. So far none of the patches have been accepted by the maintainers. The following table lists the patches:

Mail Software	Status	URL
Dovecot	Pending; the company sees no customer demand.	<a href="https://github.com/dovecot/core/pull/190">https://github.com/dovecot/core/pull/190</a>
Roundcube	Pending; status unknown	<a href="https://github.com/roundcube/roundcubemail/pull/8925">https://github.com/roundcube/roundcubemail/pull/8925</a>
Thunderbird	Pending; probably has a very low urgency within the maintenance team	<a href="https://phabricator.services.mozilla.com/D176259">https://phabricator.services.mozilla.com/D176259</a>

The recent web hosting tools study (UASG042) reported the following six bug reports:

Management Software	Status	URL
cPanel	Not planned	<a href="https://features.cpanel.net/to/pic/22490">https://features.cpanel.net/to/pic/22490</a>
cPanel	Not planned	<a href="https://features.cpanel.net/to/pic/22491">https://features.cpanel.net/to/pic/22491</a>
cPanel	Not planned	<a href="https://features.cpanel.net/to/pic/22492">https://features.cpanel.net/to/pic/22492</a>
cPanel	Not planned	<a href="https://features.cpanel.net/to">https://features.cpanel.net/to</a>



		<a href="#">pic/22493</a>
Plesk	Pending	<a href="https://talk.plesk.com/threads/.365646/">https://talk.plesk.com/threads/.365646/</a>
Plesk	Pending	<a href="https://talk.plesk.com/threads/.365647/">https://talk.plesk.com/threads/.365647/</a>

The issues with OKTA and OpenIAM were reported to the identity platform providers. More details about the support tickets can be found in the [UASG045](#) report.

### 7.7.2 INTERACTING WITH END-USER APPLICATION SERVICE PROVIDERS

ICANN Global Support provides support to the UASG by receiving and triaging cases based on issues submitted through <https://uasg.tech> website reporting page and reaching out to service providers. Whenever reports indicate that a digital service does not support UA, ICANN Global Support attempts to contact their customer support or other teams that support those services. The two key activities are to inform and educate. Some of the organizations are not aware of UA and therefore the team provides relevant information and resources to explain the scope of UA issues and recommended solutions. From 1 July 2022 to 30 June 2023, a total of 31 UA cases were reported. These 31 cases created within the system resulted in 10 outreach cases. The Global Support team reached out to the various domain name holders to make them aware of the concerns that were reported.

The UASG reached out to the top 2,000 websites which were studied in [UASG039: EAI Acceptance Rates of the Top 2,000 Global Websites in 2022](#). The relevant generic email IDs for these websites were contacted to report the issue that they were not supporting internationalized email addresses. The UA Remediation for Top Global Websites report, [UASG046](#), was published in FY23. Though the mail delivery success rate was around 85% with an open rate of 60%, 25% of responses were automatic. There were hardly any responses from the organizations, and the few received were mostly auto-generated, which is not encouraging. Since the email campaigns did not result in positive responses, a different strategy for more engagement with the website authorities and developers was followed.

More than 26 online and offline awareness/remediation workshops and coding sessions were conducted which resulted in responses from 56 organizations for possible UA implementation. The project team initiated one-to-one communication with the organizations who attended awareness/remediation online workshops and have shown interest in UA implementation. Organizational-level talks were initiated and carried out for UA awareness and compliance. This also resulted in organizations showing interest in UA. As a result of this remediation strategy, 22 out of 56 websites became UA-ready.

A summary of these websites is given in Table 15 below.

**Table 15: Summary of Websites Subject to Close Remediation Project**

Category	Number	Status
Unable to proceed further	19	– Website developers backed out from implementation without citing any reason.



Unable to evaluate	7	<ul style="list-style-type: none"><li>- Problem while sending the form.</li><li>- Email field not found.</li><li>- Email field not available, however only Google form was available.</li><li>- Provided Facebook URL instead of website.</li></ul>
Unable to implement fully	8	Contact form 7 plugin related issues
UA-ready	22	Result of remediation efforts undertaken by project team.
<b>TOTAL</b>	<b>56</b>	

The UASG046 report focuses on the outcomes of remediation activities as well as the UA remediation process undertaken to make websites UA-ready. It includes:

- Understanding the platform and technology stack used while developing the websites.
- Possible changes for making the website UA-ready.
- Suggesting the changes along with suitable process and use of requisite library.

### 7.7.3 INTERACTING ON DEVELOPER NETWORKS

One of the major reasons behind the issues and challenges of UA is lack of awareness from software developers developing and deploying end-user applications and services. There are some decisions made at the microlevel that often make a big difference for UA support, e.g., the character set that needs to be allowed while validating a domain name field. Software design documents, however detailed, can often leave these decisions to the software developer. It is critical that the developer community at-large, regardless of the programming language they use, is made aware of these challenges. One of the major sources from which a developer gathers this know-how is from their “Discussion Forums,” web-platforms that are tailor made for facilitating these conversations.

Typically, on such forums, developers ask questions about things they need immediate help with or perspective. These forums have active user communities that engage with each other and take the discourse ahead. “Stackoverflow,” a popular forum, has been active since 2008 and has several such problem-solving conversations. Many questions have been asked on such forums that have a direct bearing on UA. If the conversation and discourse around it is not up to date, then new developers will keep making the same mistakes, resulting in continued UA problems.

Through the [EAI Technical Education and Awareness Directed at Developer Community Websites \(UASG 044\)](#) project, that was published in FY23, six such prominent platforms were analyzed and engaged with. The analysis phase involved identification of relevant forums, identification of relevant topics/tags, and identification of relevant questions. The engagement phase involved creating model answers for the identified questions and posting the answers on the relevant forums. The objective is also to make sure that the relevant responses gain enough traction to get into the active discourse of the developer community.

Table 16 presents a summary list of the questions and answers and forums engaged with so far. More relevant questions and answers can be found in [UASG044](#) and [UASG044A](#).



**Table 16: Outreach to Developers on Development Platforms**

No.	Question	Forum
1.	<a href="#">How to use IDN domain with Swedish åäö when sharing on Facebook?</a>	Stackoverflow
2.	<a href="#">Example URIs for unit testing IDN domains</a>	Stackoverflow
3.	<a href="#">Nodejs Detect Punycode IDN language</a>	Stackoverflow
4.	<a href="#">What is a regular expression which will match a valid domain name without a subdomain?</a>	Stackoverflow
5.	<a href="#">Validation code for IDN (Domain) with regex (regular expression)</a>	Stackoverflow
6.	<a href="#">IDN (Domain) validation with regular expression</a>	Stackoverflow
7.	<a href="#">Domain Name Regex Including IDN Characters c#</a>	Stackoverflow
8.	<a href="#">non-latin email address validation</a>	Stackoverflow
9.	<a href="#">How to validate an e-mail address in swift?</a>	Stackoverflow
10.	<a href="#">How to validate an email address in PHP</a>	Stackoverflow
11.	<a href="#">How can I validate an email address using a regular expression?</a>	Stackoverflow
12.	<a href="#">How can I validate an email address in JavaScript?</a>	Stackoverflow
13.	<a href="#">C# code to validate email address</a>	Stackoverflow
14.	<a href="#">How to validate an e-mail address in swift?</a>	Stackoverflow
15.	<a href="#">How to validate an email address in PHP</a>	Stackoverflow
16.	<a href="#">Unicode Ranges of Indian Language Characters</a>	Stackoverflow
17.	<a href="#">Your email validation logic is wrong</a>	Stackoverflow
18.	<a href="#">Email Address Validation: Please Stop</a>	Stackoverflow
19.	<a href="#">It is impossible to validate an email address</a>	Stackoverflow
20.	<a href="#">Properly validating e-mail addresses</a>	Stackoverflow
21.	<a href="#">Case sensitive hostnames</a>	Stackoverflow
22.	<a href="#">.htaccess file, blocking IDN domains</a>	Stackoverflow
23.	<a href="#">Is there a list of Unicode character categories included/excluded in IDNs?</a>	Stackoverflow
24.	<a href="#">Validating email addresses using jQuery and regex</a>	Stackoverflow
25.	<a href="#">Which Unicode characters are allowed in IDN host labels?</a>	Stackoverflow
26.	<a href="#">Your email validation logic is wrong</a>	Reddit
27.	<a href="#">Email Address Validation: Please Stop</a>	Reddit
28.	<a href="#">It is impossible to validate an email address</a>	Reddit
29.	<a href="#">Properly validating e-mail addresses</a>	Reddit
30.	<a href="#">Is my new gTLD causing it to get filtered as spam?</a>	Serverfault
31.	<a href="#">Case sensitive hostnames</a>	Serverfault

As a part of this exercise, Frequently Asked Questions (FAQs) were identified. A list of such questions along with the model answers was published in [UASG044A](#), which will soon be incorporated into the FAQs section at <https://uasg.tech>.

The following are the actual questions per their respective categories.

1. Character/Length Specific
  1. What is the maximum length of a valid email address?
  2. What is the maximum length of a valid domain name?
  3. How to detect language/script of a Punycode?
  4. Can I treat all the domain names (IDNs and ASCII-based) to be IDNs and process them as such without any ill-effects in my code flow?
  5. Are ASCII symbols prohibited altogether from IDNs?



6. I see a special character in an IDN that I want to type out. How should I type it?
2. General Validation Specific
  1. What is the maximum length of a valid email address?
  2. What is the maximum length of a valid domain name?
  3. What is the simplest regular expression to validate emails to not accept them blindly?
  4. Are ASCII symbols prohibited altogether from IDNs?
3. Programming Language Specific
  1. What is the level of support for IDNs in Android in terms of UA-related requirements?
  2. What is the level of support for IDNs in iOS in terms of UA-related requirements?
4. General IDN/EAI Protocol Specific
  1. Are IDNs case-sensitive?
  2. Are internationalized email addresses case-sensitive?
  3. How to detect language/script of a Punycode?
  4. Is there a way to avoid showing "xn--" for IDNs?
  5. How to compare Punycode for visible similarity?
  6. Is the whole URL required to be converted to Punycode or only the domain name part?
  7. Can I treat all domain names (IDNs and ASCII-based) to be IDNs and process them as such without any ill-effects in my code flow?
  8. While converting IDN to ASCII (or vice-versa) in any programming language, can the whole URL (with protocol parameters such as HTTPS, colons, etc.) be submitted as it is?
  9. Are IDN conversion functions reversible?
  10. Just like there is a list of valid TLDs, where can I find a list of valid email TLDs?

In addition to the questions, two informative blog posts were written by a contractor on two important aspects of UA:

1. ["Universal Acceptance of Domain Names and E-mail Addresses" - What is the issue?](#)
2. [What is the ultimate goal of the Domain Name and Email ID Validation?](#)

Following up on UASG044, the table below lists questions posted after the publication of UASG044, along with answers provided by ICANN.

1.	<a href="#">An official or defacto EXAMPLE.COM for IDNs?</a>	Stackoverflow
2.	<a href="#">Any way to show IDNs correctly?</a>	Stackoverflow
3.	<a href="#">There is something wrong with my Puny code</a>	Stackoverflow
4.	<a href="#">Convert non-ASCII email addresses to punycode in java</a>	Stackoverflow



## 8. UA LOCAL INITIATIVES

The [UA Local Initiatives](#) work in close consultation with UASG leadership and working groups, and with the support and collaboration of ICANN org. The purpose of these initiatives is to plan and undertake outreach to and collaborate with local stakeholders identified by the UASG in their region to promote UA-readiness. The initiatives also engage with local stakeholders to encourage them to become UA-ready. These include initiatives in China, Commonwealth of Independent States and Eastern Europe (CIS-EE), India, Sri Lanka, and Thailand.

The [Chinese Domain Names Initiative](#) (CDNI) of the [Internet Society of China](#) (ISC) was established on 9 January 2020. With the leadership of CDNI and the support of ICANN, the UASG, and three UA Ambassadors, a series of outreach, training and industry engagement sessions with stakeholders were carried out to actively promote UA-readiness.

The support of Chinese domain names by Internet applications is varied. According to the [Test Report](#) of Chinese Internationalized Domain Names conducted by CDNI, the overall support rate for the Universal Acceptance of Chinese domain names reaches 35%.

Specifically, the support rate of browsers for Chinese domain names has made big progress. PC browser support reaches 92% and mobile browsers 80%. The UA-readiness of search engines keeps improving and is currently 61%. Some technical support has been realized, but in actual application there is sometimes low inclusion or trans coding display rates. EAI-readiness is not yet common, but the QQ Mailbox (new system) has set a good example to realize EAI support.

Applications that support Chinese domain names include:

- Email: QQ Mailbox and Wo Mailbox
- Search Engines: Baidu and 360
- Browsers: QQ Browser and 360 Browser
- Mobile Browsers: UC Browser (Android and IOS), Xiaomi, OPPO
- Main Application Platforms: WeChat Official Account
- SNS: Zhihu and Ding Ding
- Short Video Platform: Kuaishou Android
- Development Platform: WeChat Official Account

These applications are able to fully identify Chinese domain names, carry out link redirections, and search engines include Chinese domain name websites and display them normally. The current challenge is how to further facilitate and motivate more Internet service providers to adopt UA-readiness through market engagement and cultivation. The full report, including methodology and other applications tested, can be found [here](#).

The Commonwealth of Independent States (CIS) and Eastern Europe (EE) UA Local Initiative was set up by the regional community in 2019. During FY23, it gathered experts from nine countries: Armenia, Belarus, Georgia, Latvia, Russia, Serbia, Ukraine with new members from Azerbaijan and Kyrgyzstan. The members are mostly representatives of IDN ccTLD and new gTLD registries, registrars, and local IT companies from the region. For UA remediation, because of the local initiative's dialogue with local software developers, few local software products have been checked and started to work on their UA-readiness in FY23, e.g., ISPmanager (web hosting panel), Angie (web server), Mail.ru (mail service) and some others. EAI testbeds are now available for local developers in Armenia and Russia.



According to recent [research](#), more than 259,000 mail servers already announce SMTPUTF8 support (among those pointed as MX for delegated SLDs in ccTLDs .RU and .РФ). Quick guides on how to copy-paste URLs with Unicode symbols, type sign “@” in EAI addresses without switching a keyboard, make correct linkification of IDN/EAI in applications, and other explainer materials were published on the local portal [поддерживаю.рф](#) (in Russian). The [guidelines](#) on linkification for URLs with non-ASCII characters were developed and shared with the UASG community.

The [FICCI-Indian Language Internet Alliance](#) started hosting the UA Local Initiative in India in 2019. The Indian Government has shown support for Universal Acceptance and a multilingual Internet and has made a clear plan for UA implementation that can be a role model for other governments. <https://www.registry.in/system/files/ua-and-multilingual-internet-india.pdf>.

In October 2022, Theekshana, a research and development company, established a [UA Local Initiative in Sri Lanka](#). This initiative aims to raise awareness of the IDN ccTLDs and address technological gaps within the country. LK Domain Registry released reserved domain name email.lk and its equivalent in Sinhala script ඉතපාල්.ලංකා and Tamil script இதபால்.இலங்கை to Theekshana to set up an EAI-ready email platform. At the UA Day celebrations on 28 March 2023, the Sri Lanka Local Initiative launched two EAI supporting email platforms in Sri Lanka. UA adoption has begun in a small way and on 24 August 2023, the equivalent of gov.lk in Sinhala script, රජය.ලංකා, was delegated to ICTA by LK Domain Registry to issue government organizations Sinhala domain names pointing to Sinhala content for the first time.

The Thailand UA Local Initiative is spearheaded by the Thai Network Information Center Foundation ([THNIC](#)). In FY23, the local initiative sought cooperation with Thailand government agencies to assist in formulating policies or utilizing IDN domains, as well as developing Thai-language email systems that are under consideration. THNIC has sought collaboration with universities, namely Thammasat University and Kasetsart University, to offer courses for developing email systems that support local languages or EAI as part of the curriculum. This is currently under consideration.



## 9. LOOKING AHEAD

The UASG recently finalized and published its [FY24 Action Plan](#). Although the studies to analyze technology gaps in UA continue, the focus of work has shifted towards UA remediation during FY23. The remediation work focuses on the technical community developing underlying tools and frameworks as well as those programmers and system administrators involved in developing and deploying end-user software applications and email services. For the former, the focus will be to submit bug reports and engage with them to help prioritize the fixes. For the latter, outreach and training is the focus as it is a much broader set of stakeholders.

The UASG has been able to make progress on a local level through its UA Local Initiatives and UA Ambassadors. The UA Local Initiatives provide the right connection to the community to raise awareness and train stakeholders. Therefore, the UASG aims to expand the program and looks forward to supporting additional initiatives, especially in the Latin American and Caribbean (LAC) and European regions, while continuing to support those already established in China, CIS-EE, India, Sri Lanka, and Thailand. Similarly, the UASG will continue to conduct outreach through its UA Ambassador Program and UA Day (<https://universalacceptance.day>).

The UASG Measurement Working Group completed and published two reports in FY23: [UASG042](#) web hosting tools report and [UASG045](#) identity platforms report. In addition, the working group is planning to launch an IT-related [UA Curricula for Academic Programs](#) and outreach to academia. In FY24, the WG will continue to prioritize other technologies for measurement-related work, which could include a second and more comprehensive phase on CMS. The working group also aims to study the HTML5 standard and its use to see the extent of its impact on UA. The WG is working on a white paper to advocate for IDNA 2008 support over IDNA 2003, targeting software developers that handle domain names.

The UASG Technology Working Group completed and published two reports in FY23: [UASG043](#) code samples supporting UA and [UASG046](#) website remediation report. The working group will continue focusing on technology remediation that will include more technology stacks and training materials, and other technologies like CMS and web hosting tools. The working group has been designing a survey to determine the technical challenges and opportunities in adopting UA and plans to conduct a series of surveys in 2023. The working group will also contribute to academic curriculum development to include IDNs and UA.

The EAI Working Group completed and published two reports in FY23: [UASG044](#) - EAI Technical Education and Awareness Directed at Developer Community Websites, and [UASG044A](#) - Proposed FAQs. The working group finished the first phase of the EAI self-certification guide and updated the guide with input received from early adopters. The guide is now evolving into a EAI Self-Certification Program that aims to develop a tool to assist with generating an EAI support level score and self-assessment. It will focus on developing success stories on EAI implementation to create greater enthusiasm and to showcase working solutions. The working group also aims to create user-centric documentation on how to find a service provider and use/test that provider's services, and provide system administrators with working EAI setups, configurations, and tests for self-hosting. The working group also wants to reach out to global technology leaders like Facebook, eBay, and others to encourage them to support EAI.

The UASG Communications Working Group assisted in Internet Global Forum (IGF) and UA Day sessions held at both the regional and global levels during FY23. The working group



plans to coordinate with other UA working groups to make sure their output is effectively disseminated to the right audiences by publishing blogs, case studies, videos, while also actively engaging with the community through its social media channels. The working group also aims to develop a cohesive strategy for outreach at national, regional, and global IGFs. The working group continues to maintain and improve the [UASG.tech](https://uasg.tech) website experience and is also planning to develop short step-by-step instructional videos explaining different aspects of UA to the technical community.

The UASG leadership team is focused on diversifying the geographical expansion of UA Local Initiatives. For FY24, five more UA Ambassadors will be joining the UASG to contribute to local training and outreach activities. UASG leaders will be coordinating and planning for the second annual UA Day in early 2024. The first UA Day was held on 28 March 2023 and marked the first time a diverse set of technical and language communities, companies, governments, and DNS industry stakeholders mobilized to champion UA and a multilingual Internet on a global scale.

ICANN plans to continue its UA remediation efforts, including submitting bug reports and even patches to key applications, libraries, and programming languages. ICANN will also focus on interoperability testing to discover problems proactively and expand its UA training and outreach programs to provide UA-related expertise to ISPs, registrars, university students, and more. In cooperation with other organizations, ICANN is open to organizing workshops and hackathons on UA-related subjects and will keep engaging with tech companies with large online user bases to promote UA readiness. ICANN will also engage with standards organizations and work to improve the UA-friendliness of relevant standards as they are updated.

Achieving UA is a collective responsibility, and it will only be fully achieved if we make all our own systems UA-ready. With an abundance of literature, materials, training, and code examples now available through the UASG, we hope that the community will also engage and help implement UA adoption.

## 10. ENGAGE WITH THE UASG

- For further information about the UASG visit <https://uasg.tech/>
- Contact the UASG at <https://uasg.tech/contact/> or email [info@uasg.tech](mailto:info@uasg.tech)
- Meet the people of the UASG: <https://uasg.tech/about/people/>
- Join UASG working groups: <https://uasg.tech/join/>
- Participate in UA general discussions: <https://uasg.tech/subscribe>
- Report UA problems with other applications: <https://uasg.tech/global-support-centre/>
- Follow the UASG on its social channels: [Twitter](#) | [Facebook](#) | [LinkedIn](#) | [Youtube](#)