

Universal Acceptance Testing Guide: Regional Evaluation of Websites for Acceptance of E-mail Addresses [Draft]

Maintainer: Mark W. Datysgeld

Purpose

This is a methodology guide for local UA initiatives that are interested in performing compliance tests with the forms of websites from their region. These instructions conform to but expand to some degree those presented in “UASG025: Global Evaluation of Websites for Acceptance of E-mail Addresses in 2019”, generating results that are compatible and enable cross-reference of datasets.

PHASE 1: Preparation

While the number of websites tested for the global evaluation is 1,000, the suggested number for local initiatives is 100. This is a number that is manageable for a smaller initiative, but still meaningful enough to produce a general impression of UA compliance in the region.

The data used for these tests comes from the competitive analysis tool Alexa. A list of top websites broken down by country can be found here:

<https://www.alexa.com/topsites/countries>

The free version of Alexa will only output the top 50 results for a given country, which is insufficient. A cost-effective approach is to **register** to the service for a month and make local copies of data from all countries in the region, creating a reference dataset for a given year and enabling further evaluations without needing to pay again.

These data then need to be **scraped** by whatever method the team deems more adequate. For coders, the Alexa API is probably the best option available, while for other teams it might be better to use software such as OutWit Hub to extract the data from the tables.

The data points that need to be scraped are Ranking and Domain Name, with the other data not being useful for this test. This should be done to the top 300 websites, for safety.

The data needs to be **inputted** into a spreadsheet in whatever office suite the team finds the most suitable. Eventually the Global Evaluation team intends to create a unified online database for these tests, but as of the writing of this document it is still a project.

The **naming** of that file should follow the convention “ua_<scope>_<region>_<year>”. For example, a survey carried out in Mexico in 2020 should be named “ua_regional_Mexico_2020”.

The spreadsheet should contain the following columns:

domain	url	rank	testable	code	comments	eai	box1	box2	box3	box4	box5	box6
--------	-----	------	----------	------	----------	-----	------	------	------	------	------	------

This is what each column is supposed to track:

- **domain:** Domain, not prefixed by “www”.
- **url:** URL containing the full path to the page containing the form.
- **rank:** Ranking of the website in Alexa within that dataset.
- **testable:** Indicates if the team managed to test the website.
- **code:** Contains the code that validates the form, if found.
- **comments:** Any relevant comment.
- **EAI:** Denotes if the website was found to be EAI-compatible.
- **mailboxes:** List of ratings for each test case.

Once the spreadsheet is ready, a **screening** process needs to be done, with the following criteria:

- The content **needs to be regional**, not necessarily to a specific country, but to the subregion it is in; “facebook.com” and “google.com” are not relevant, but “lanacion.com.ar” and “indiamart.com” are.
- The website’s main purpose cannot be the spread of malware or spam.
- The website needs to have an e-mail input field available somewhere in their pages that the tester can interact with.

Questions of content are generally not accounted for if the website passes the criteria established above. It is not the job of Universal Acceptance and its collaborators to act as content police, making decisions of what is worthy of evaluation or not. Of course, regional sensibilities should be accounted for. If some material is disallowed by law in the testers’ region and other such cases, it is ultimately at their own discretion to remove the entries from their dataset, preferably but not necessarily including a note in the test report about the criteria for exclusion.

A **list of URLs** needs to be appended to the initial scraped data containing the actual address of where the testable forms are located. For instance, if the contact form for the website “test.org” is located at “https://www.test.org/contact/form.aspx”, that is the data for its URL field.

Then, a set of 6 test mailboxes needs to be **generated** and set up for use by the team, each having an increasing level of complexity in relation to the other. This can be

coordinated to some degree with local service providers, but a compatible Registry will likely need to be found for the generation of the RTL address, for example. These mailboxes have to follow the patterns detailed below.

Test Case	Format	Example
New gTLD short	ascii@ascii.newshort	test@test.exp
New gTLD long	ascii@ascii.newlong	test@test.example
IDN	ascii@idn.ascii	test@普遍接受-测试.org
Unicode simple	unicode@ascii.ascii	测试1@test.org
Unicode complex	unicode@idn.idn	测试5@普遍接受-测试.世界
Right-to-Left (RTL)	rtl.rtl@rtl	دون@رسيل.السعودية

The standard script for **Unicode** testing has been Han, but many other suitable options are available, as long as they fall outside of ASCII's "A to Z" limitation.

The standard script for **RTL** testing has been Arabic, but other suitable options are: Hebrew/ Samaritan, N'Ko/Adlam, Syriac/Mandaic, and Thaana.

Further help with the mailbox setup process can be sought at the "UA-Discuss" mailing list, where qualified volunteers are available to provide support.

PHASE 2: Testing

The next step in the procedure is the **testing of the e-mail forms**.

The tester should **navigate** to each URL from the finalized list and as their first measure, and then use the "**view source code**" function of their browser to look for code relevant to e-mail validation. This can come in the form of:

- **Regular Expressions:** an old method of validation that relies on comparing the e-mail the user typed with a usually narrow set of characters detailed in JavaScript. It usually results in low UA compliance. [The code should be copied to the dataset.](#)
 - Sample code:

```
var emailExp = new RegExp(/^\b[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\b$/i);
```
- **HTML5:** known not to be compliant with UA as of 2019, and can be identified in the body of the code by the tag `<input type="email">`. [This can be simply marked as "HTML5" in the dataset.](#)

- **Plug-ins:** software packages made to validate forms, usually loaded in the head section of the code. “Validate.js” is a well-known plug-in of this kind. [The name of the plug-in and if possible, the version number should be noted on the dataset.](#)
- **Unknown:** sometimes the procedure does not become clear even after evaluation.

After that is done, the tester manually inserts the address of one mailbox at a time on the forms identified on the previous phase, submitting them to be processed by the websites.

Each time this is accomplished, ratings of Accepted or Rejected are **assigned** according to the following criteria:

Marked as Accepted when:

- ✓ Submission results in a success message, such as “E-mail has been registered”.
- ✓ Submission goes through and no error is reported.
- ✓ An “already registered” e-mail message is displayed; this is often the result of the website sharing a database with another one that has already been tested.

Marked as Rejected when:

- ⊗ The website displays an error once the address is inputted on the form.
- ⊗ An error message is reported upon submission.
- ⊗ Submission is not allowed.

At the end of 2 cycles, this section of the spreadsheet should look like this example:

Website	Box 1	Box 2	Box 3	Box 4	Box 5	Box 6
test.org	Accepted	Accepted	Accepted	Rejected	Rejected	Rejected
ページ.日本	Accepted	Accepted	Accepted	Accepted	Accepted	Rejected

In case a form claims success in sending an e-mail to the “ascii@idn.ascii” mailbox, a final step should be taken, which is to **check** if a message actually arrived in the mailbox. This has the aim of identifying if the server is compliant with Email Address Internationalization (EAI-ready). Since an answer may take hours or in some cases days,

this check is best performed one week after the initial testing, making it the final data point that to be added to the spreadsheet.

PHASE 3: Report

With those results in hand, all that is left is to **analyze** them. It is left up to each team to understand how to better present their findings in ways that are relevant to their region. However, for the best use of data to be achieved, the spreadsheet should be **sent** to the UASG leadership, so that it can be compiled and compared with data from other regions, eventually enriching Global reports.

The report produced by the team should strive to clearly show their findings, and also present a comparison between the overall regional results and whichever is the latest Global survey, in order for a sense of comparison to be established.

OVERVIEW: Step-by-step Guide

Phase 1

- Register to obtain Alexa's paid statistics.
- Scrape the top websites from the region.
- Create a spreadsheet following the project's conventions.
- Input the top 300 websites into the spreadsheet.
- Screen the websites to arrive at 100 testable regional URLs.
- Generate 6 mailboxes following the project's model.

Phase 2

- Navigate to each URL from the finalized list to perform the tests.
- View the source code and look for any code relevant to e-mail validation.
- Test the e-mail forms of selected URLs against the 6 mailboxes.
- After one week, check the "ascii@idn.ascii" mailbox for confirmation messages and responses, potentially finding out if the server is EAI-ready.

Phase 3

- Analyze the results obtained.
- Send the finalized spreadsheet to UASG leadership.
- Produce a report that is compatible with the region's expectations.