

Background

Universal Acceptance (UA) is the state where all valid domain names and email addresses are accepted, validated, stored, processed and displayed correctly and consistently by all Internet-enabled applications, devices and systems.

Due to the rapidly changing domain name landscape, many systems do not recognize or appropriately process all domain names, primarily because they may be more than three characters in length or in a non-ASCII format. The same is true for email addresses that incorporate these new extensions or might have non-ascii mailbox names.

The Universal Acceptance Steering Group (UASG), supported by Internet Corporation for Assigned Names and Numbers (ICANN), is a community-led, Internet industry-wide initiative working on creating awareness and identifying and resolving problems associated with the universal acceptance of all domain names and email addresses. The purpose of these efforts is to help ensure a consistent and positive experience for Internet users globally.

For more information on the UASG and recent development, visit www.uasg.tech.

Linkification and Its Challenges

Modern software sometimes automatically creates a hyperlink by a user simply typing in a string that looks like a web address, email name or network path. For example, typing “www.icann.org” into an email message may result in a clickable link to <http://www.icann.org> being automatically created if the application treats “www.” as a special prefix or “.org” as a special suffix.

Linkification should work consistently for all well-formed web addresses, email names or network paths.

Linkification is the action where an application accepts a string and dynamically determines whether it should create a hyperlink to an Internet Location (URL/IRL) or an email address (mailto:). While browsers, websites, messaging application, email clients and word processors are obvious places, there are many more applications that make these decisions.

Linkification uses algorithms and rules created by software developers to determine whether a string should be deemed a link – or not. Related to this is how people might interpret a string as

a domain name^[DH1] or email address. Users have expectations and developers need to code for those expectations.

Principles of Linkification

The following principles should apply when determining when to create a link:

1. **Standards Principle.** Linkify well-formed URLs per RFCs 1738¹ and 3987.
2. **Universal Acceptance Principle.** Accept, Validate, Store, Process and Display all presumed domain names and email addresses with the same rules (e.g. do not discriminate based on the top-level domain or script).
3. **User Expectations Principle.** Users expect consistent behaviour from their applications. If you have your own rules to transform presumed domain names and email addresses to URLs based on context or intent, apply rules consistently (i.e. see Universal Acceptance Principle).
4. **Safe Practice Principle.** Links should NOT be created from strings that are “Highly Restrictive” according to the Unicode Technical Standard #39²

Good Practice Recommendations

1. Attempt to linkify based on explicit protocol prefixes (e.g. “http://”, <https://>, ftp://”, “mailto:”) and the rest of the string is well formed.
2. Attempt to linkify based on implicit protocol prefixes (e.g. “www” infers “<http://www>”) but only complete the action if the rest of the string is well formed.
3. Map the Ideographic Full Stop “。” (U+3002) to Full Stop “.” (U+002E) (e.g. <http://田中。com> to <http://田中.com>) if the string is otherwise well formed.
4. When a Top Level Domain is used as a Special Suffix to determine linkification, the TLD should be validated either in real-time through the DNS or from a table that is updated from an authoritative source at least daily.
5. Linkification should be determined by the implied intent of the user's entry.

¹ <https://www.ietf.org/rfc/rfc1738.txt>

² http://www.unicode.org/reports/tr39/#Restriction_Level_Detection

6. Linkification should be done only at the time of creation of the content, which includes editing of existing content.

7. No middleware application or viewer application should generate links.

8. Users should be able to disable a link once a link is automatically created.

9. Links should NOT be created where a domain name label contains uncharacteristic mixed scripts or might otherwise be unsafe. See <https://www.m3aawg.org/sites/default/files/m3aawg-unicode-best-practices-2016-02.pdf>

Example String	Expected Behaviour
example.com	Linkify because the protocol (http://) is inferred and there are two labels connected by a “.”
मेल.डाटामेल.भारत	Linkify because the protocol (http://) is inferred and there are three labels each connected by a “.”
http://example.technology	Create hyperlink because protocol is explicit and string is well formed.
https://example.build	Create hyperlink because protocol is explicit (secured) and string is well formed.
http://xn--r2bi6d.xn--c2bdw5c1bb.xn--h2brj9c/	Create hyperlink because protocol is explicit and string is well formed. The domain name is input Punycode Format and should be converted to Unicode format
http:example.com	Do NOT Linkify. String is not well formed. No linkification because of bad syntax (missing //)
<u>http://example.a</u>	String is not well formed because of ICANN Policies require a TLD to be at least two characters.
<u>http://example..abc</u>	No linkification because of bad syntax (consecutive dots). String is not well formed

http://普遍接受-测试.世界	Create hyperlink because protocol is explicit and string is well formed.
http://192.168.10.15	Create hyperlink because protocol is explicit and string is well formed (in this case an IPv4 address)
https://2001:4860:4860::8888	Create hyperlink because protocol is explicit and string is well formed (in this case an IPv6 address)
mailbox@example.berlin	Create mailto: mailbox@example.berlin because protocol is implied by an email format.
ftp://xn--11b1b9b0ah0f.xn--h2brj9c/ Or ftp://निकसी.भारत	Create hyperlink because protocol is explicit and string is well formed.

[1] Note: it might be the case that the actual website requires that end users type https:// instead of http://. If this is the case, then the hyperlink may not resolve or may return an error page.
