

การพัฒนาโปรแกรมให้รองรับ Universal Acceptance

พจนันท์ รัตนไชยพันธ์ และ สมเกียรติ ปุ้ยสูงเนิน

Thailand UA Local Initiative
21 สิงหาคม 2564

Training/Workshop ด้านการยอมรับชื่อโดเมนและอีเมลภาษาไทย

- ◎ 7 มีนาคม 2564 การปรับระบบอีเมลให้รองรับการรับส่งชื่ออีเมลภาษาไทย (systems)
- ◎ 12 มิถุนายน 2564 การพัฒนาเว็บให้รองรับ UA (web developers)
- ◎ 21 สิงหาคม 2564 การพัฒนาโปรแกรมให้รองรับ UA ด้วย Java (Programmer, DevOps)

ติดตามรายละเอียด <https://academy.thnic.or.th/webinars-on-eai/>

หัวข้อ

- Introduction to Universal Acceptance (20 mins)
- Fundamentals for IDNs and Email Address Internationalization (EAI) (20 mins)
 - Break (5 mins)
- Programming for UA (60 mins)
- Conclusion (5 mins)
- Q&A (10 mins)

Introduction to Universal Acceptance

Universal Acceptance of Domain Names and Email

Goal

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

ชื่อโดเมนและอีเมลทุกชื่อสามารถใช้งานได้กับทุกซอฟต์แวร์แอปพลิเคชัน

Impact

Promote consumer choice, improve competition, and provide broader access to end users.

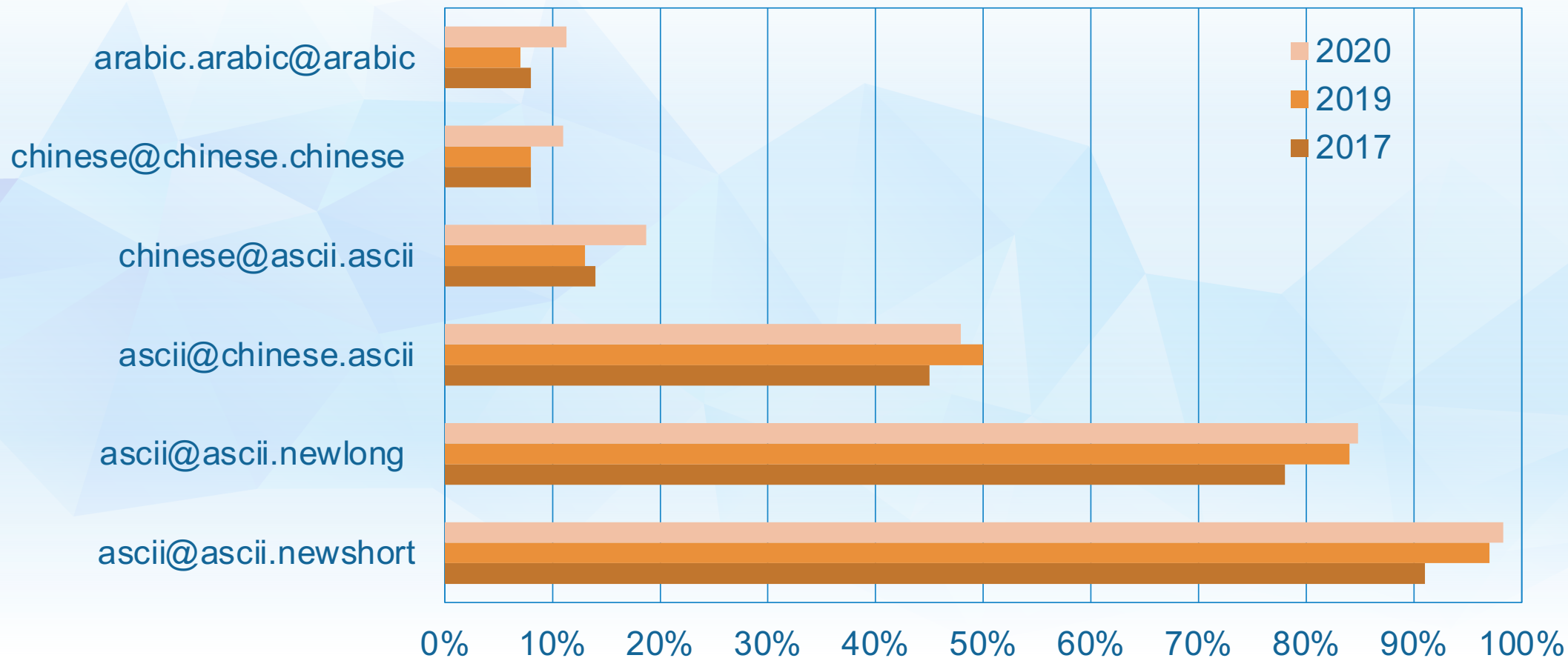
เพิ่มทางเลือก เกิดการแข่งขัน และเปิดการเข้าถึง [ซอฟต์แวร์แอปพลิเคชัน] ให้แก่ผู้ใช้งาน

Categories of Domain Names and Email Addresses

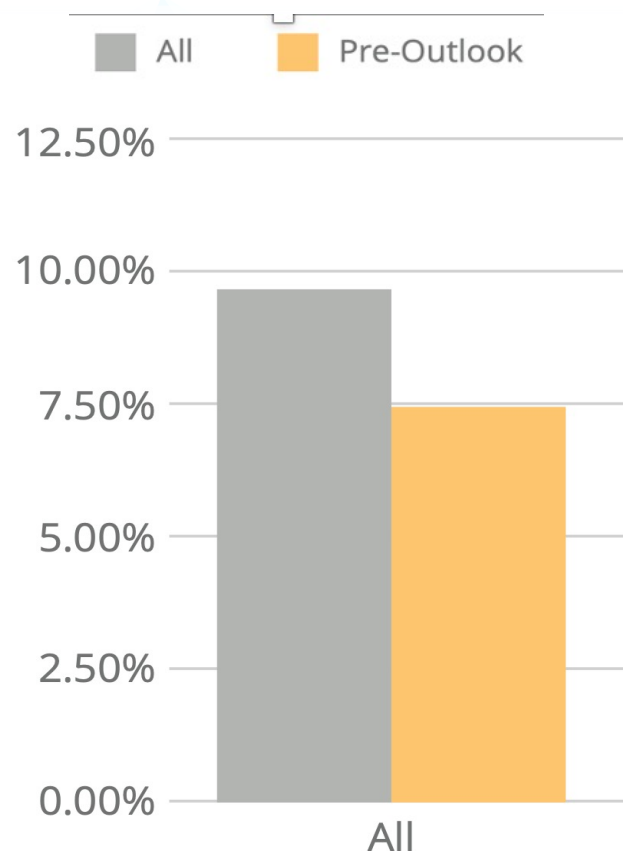
- ◉ It's now possible to have domain names and email addresses in local languages using UTF8.
 - Internationalized Domain Names (IDNs)
 - Email Address Internationalization (EAI)
- ◉ Domain names
 - **Newer** top-level domain names: example.sky
 - **Longer** top-level domain names: example.abudhabi
 - **Internationalized** Domain Names: รู้จัก.ไทย
- ◉ Internationalized email addresses (EAI)
 - ASCII@IDN marc@société.org
 - UTF8@ASCII ईमेल@example.com
 - UTF8@IDN พจนันท์@คน.ไทย
 - UTF@IDN; right-to-left scripts ای-میل@مثال.موقع

Acceptance of Email Addresses by Websites Globally

For details, see [UASG027](#)



Estimated Support of EAI in Email Systems Under All TLDs



Only 9.7% of the domains sampled were EAI ready in 2019.

This is based on mail servers found through MX records in zones of all top-level domains (TLDs).

For details on methodology, see [UASG021D: EAI Readiness in TLDs](#)

Acceptance of Thai Email Addresses by Thai Popular Websites

ปี	จำนวนเว็บไซต์ทดสอบ	จำนวนเว็บไซต์รับชื่ออีเมล (A)	จำนวนเว็บไซต์รับชื่ออีเมลภาษาไทย (B)	จำนวนเว็บไซต์ตอบกลับชื่ออีเมลภาษาไทย (C)	อัตราการรับชื่ออีเมลภาษาไทย (B)/(A)	อัตราการตอบกลับชื่ออีเมลภาษาไทย (C)/(A)
2561	100	44	4	1	9%	2%
2564	100	43	2	1	5%	2%

(A) จำนวนเว็บไซต์รับสมัครสมาชิก รับติดต่อผ่านฟอร์ม หรือ ให้สมัคร mailing list ด้วยชื่ออีเมล

(B) จำนวนเว็บไซต์รับสมัครสมาชิก รับติดต่อผ่านฟอร์ม หรือ ให้สมัคร mailing list ด้วยชื่ออีเมลภาษาไทย

ทดสอบเว็บไซต์ที่เป็นที่นิยมในไทยจากการจัดอันดับเว็บไซต์ของ Alexa.com และ TrueHits

Universal Acceptance of Domain Names and Emails

การที่ชื่อโดเมนและชื่ออีเมลทุกชื่อที่ถูกต้องตามกฎหมายเกณฑ์
สามารถใช้งานได้โดยซอฟต์แวร์ประยุกต์ อุปกรณ์ ที่เชื่อมต่อ
กับอินเทอร์เน็ต ได้อย่างเท่าเทียมกัน



Scope of UA Readiness for Programmers

1. Support All Domain Names



Accept



Validate



Process



Store



Display

2. Support All Email Addresses



Accept



Validate



Process



Store



Display

Technology Stack for UA Consideration

Applications and Websites

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

Social Media and Search Engines

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

Programming Languages and Frameworks

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

Platforms, Operating Systems and System Tools

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

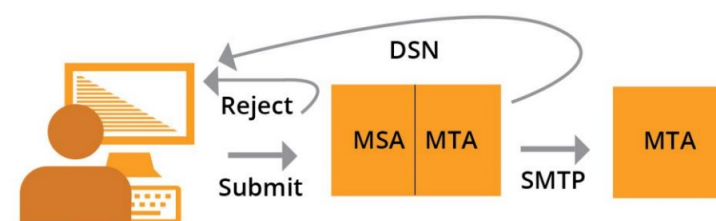
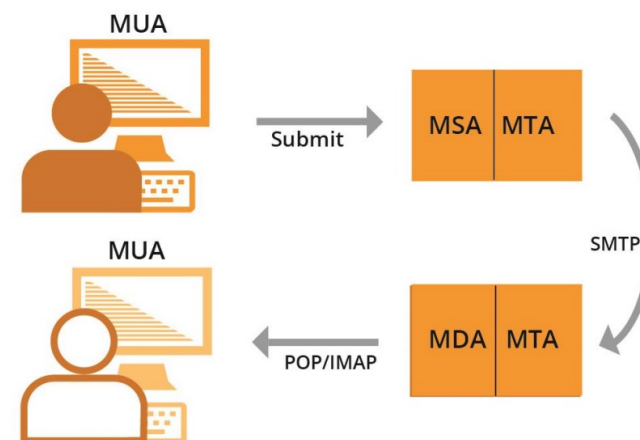
Standards and Best Practices

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

Accept, validate, process,
store and display
all domain names and
email addresses.

Email Systems and EAI Support

- ◎ All email agents must be configured to send and receive internationalized email addresses. See [EAI: A Technical Overview](#) for details.
 - **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 Transmission Agent: A server program that sends and receives mail to and from other Internet hosts. An MTA may receive mail from an MSA and/or deliver mail to an MDA.
 - **MDA** – Mail Delivery Agent: A server program that handles incoming mail and typically stores it in a mailbox or folder.



Quiz

Quiz 1 Question

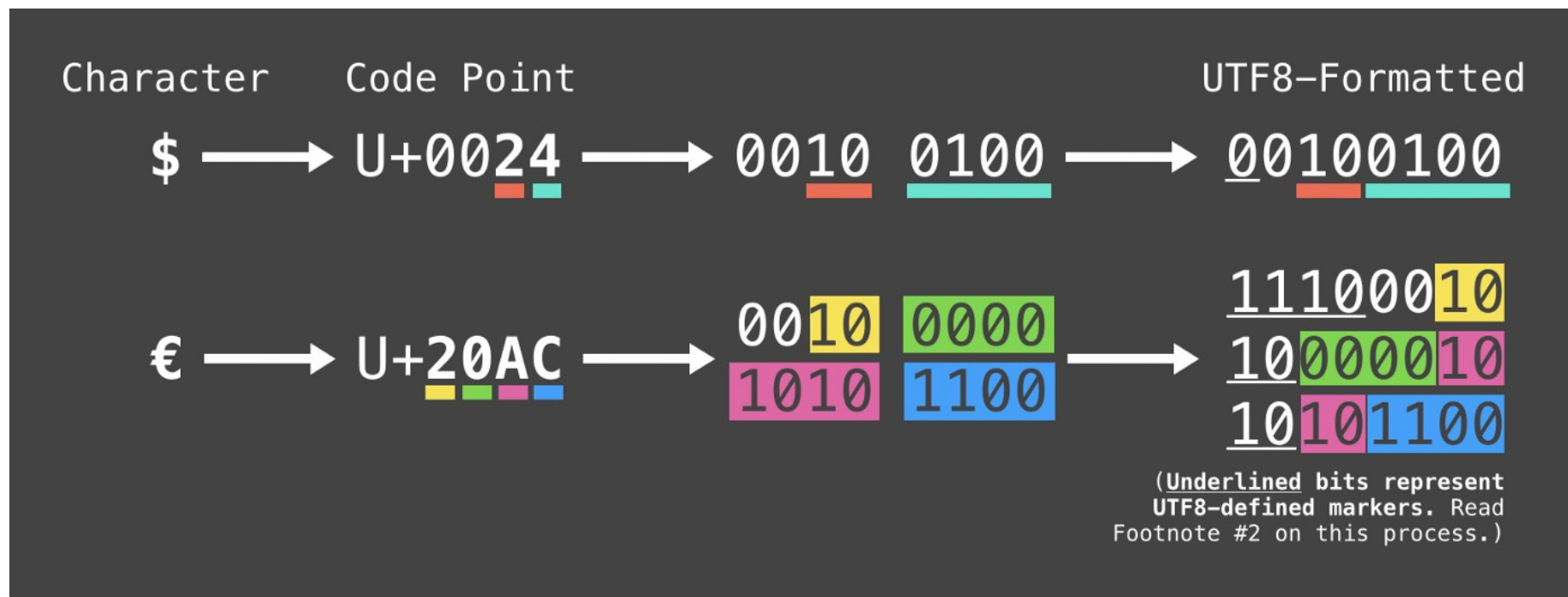
- ☉ To enhance systems to be Universal Acceptance (UA) ready, which of the following categories of **domain names** and **email addresses** are relevant?
1. ASCII domain names.
 2. Internationalized Domain Names (IDNs).
 3. Internationalized email addresses (EAI).
 4. All the above.
 5. Only 2 and 3.

Fundamentals for Internationalized Domain Names and Email Addresses

- Unicode encodes glyphs into codepoints for different scripts of the world.
 - Codepoints shown in hex using the U+XXXX notation.
 - Unicode files typically in UTF8 format, using a variable number of bytes for a codepoint.
 - ASCII is used as is in Unicode: `e` = ASCII 65 = U+0065.
- There are multiple ways to encode certain glyphs in Unicode:
 - `è` = U+00E8
 - `e + `` = `è` = U+0065 + U+0300
- Normalization ensures that the end representation is the same, even if users type differently.
 - IDN standards recommend using [Normalization Form C \(NFC\)](#).
 - Generates U+00E8 for both input versions above.



Unicode and UTF8



Domain Names and Internationalized Domain Names (IDNs)



- ◎ A domain name is an ordered set of labels or strings: [example.co.th](#)
 - The top-level domain (TLD) is the rightmost label: "th"
 - Initially, TLDs were only two or three characters long (e.g., [.th](#), [.com](#)).
 - Now TLDs can be longer strings (e.g., [.info](#), [.google](#), [.engineering](#)).
 - TLDs delegated in the [root zone](#) can change over time, so a fixed list can get outdated.
- ◎ Domain names can also be internationalized when one of the labels contains at least one non-ASCII character.
 - For example: [exâmp!e.com](#) or รู้จัก.ไทย
- ◎ Use the latest IDN standard called IDNA2008 for IDNs.
 - Do not use libraries for the outdated IDNA2003 version.

- ◉ There are two equivalent forms of IDN domain labels: **U-label** and **A-label**.
 - Human users use the IDN version called U-label (using UTF-8 format): **exâmp~~le~~**
 - Applications or systems internally use an ASCII equivalent called A-label:
 1. Take user input and normalize and check against IDNA2008 to form IDN U-label.
 2. Convert U-label to punycode (using RFC3492).
 3. Add the “xn--” prefix is added to identify the ASCII string as an IDN A-label.
 - **exâmp~~le~~** => **exmp~~le~~-xta** => **xn--exmp~~le~~-xta**
 - **普遍接受-测试** => **--f38am99bqvcd5liy1cxsg** => **xn----f38am99bqvcd5liy1cxsg**
 - **รู้จัก** => **12cn4frcvb5f** => **xn--12cn4frcvb5f**
- ◉ Email address syntax: **mailboxName@domainName**
 - EAI has the **mailboxName** in Unicode (in UTF8 format).
 - The **domainName** can be ASCII or IDN.
 - For example: **kévin@example.org** or **พจนันท์@คน.ไทย**.

หลักเกณฑ์การตั้งชื่อโดเมน .th

- 3.1. ชื่อโดเมน .th แต่ละชื่อจะต้องประกอบด้วยตัวอักษรภาษาอังกฤษ (a-z) และ/หรือ ตัวเลขอารบิก (0-9) และชื่อโดเมนต้องมีความยาวอย่างน้อย 1 ตัว แต่ไม่เกิน 63 ตัว ทั้งนี้ ชื่อโดเมนที่มีความยาว 1 ตัว จะอยู่ภายใต้การจดทะเบียนที่มีข้อกำหนดและเงื่อนไขเฉพาะ ซึ่งมูลนิธิฯ สงวนสิทธิในการพิจารณาเป็นกรณีไป
- 3.2. ชื่อโดเมนสามารถประกอบด้วยเครื่องหมายยัติภังค์ “-” คั่นระหว่างตัวอักษรหรือตัวเลข ได้ แต่ต้องไม่ใช่เรียงติดกันมากกว่า 1 ตัว นอกจากนี้ ไม่อนุญาตให้ใช้เป็นตัวอักษรแรก หรือตัวอักษรสุดท้ายของชื่อโดเมน

ชื่อโดเมน .th และ .ไทย

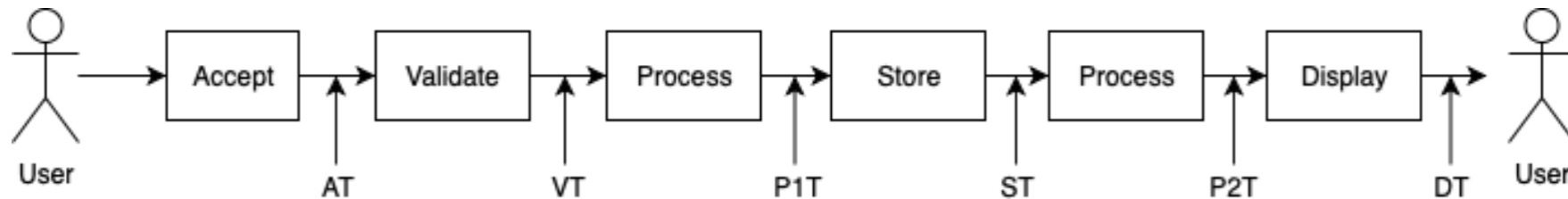
หลักเกณฑ์การตั้งชื่อโดเมน .ไทย

- 4.1. ชื่อโดเมน .ไทย แต่ละชื่อจะต้องประกอบด้วยตัวอักษรภาษาไทย พยัญชนะ(ก-ฮ อ อักษรนำ) สระ วรรณยุกต์ ยมก(ๆ) พินทุ(อ) นฤคหิต(อ) ไปยาลน้อย(ฯ) การันต์(์) และ/หรือ เลขไทย(๐-๙) โดยชื่อโดเมนจะต้องมีความยาวหลังจากแปลงเป็นพินยืไค้ด(Punycode) ไม่เกิน 63 ตัว
- 4.2. ชื่อโดเมน .ไทย ประกอบด้วยตัวเลขอารบิก(0-9) ได้ แต่จะต้องประกอบด้วยตัวอักษรภาษาไทยที่กล่าวไว้ข้างต้นอย่างน้อย 1 ตัว
- 4.3. ชื่อโดเมนประกอบด้วยเครื่องหมายยัติภังค์ “-” คั่นระหว่างตัวอักษรหรือตัวเลขได้ แต่ต้องไม่ใช่เรียงติดกันมากกว่า 1 ตัว นอกจากนี้ ไม่อนุญาตให้ใช้เป็นตัวอักษรแรกหรือตัวอักษรสุดท้ายของชื่อ

- ◉ Some applications are still verifying domain names incorrectly by using one of the outdated methods:
 - Check for a fixed length of TLD between 2-4 characters (TLD can be up to 63 characters).
 - Check from a fixed set of TLDs, e.g., using static list of strings.
 - Check for only ASCII characters.
- ◉ Some applications do not cater to additional requirements for validating EAI:
 - Check mailbox name to be a valid string in UTF-8 format.
 - DomainName can be ASCII or IDN.

UA Readiness Testing Framework

- Based on [UASG026](#), the application components can be generalized to put emphasis on the processing of internationalized identifiers.
- Each gate has its own set of requirements and processing.



- AT: Accept test
- VT: Validate test
- P1T: Process test on the input
- ST: Store test
- P2T: Process test on the output
- DT: Display test

- ◉ Validating user input, or any input, is extremely useful for various reasons, some of which include: a better user experience, increased security, and avoiding irrelevant issues.
- ◉ Validating domain names and email addresses is useful.
- ◉ Some validation methods for domain names and email addresses:
 - Basic syntax checks: is the syntax of the string correct?
 - Does the domain name contain '.' ?
 - Does the email address contain '@' and a valid domain name part?
 - Functional checks: does the domain name or email address work?
 - Is the top-level domain (TLD) in use?
 - Is the whole domain name in use?
 - Is the email in use?

Validating Domain Name

- ⦿ Validating syntax:
 - ASCII: RFC1035
 - Composed of letters, digits, and hyphen.
 - Max length is 255 octets with each label up to 63 octets.
 - IDN: IDNA2008 (RFCs 5890-5894)
 - Valid A-labels
 - Valid U-labels
- ⦿ Validating function:
 - Is the top-level domain (TLD) in use?
 - Verify against the list of TLDs.
 - Verify using a DNS request.
 - Is the whole domain name in use?
 - Verify using a DNS request.

Resolving Domain Name

- ◉ After validation, a software would then use the domain name identifier as:
 - A domain name to be resolved in the DNS.
- ◉ Therefore, to be UA compliant, the software has to use proper methods that support UA.
 - For example, passing a U-Label to the traditional functions or methods may not succeed, as it is not expecting a UTF8 domain name.

Validating Email Address

- ◉ An email address is composed of: mailboxName@domainName
- ◉ Validating syntax:
 - For domainName, see earlier discussion.
 - For mailboxName:
 - ASCII
 - UTF8 (for EAI)
- ◉ Validating function:
 - Is the domain name set up to send and receive emails?
 - Is the mailbox name able to send and receive emails?

Sending Email

- ◉ After validation, a software would then use the email identifier as:
 - An email-address based user id.
 - To send an email.
- ◉ Therefore, to be UA compliant, the software must use proper methods that support UA.
 - For example, passing an UTF8 mailbox name email address to a mail sender may not succeed, as it is not expecting a UTF8 mailbox name in the email address.

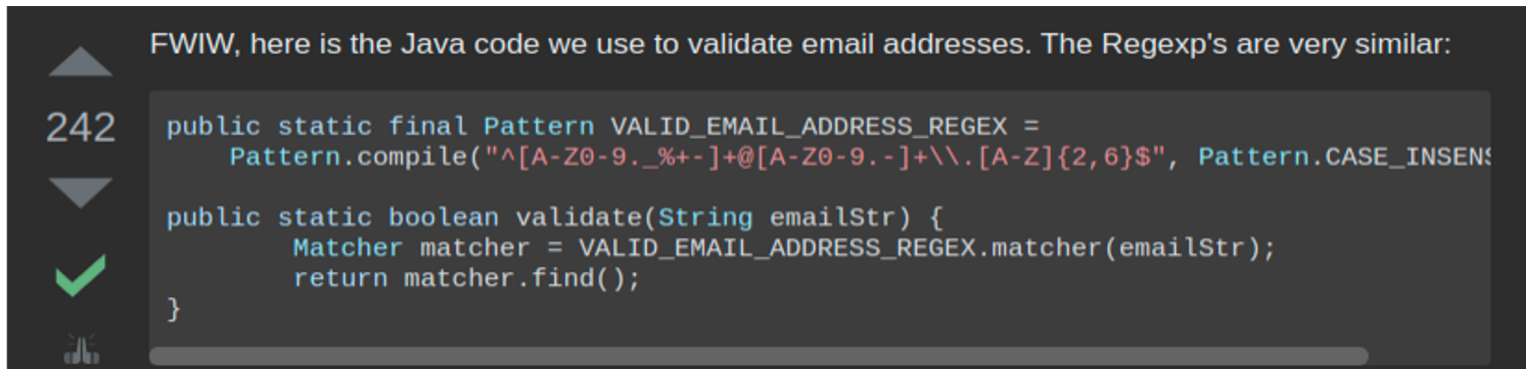
Sending Email

- ⦿ A comprehensive list of UA test cases is documented in [UASG004](#).
- ⦿ Developers are strongly encouraged to use these test cases in its unit and system testing.

Quiz

Quiz 2: A Real Example

- A company built a website where international consumers can subscribe via their email. Since the subscription form is user input, developers validated the email address before trying to send the email.
 - Developers went to *Stackoverflow* and found a **regular expression** (regex) to perform the validation:



FWIW, here is the Java code we use to validate email addresses. The Regexp's are very similar:

```
242 public static final Pattern VALID_EMAIL_ADDRESS_REGEX =  
    Pattern.compile("^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,6}$", Pattern.CASE_INSENSITIVE);  
  
public static boolean validate(String emailStr) {  
    Matcher matcher = VALID_EMAIL_ADDRESS_REGEX.matcher(emailStr);  
    return matcher.find();  
}
```

- The regex limits mailbox to letters A-Z, digits 0-9, and some symbols, the domain labels to letters, digits and hyphen, and the top-level domain to letters only with length 2-6.
- Would this regex work for the company's website? Why or why not?

- ◉ See <https://uasg.tech> for a complete list of reports.
 - Universal Acceptance Quick Guide: [UASG005](#)
 - Introduction to Universal Acceptance: [UASG007](#)
 - Quick Guide to EAI: [UASG014](#)
 - EAI – A Technical Overview: [UASG012](#)
 - EAI – Evaluation of Major Email Software and Services: [UASG021B](#)
 - Universal Acceptance Readiness Framework: [UASG026](#)
 - Considerations for Naming Internationalized Email Mailboxes: [UASG028](#)
 - Evaluation of EAI Support in Email Software and Services Report: [UASG030](#)

Programming for UA

Get Involved!

มีส่วนร่วมกับ Thailand UA Local Initiative และ UASG

- ◉ Visit Thai UA local initiative web page at รู้จัก.ไทย/ua/thailand-initiative
- ◉ Subscribe [APAC EAI Implementers' Group](#) for technical support (by THNIC)
- ◉ Access UA technical content by THNIC at: <https://wiki.thnic.or.th>
- ◉ Access all UASG documents and presentations at: <https://uasg.tech>
- ◉ Access details of ongoing work from wiki pages: <https://community.icann.org/display/TUA>
- ◉ Register to participate or listen in the UA discussion list at: <https://uasg.tech/subscribe>

Engage with THNIC

Webs site: thnic.or.th



facebook.com/THNIC.Foundation/



[twitter@thnicf](https://twitter.com/thnicf)



youtube.com/user/thnicf