

# Considerations for Naming Internationalized Email Mailboxes

2 November 2020



## Table of Contents

<b>Introduction</b>	<b>3</b>
<b>Executive Summary</b>	<b>4</b>
<b>Policy Topics for Consideration</b>	<b>5</b>
Supported Scripts	5
Length of a Mailbox Name String	5
Script Mixing	6
Preventing Invalid Strings and Unstable-rendered Strings	8
Right-to-Left (RTL) Script Consideration	9
Aliases and Display Names Consideration	11
Signs and Symbols	12
Unicode Character Normalization	12
Equivalence Considerations	13
Other Considerations	15
<b>Glossary</b>	<b>16</b>



## Introduction

This document lays out considerations for naming internationalized mailboxes and is intended for administrators of email systems. Email standards [RFC6533 Internationalized Delivery Status and Disposition Notifications](#) and [RFC6430 Email Feedback Report Type Value: not-spam](#) now allow both mailbox names (the part of an email address before the '@' sign) and domain names (the part after the '@' sign) to use almost any language or writing system; they are no longer restricted to the letters A-Z and numbers 0-9.

This is an important improvement for users who use and prefer languages other than English, and for the people with whom they exchange messages. However, system administrators need to adopt policies on mailbox names in order to maximize the benefits of having a greater choice of allowable characters, while limiting confusion, error, and opportunities for malice.

This document is intended for administrators of email systems that wish to provision internationalized mailboxes for their users and/or configure and manage systems compatible with internationalized email addresses. It will help administrators ask the right questions and make good choices when setting their policy.

In this document, familiarity with email systems and experience with Phase 1 of Email Address Internationalization (EAI) – the ability to send emails to and receive emails from internationalized addresses – is assumed. This document addresses issues in Phase 2 of EAI, such as naming and hosting internationalized mailboxes.

The Glossary section provides definitions of specialized terms and acronyms used in this report.



## Executive Summary

This document is intended for email administrators, systems administrators, and IT managers. Its purpose is to aggregate and specify the main operational and policy considerations to be considered when creating internationalized email address mailbox names, aliases, and display names.

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 for email administrators to adopt good policies for naming mailboxes. Historically, email addresses have been limited to Latin-script letters and digits, but now email technology has evolved to allow for both mailbox names and domain names to be written in almost any language and script. This flexibility makes policy choices more complex, and this document 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 actually 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 experience difficulty 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.

This 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.



## Policy Topics for Consideration

This document considers topics that might be included in a mailbox naming policy. Each section deals with one of these topics as follows:

- Description of topic.
- Explanation of considerations regarding the context of email systems, the user base, and user base needs.
- Policy recommendations (in some cases).
- Check list of actions that should be taken.

## Supported Scripts

When setting up an email system, email administrators need to decide which scripts will be the default scripts of the system.

The term “script” is used in the linguistic sense and is defined as: the conventions and sets of characters used together to write one or more languages. For instance, the Latin script is the set of letters A-Z and other characters commonly used to write English, French, and many other languages in Europe, North and South America, and Africa. The Arabic script is used to write many languages in the Middle East, North Africa, and South Asia.

### Considerations:

- Most systems will include ASCII code points for backward compatibility. EAI allows administrators to use additional scripts.

## Checklist

Research the expectations of your users and the target market to understand the writing systems needed for both the mailbox name and domain name portion of an email address.

Review this document to understand the complexities involved in providing additional scripts (e.g. possible confusion, errors, and opportunities for malice) and any potential impact on your users.

Based on the expectations of your users, make a list of the allowable mailbox name scripts.

## Length of a Mailbox Name String

Consider the length of the mailbox name in characters. What are the minimum and maximum lengths you want to permit?

### Considerations:

- There may be system constraints on the mailbox name string. Although email standards allow up to 64 octets in a mailbox name, some systems limit the string to less than the standard character limit.
- There can be user benefits to short names as they may be easier to remember and write. There can also be user benefits to long names as they can be more specific and are less likely to conflict with other mailbox names. You may already have a policy requiring mailbox names to be based on the personal name of the user, for instance, personal name and family name separated by a “.”.



- Consider keeping your existing policy but expanding it to a wider range of characters and languages. Also, consider what languages and characters your users and their correspondents are likely to be able to understand, remember, and enter into email address fields.
- When using UTF8 to add more languages and scripts, a single character can be represented in multiple octets which will increase the mailbox character length. For example, a mailbox name that consists of ASCII letters that requires 20 octets can be expanded to 60 octets when representing 20 letters in another language or script.

**Recommendations:** None.

**Checklist:**

[ ] Know the constraints of your system.

[ ] Understand the expectations of your users related to internationalized mailbox names.

[ ] Consider applying the same or a similar policy on length to internationalized mailbox names as to legacy mailbox names.

**Script Mixing**

Script mixing refers to the use of characters from multiple scripts in a single mailbox name in a way that is not used in normal writing. In particular, some characters from other scripts might be confusingly similar to characters from the primary script, which opens possibilities for malice or confusion.

**Considerations:**

- Script mixing leads to security issues and user confusion. For example, the below image shows several different but similar-looking Unicode code points that might lead to user confusion<sup>1</sup>.



---

<sup>1</sup> Source : <https://thenextweb.com/wp-content/blogs.dir/1/files/2014/08/confusables-3.png>



- Your correspondents will be writing the mailbox name and domain name together as an email address. If the domain name is in a certain script, it is natural for the mailbox name to be in the same script. Consider if any other script might be used with the domain name's script.
- Some combinations might be necessary, while others can be confusing or perhaps malicious.
- Some cultures use multiple scripts for ordinary writing. For instance, most of the Japanese text is a mixture of characters from the Kanji, Hiragana, Katakana, and Latin ("Roman") scripts. However, it is unusual to see Arabic script in Japanese texts.
- Latin-script mailbox names are typically case-insensitive. Upper-case and lower-case letters are treated as equivalent. For the scripts you use, consider if there are other equivalencies your users expect. For clarity, we do not generally consider mixtures of upper and lower-cases to be script mixing; see the [Equivalence Considerations section for more detail](#).
- Within the Domain Name System (DNS), Internationalized Domain Names (IDNs) can be represented by two formats: U-label and A-label. In this document, A-label usage is not considered to be script mixing; see the [Aliases Consideration section for more detail](#).

### Recommendations:

- For the best usability, it's recommended that the mailbox name use the same script as the domain name portion of the email address, unless otherwise required to achieve your organization's goals.
- Avoid the mixing of languages using the same script in the mailbox name to avoid using or creating visually similar mailbox name characters that are using different code points.
- Only characters that are used together in the ordinary writing of a language should be allowed in a given mailbox name in that language. Within the mailbox names, script mixing should be prohibited. For example, an Arabic mailbox might be limited to just Arabic characters, while a Japanese mailbox name might be permitted to include Kanji, Hiragana, Katakana, and Latin characters.

Sample 1: Arabic script email address: `مستخدم@رسيل.السعودية`

Sample 2: Japanese script email address with script mixing:

`誰か@日本語ドメイン名試験.jp`

### Checklist:

[ ] Consider the possible confusion, errors, and opportunities for malice due to script mixing both within the mailbox name and between the mailbox name and the domain name.

[ ] Allow limited script mixing where there is a clear user need based on local practice and market requirements. For example, users of a retail email service may demand names with certain kinds of script mixing, while members of an organization may have fewer objections to a limited choice of mailbox names that exclude that script mixing.

[ ] If users are allowed to request their mailbox names, provide examples for users.



## Preventing Invalid Strings and Unstable-Rendered Strings

In some writing systems, characters can be combined in valid ways allowing productive communication, but also in invalid ways leading to confusion and security risks. Such strings of characters will not be rendered consistently between platforms and applications – they will look different on different systems or displays. This topic addresses ways to avoid invalid strings.

### Considerations:

- The DNS has defined clear rules for which characters can be used and which should be avoided for both languages and scripts. For some scripts, particularly Southeast Asian and South Asian scripts, characters and marks can be combined on multiple levels.
- For example, consider the Thai script strings “ดี” vs “ดี̄”. The first one, “ดี” U+0E14(ด) + U+0E35 ( ี) + U+0E35 ( ี) is invalid because it looks like ดี U+0E14(ด) + U+0E35 ( ี). The Thai script user community has identified a rule that U+0E35 cannot follow U+0E35.
- These rules used in the DNS are known as Label Generation Rules (LGRs). You might want to adopt these rules for the mailboxes on your system.
- However, the LGRs might be more restrictive than necessary to be on your system. For instance, the Root Zone LGRs forbid using digits, but digits may be fine to use in your mailbox names. Some scripts do not yet have LGRs, so you cannot use them as a reference.
- You can refer to the existing [Second Level Reference LGRs](#) to validate the mailbox names, and an LGR validation tool is available online at <https://lgrtool.icann.org/>. The source code for the tool is also available on GitHub which you can download and integrate into your system.

### Recommendations:

- Avoid potential security issues caused by creating mailbox names using invalid strings and its equivalents.
- If you intend to offer mailbox names in one language, refer to the language-based LGR. For example, if you want to offer mailbox names in German then use the German language LGR.
- If you intend to offer mailbox names in one script which may cover multiple languages, refer to the script-based LGR. For example, if you want to offer mailbox names in German, French, and Swedish then use the appropriate subset of the Latin script LGR.
- For scripts not covered by the [Second Level Reference LGRs](#), you can refer to the Root Zone LGRs or follow high-level principles in [RFC6912](#). For rules on digits in names using right-to-left scripts, check [the Right-to-Left Scripts for Internationalized Domain Names for Applications \(IDNA\) RFC5893](#).

### Checklist:

[ ] Determine if the Second Level Reference LGR meets your desired mailbox name string requirements. If it does, adopt the reference LGR. If it does not exactly meet your requirements, update it as needed.

[ ] Share the Label Generation Rules (LGRs) as a mechanism to allow your end users to understand which characters and combinations will be considered valid.



[ ] If applicable, use the LGR validation tool to manually validate the mailbox strings.

[ ] Consider if you want to use the LGR validation tool as-is, or if you want to deploy it on an internal server to streamline your workflow and to keep the mailbox names it processes inside your network.

### Right-to-Left (RTL) Script Consideration

Arabic, Hebrew, Persian, and Urdu are current widespread right-to-left writing systems. Over time, as the usage of the Arabic script spread, the repertoire of 28 characters used to write the Arabic language was supplemented to accommodate the sounds of many other languages such as Pashto, Persian, Urdu, etc. While the Hebrew alphabet is used to write the Hebrew language, it is also used to write other languages such as Yiddish.

In a right-to-left (RTL), top-to-bottom script, writing starts from the right of the page and continues to the left. This can be contrasted against left-to-right (LTR) writing systems, where writing starts from the left of the page and continues to the right.

Note, however, that within a right-to-left script, some text may be written left-to-right. For instance, in Arabic, numbers and words in Latin script are written left-to-right. Thus, these scripts are sometimes referred to as "bidirectional".

#### Considerations:

- Right-to-left scripts have potential for confusion that left-to-right scripts do not. The Label Generation Rules (LGRs) for these scripts reflect careful consideration of these possibilities. Take what the LGRs require or forbid seriously as there may be good but not obvious reasons for it.
- When exchanging email between a right-to-left address and people who do not read right-to-left script, it is even harder to be sure the address is understood. The use of alias names may be especially helpful. See the section, [Aliases and Display Names Consideration for more.](#)
- Some right-to-left scripts (e.g., Arabic and Hebrew) can have the "same" labels with more than one representation. These alternatives lend themselves to being treated as equivalents to an email address, rather than independent addresses. See the section, [Equivalence Considerations for more details.](#)
- It is common for text in a right-to-left context to be bidirectional. For example, Latin-script words (e.g., "Cola") and Arabic numerals (e.g., 123) will be written left-to-right. Also, some punctuation takes on the directionality of surrounding text. This means that mailbox names that include left-to-right components may be more likely to be mistyped.
- Arabic script uses its own numerals in addition to ASCII numerals. If a mailbox name includes a number, consider the type of number to use and whether the name using the other kind of number should be treated as an "equivalent". Script mixing concerns apply for names that have both kinds of digits.
- With bidirectional text, it can be visually unclear what the order of characters is. Usually, text is entered and stored in the same order that it is spoken, but this may not have a simple correspondence to the right-to-left position of the text in the displayed address. This is generally not a problem for someone who can read the script and understand when the direction changes. However, those who cannot read the script may have difficulty determining keypress order from the address as displayed.



- Local keyboards are adapted to the local script and may not conveniently include all characters used in email addresses. Some keyboards do not include keys for period ('.') or at-sign ('@'). This makes email addresses more difficult to type. Consider the type of keyboard that will be used to type in your mailbox names, and whether to allow characters that are harder to type.
- Many right-to-left scripts include diacritics for representing vowels and pronunciation. These are essentially optional. If a mailbox name allows these diacritics, users may forget to include them. It is simpler to allow names without these diacritics.
- Arabic script, which usually displays characters joined together in calligraphy, has optional characters to force or prevent joining between characters. These are known as "zero width joiner" and "zero width non-joiner". They are a lesser-known mechanism – an artifact of computer-encoded text – so many users may not know how to use them. Mailbox names that rely on joiners may be more difficult for users to type correctly.

### Recommendations:

- It's not recommended to use combining marks especially in Arabic script. For example, use محمد instead of مُحَمَّد and شؤون instead of شُؤُون .
- Mailbox names that use right-to-left scripts should not begin with a digit. Digits at the start of a text may be placed confusingly by the text direction rules.
- Avoid potential security issues caused by invalid strings and their "equivalents". Invalid strings and equivalents are particularly common for Arabic script. See the [Preventing Invalid Strings and Unstable-Rendered Strings section](#) above and the [Equivalence Considerations section](#) below.
- If you intend to offer mailbox names in one language, refer to the language-based LGR. For example, if you want to offer mailbox names in Arabic then use the Arabic language LGRs. Check the [Equivalence Considerations section](#) for details.
- If you intend to offer mailbox names in one script that may cover multiple languages, refer to the script-based LGR. For example, if you want to offer mailbox names in Arabic, Pashto, and Urdu, then use [the Arabic script LGR](#).
- For the scripts not covered by the [Second Level Reference LGRs](#), refer to the Root Zone LGRs or follow high-level principles in [RFC6912](#).
- For rules on digits in mailbox names using right-to-left scripts, check [RFC5893](#) which provides the needed guidance on the use of right-to-left scripts for Internationalized Domain Names for Applications (IDNA).

### Checklist:

[ ] Avoid script mixing with right-to-left scripts as it can lead to confusability and security issues.

[ ] Use Second Level Reference LRGs or Root Zone LGRs.

[ ] Avoid allowing two mailboxes with mailbox names that are equivalent to each other.



## Aliases and Display Names Consideration

Some email systems make it easy for users to have multiple email mailbox names or multiple display names directing mail to and from the same mailbox. These aliases and display names serve many purposes and can be used for several EAI use cases.

Use cases include:

- Users requiring multiple script email addresses, such as a bilingual business card in both the native script and Latin script.
- Working around incompatible systems when sending or receiving emails.
- Making it easier for senders to recognize the email address and aliases in their address book.

### Considerations:

- Assess the email delivery paths from your users to their correspondents and from their correspondents back to your users. Be aware that mail delivered in these two directions might encounter different systems, so one direction might work well and the other might have problems.
- The more varied the correspondents and the wider variety of delivery paths for emails to and from your system, the more likely you are to encounter a delivery system that fails to deliver mail to or from internationalized email addresses.
- Allowing aliases or display names that are in a different script than the mailbox name might create security issues; however, it could also be beneficial when the EAI address is used across different language communities.

Below is an example of Arabic display name for an English email address:

"مشرف@اختبار-سجل.مصر"	< <a href="mailto:adminmail@datamail.asia">adminmail@datamail.asia</a> >
Display name	Email address

### Recommendations:

- Consider providing an ASCII email address as an alias for your internationalized email addresses. The alias should deliver emails to the same location as the internationalized address. The user should be able to choose whether to send a message or a reply from the internationalized address or from the ASCII alias address.

### Checklist:

[ ] Understand if your system supports aliases.

[ ] If your system supports aliases, add an alias-creation option to the user interface during the mailbox name selection process. The user will be informed that they should have an ASCII alias and be allowed to create one associated with their EAI mailbox name.

[ ] You may optionally add a user interface allowing the user to add additional aliases at a later time, if your system supports this.

[ ] If you assign mailbox names to users rather than allowing them to select mailboxes themselves, then assign the ASCII alias at the same time as you assign the EAI mailbox name.



## Signs and Symbols

Some signs and symbols are allowed in ASCII mailbox names, for example, (.) dot, underscore (\_), hyphen (-), and plus sign (+). When deciding or selecting an EAI mailbox name(s) the administrator also needs to consider what Unicode signs or symbols should be allowed for use in the scripts being supported.

### Considerations:

- The inclusion principle (refer to [LGR procedure](#)) should be considered. The inclusion principle advises to only include the set of characters, e.g., signs and symbols, that are needed by the users, starting from an empty list. Symbols that create security issues specific to your implementation should be avoided.
- Only allow signs and symbols that are feasible for your users and their correspondents to read and input. The mailbox name should be typeable by commonly used Input Method Editors (IMEs).
- Set a policy that relates and have similar limitations considered for the Latin mailbox name as the internationalized mailbox name in a consistent and understandable way.

### Recommendations:

- Avoid mathematical marks that could create security issues, e.g., # \$ % = " \* < , > .
- Emoji should not be allowed (reference to ICANN's SSAC Advisory [SAC95](#)).
- The combining marks that fail the type-able principle should be disallowed, e.g., Arabic Hamza Above (U+0654).

### Checklist:

By default, avoid using mathematical marks.

By default, avoid using any signs and symbols that do not exist in your user's keyboard/input devices. Some signs and symbols are not available in input devices for specific markets.

Understand your users and local market expectations. If signs and symbols are required for your market, the dot (.), underscore (\_), hyphen (-) and plus sign (+) are commonly used and could be added to the repertoire.

Review the use of additional signs (if needed) and ensure it does not cause a security issue for the end users, including customer confusion, cross-scripting, and other security issues.

## Unicode Character Normalization

The Unicode standard defines a text normalization procedure that replaces equivalent sequences of characters so that any two texts that are equivalent will be reduced to the same sequence of code points.

Unicode defines normalized forms of a characters that can be encoded in multiple ways as there are various types of standardized forms, e.g., Normalization Form D (NFD) and Normalization Form C (NFC).



Example: Å can be encoded by [212B](#) or [00C5](#). Both should refer to the same mailbox, not to different mailboxes, and would be normalized to the same NFD and NFC.

Note that although these examples use Latin characters, similar examples are found in other scripts.

More details on the other types of normalization, with samples, can be found on the [Unicode consortium website](#).

## Recommendations

- Mailbox names should use NFC for consistency with domain names, and also because NFC strings, in most cases, are shorter than NFD strings.
- Typically, standard software already encodes Unicode normalization in their libraries. Email administrators should know the normalization type of the system they operate in order to be able to assign and manage its mailboxes.
- Domain names use Normalization Form C (NFC). When building a system, ensure that the libraries support NFC for consistency.

## Checklist

- Get to know the normalization type of your email system.
- Ensure that your email program does normalized-form-independent name comparisons.
- If it's possible to select normalization type, select NFC unless there is motivation to use other forms.

## Equivalence Considerations

When domain name labels are considered the “same” or “interchangeable” by the community, they are called “variant” domain name labels. When email mailbox strings are considered the “same” by the community, this document refers to them as “equivalent” mailbox names.

Sometimes equivalent names are useful and can be used as aliases for the primary mailbox name. In other cases, they are confusing and may lead to security issues if not identified or assigned to a different user.

## Considerations:

- In ASCII email addresses, some mail services may consider upper-case and lower-case letters as equivalent.
- The DNS community has done a great deal of work to define clear Label Generation Rules (LGRs), including identification of variant labels in different scripts. Consult the LGRs to see if you want to use them as the basis of your definition of mailbox name equivalent.

Examples:

1. Traditional Chinese and Simplified Chinese equivalents:  
名称 which means “name” (U+540D U+79F0)



名稱 which means “name” (U+540D U+7A31)

名称 (Míngchēng, means “Name”) is used in China mainland and 名稱 (Míngchēng, means “Name”) is used in Taiwan and Hongkong, however, all of the Chinese community could recognize these two words as the same.

2. Arabic script equivalent (the word means “network” in Arabic):

شبكة (Unicode U+06C3 U+06A9 U+0628 U+0634)

شبكة (Unicode U+0629 U+06A9 U+0628 U+0634)

شبكة (Unicode U+0629 U+0643 U+0628 U+0634)

3. Full width versus half-width characters in Japanese and Chinese scripts. “J I M” vs “JIM”, “1 2 3” vs “123”, “アイオ” vs “ア伊”.

4. Case folding equivalent:

John.Callen  
john.callen  
John.callen  
john.Callen

5. Variation of name and separator (may also be considered equivalent):

john.callen  
johncallen

- For an email system, you might consider delivering any emails addressed to the equivalent mailbox names to the same mailbox. This avoids customer confusion, cross-scripting, and other security issues.
- Additional equivalent names can be assigned to the same mailbox as aliases, but these can get very numerous (hundreds or thousands) and become a management challenge. So, make the trade-off between the usefulness and the overhead of equivalents with caution.
- Note that variants in the domain name portion of the email address may have similar user impact. The discussion of domain names and their variants is outside of the scope of this document
- Be aware of the kinds of errors or confusion that may occur when users type in email addresses for your system. This document covers confusion in the mailbox name part of the address.
- There may also be confusion about the correct spelling or normalization of the domain name part of the address. Domain name variants are outside of the scope of this document.

### Recommendations:

- After a mailbox name has been assigned, the equivalent of that mailbox name should be identified and marked as registered and withheld from registration to a different user.
- Consider treating the most useful equivalents of a mailbox name as aliases (see [Aliases Consideration section](#) of this document), and deliver emails addressed to the equivalent mailbox to the same location as the original mailbox name. However, its



management might be difficult, so make the trade-off between its usefulness and the overhead of handling equivalents with caution, and on an as-needed basis.

- Consider practical, language-based equivalent rules for an equivalent mailbox. For example, with ASCII mailbox names, treating all upper and lower-case variations as equivalent is practical. With a Simplified Chinese mailbox name like 中國 which means “China”, consider treating the Traditional Chinese name with the same meaning, 中國, as an equivalent.
- All [case folding](#) permutations should be considered as the same mailbox.
- When there is more than one digit set being used in the same repertoire (set of possible codepoints used to create mailbox names), those digits should be mapped as equivalent codepoints. For example, a Hindi mailbox administrator may consider abc123 and abc१२३ as equivalent mailbox names:

Mailbox name: abc123

Equivalent mailbox name: abc१२३

### Checklist:

[ ] Define a policy for determining the “same” or equivalent mailbox names that will apply to your system based on the writing system, user expectations, and technical capabilities of your implementation. Your policy will need to examine the LGR, case-folding, separators, numerals, and symbols in the considerations section above.

[ ] Determine if the [Second Level Reference LGR](#) meets your desired mailbox name string equivalence requirements. If it does not meet your requirements, adapt it as needed.

[ ] Share your policy to let end users understand which characters and combinations will be considered valid and which ones might have equivalence.

[ ] When using equivalent strings, consider if you want to use the validation tool as is or if you want to deploy it on an internal server to streamline your workflow and to keep the mailbox names it processes inside your network.

### Other Considerations

The internationalized domain name part of an email address can be spelled with Unicode characters or with an equivalent ASCII-only name beginning with “xn--”.

### Considerations:

- The “xn--” spelling of a domain name is an underlying technical detail. Ideally, the user never needs to see it. However, in a transition time where not every email system supports internationalized email addresses, the “xn--” spelling might be a useful workaround.

### Recommendations:

- Spell domain names with their internationalized non-ASCII names. Avoid displaying the “xn--” alternative name.
- Some email clients might not automatically link the U-label and A-label forms of email address mailbox names, so ensure that both labels are mapped to each other.



## Glossary

*Alias*: An email address that delivers mail to the same place as another email address. Typically, an alias is an ASCII email address that provides a fallback address for an EAI address in case a correspondent cannot send to the EAI address due to delivery systems not supporting EAI.

*ASCII*: Refers to a limited character set made up of Latin-script letters without accents, numeric digits, and a few punctuation marks. The term ASCII is a computer industry acronym.

*Display Name*: The name that appears next to the email address.

*Domain name*: the sequence of names used to identify a website or email server, e.g. "uasg.tech" or "icann.org".

*EAI*: Email Address Internationalization.

*Email Address Internationalization*: The practice of email addresses that are made up of all practical languages and scripts, not just ASCII or Latin script.

*Equivalent*: When two or more Unicode code points share one meaning or are linguistically deemed equivalent by local authorities, resulting in "character variants". For example, they occur between Simplified and Traditional Chinese scripts, and are potentially confusing to people who are using these scripts.

*IDN*: Internationalized Domain Name.

*IME (input-method editor)*: IME is an application that allows a standard keyboard (such as a US-101 keyboard) to be used to type characters and symbols that are not directly represented on the keyboard itself.

*Internationalized Domain Name*: a domain name with non-ASCII characters.

*Label Generation Rules*: rules for the use of characters in domain names.

*LGRs*: Label Generation Rules.

*Mailbox Name*: The portion of an email address identifying the mailbox location under a domain name. Used interchangeably with other terminology such as "mailbox name", "account name", "username", "UTF-8 part", "local-part".

*Punycode*: an algorithm to encode Unicode in the limited ASCII character subset used for Internet hostnames.

*Phase 1 of EAI*: an email system that can successfully send emails to and receive emails from, internationalized email addresses, but cannot host mailboxes with internationalized names.

*Phase 2 of EAI*: an email system that can successfully host internationalized mailbox names and successfully send emails to and receive emails from, internationalized email addresses.

*Rendering*: the process by which a computer display system turns a sequence of characters



into a visual appearance on the screen.

*Script*: the conventions and sets of characters used together to write one or more languages.

*Script.txt*: The data files associated with the Unicode Character Database, identifying Script property values of all code points (<http://www.unicode.org/Public/12.1.0/ucd/Scripts.txt>).

*Unicode*: a character encoding standard, encompassing most of the world's scripts and written languages. See <http://unicode.org>.

*Variant names*: names that are considered the “same” by the community.