Draft –
Configuring for Email Address Internationalization (EAI) Tutorial

April 2020
Agenda

- Tutorial Target Audience and Objectives
- Key Fundamental Aspects: Unicode, IDNs, UA
- Key Fundamental Aspects: Email
- Email Address Internationalization (EAI)
- Demonstration: Setup
- Demonstration: Action!
- Considerations Deploying EAI: Case Mapping and Matching
- Considerations Deploying EAI: Spam
- Considerations Deploying EAI: MX Consistency, Mailing Lists, Delivery Notifications, and User Accounts
Tutorial Target Audience and Objectives

- Target Audience: System Administrators of Email Servers

- Objectives:
  - Understand base key concepts related to internationalized email.
  - Understand Email Address Internationalization (EAI)
  - Configure Postfix and Courier to support EAI.
  - Understand issues regarding the deployment of EAI.
Key Fundamental Aspects: Unicode, IDNs, UA
Encoding glyphs into codepoints.

In specifications, codepoints are shown in hex using the U+XXXX notation.

Codepoints are typically carried using the UTF-8 (Unicode Transformation Format, 8 bit) format.
- Variable number of bytes for a single codepoint.
- ASCII is used as is.
- Gold standard for carrying Unicode codepoints in web, protocols, etc.
There are multiple ways to use a glyph:

- “è” = U+00E8
- “e`” = “è” = U+0065 U+0300
- Normalization is a process to insure that no matter the user type, the end representation will be the same. For the two entries above, Normalization Form C (NFC) will generate U+00E8 for both.
Domain Names

- A domain name is an ordered set of labels: a.b.c.d
- A top-level domain is the rightmost label: “d” in left-to-right scripts.
- The Domain Name System (DNS) is the distributed database and service for querying domain name records.
Domain Names

- A domain name may have multiple DNS records such as:
  - IPv4 address for that domain name.
  - IPv6 address for that domain name.
  - Hostname of the email server responsible for that domain name.
  - ...

- A zone is the list of domain name records – called Resource Records (RR) – for the labels under another label (a bit simplified…)
Internationalized Domain Names (IDNs) enable the use of non-ASCII characters for any label of a domain name.

- Not all labels of a domain name may be internationalized.

Example: exâmple.ca

User uses the IDN version, but the IDN is converted into ASCII for DNS resolution.

- exâmple => exmple-xta => xn--exmple-xta
- The xn-- prefix is added to identify an IDN.
Internationalized Domain Names (IDNs)

Example process of using an IDN:

- User enters in a browser: http://exâmple.ca
- Browser does normalization on the user entry.
- Browser converts exâmple.ca in an ASCII compatible representation called Punycode [RFC3492], and adds ‘xn--’ in front of it.
  - exâmple.ca becomes: xn--exmple-xta.ca
- Browser calls the DNS to get the IP address of xn--exmple-xta.ca
- Browser connects to the HTTP server at the received IP address.
Internationalized Domain Names (IDNs)

- The protocol for handling IDNs is named IDN for Applications (IDNA).
  - Two versions: IDNA2003 and IDNA2008. The latter (IDNA2008) is the one currently used.

- U-label is the Unicode native representation of an IDN label: exâmple

- A-label is the Punycode representation of an IDN label: xn--exmple-xta
Universal Acceptance (UA)

- UA is about how to appropriately support internationalized identifiers, as well as new and long TLDs.
  - Internationalized identifiers:
    - IDN
    - EAI
Universal Acceptance (UA)

- UA is also about:
  - New or longer string TLDs (new generic top-level domains – new gTLDs)
  - In the early days, TLDs were either two or three characters long (.ca, .com). Recently, TLDs started to have longer strings (.info, .google).
  - A TLD label may have up to 63 octets.
  - Some applications are still verifying that the TLD entered by a user
    - is from the list of earlier TLDs, or
    - has a maximum of 3 characters.
Universal Acceptance (UA)

- UA is also about:
  - Added/removed TLDs:
    - TLDs come and go on a « daily » basis.
    - Some applications are verifying the correctness of a TLD based on a static list which is not the latest one, thus making wrong assumptions about the existence of a TLD.
Key Fundamental Aspects: Email
Email Terminology

- Terminology:
  - Mail User Agent (MUA):
    - The software used by the user who sends and receives email.
    - With web mail, the MUA is an application run in a browser environment.
  - Mail Transfer Agent (MTA)
    - A software, usually on servers, that transfers mail on behalf of the user to another MTA.
  - Mail Submission Agent (MSA)
    - A software, usually on servers, which receives the email from the MUA. Typically, this function is bundled with an MTA.
Email Terminology

》 Terminology:

》 Mail Delivery Agent (MDA):
   - A software, usually on servers, which receives the email from an MTA and is the final destination for the email.
   - It typically stores the email in a file (or a database) and waits for the MUA of the destination user to fetch the email. Typically, this function is bundled with an MTA.
Email Terminology

- For example, Postfix is typically used as an MTA, MDA, and MSA.
- For more details, see UASG012 report.
- For simplicity, MSA and MDA are not shown in the next slides.

Sending an email
Email: How To Find the Destination Server

- When sending email to user@example.com, the method to find the destination email server is by querying the DNS for the MX records of the domain.

- For example, the MX records for example.com could be:
  - MX 10 server1.example.com
  - MX 10 server2.example.com
  - MX 20 server3.example.com
The sender email server would then try connecting to either server1 or server2 since they have same priority (10). If none respond, it would then try server3 since it has a lower priority (20).

- The higher number means lower priority.
Email Delivery Path

Mix is also very common: Email software for one user, web email for other user.

Mail server is the MTA; the source and destination servers are MSA and MDA, respectively.

Mail User Client can be on desktop, laptop, or mobile.
Email Delivery Path Considerations

- Each user of an email communication chooses his own email environment/software/setup independently.

- The sender does not know the receiver email environment, meaning:
  - The sender does not know which protocols are used to deliver email.
  - The sender does not know if the receiver email supports some features.
Email Delivery Path Considerations

- The delivery goes through a chain of email servers.
  - The number of email servers is unknown.
  - The actual chain of servers:
    - Is unknown at the beginning.
    - May change for any subsequent email sent.
  - The features supported by each email server is unknown to the path or from the sender.
  - Features are only discovered one hop at a time (i.e. the next hop).
Email Address
Internationalization (EAI)
Email Address Internationalization (EAI)

- Email syntax: leftside@domainname
- Domain name can be internationalized as an IDN (U-labels or A-labels).
- Left side (also known as local part/mailbox name) with Unicode (UTF-8) is EAI.

- Side effect:
  - Mail headers need to be updated to support EAI.
  - Mail headers are used by mail software to get more information on how to deliver email.
Email Address Internationalization (EAI)

- Not all email servers support EAI, so a negotiation protocol is used to only send EAI when the target server supports it. If not, then it falls back and returns an ‘unable to deliver’ message back to the sender.

- The SMTPUTF8 option is used within the mail transfer protocol (SMTP: Simple Mail Transport Protocol) to signal the support of EAI by an email server.
EAI Protocol Changes

- **SMTP**
  - Is augmented to support EAI.
  - Has a signaling flag (SMTPUTF8) to specify support of EAI.
  - All SMTP servers in the path must support EAI to successfully deliver the email.

- **POP/IMAP**
  - Are augmented to properly support EAI.
  - Have a signaling flag to specify support of EAI.
  - Could “half support” EAI by providing a downgraded email version to the non-EAI conforming email software clients (more details on downgrading later).
EAI Protocol Changes: SMTP

- SMTP Server announcing the support of EAI on the initial greeting.
  - EHLO SMTPUTF8

- SMTP Client connecting to the compliant SMTP Server.
  - MAIL SMTPUTF8

- Headers may have UTF-8 content.

- Email body already supports UTF-8.
SMTPUTF8 Example

Server S forwarding an email to server R

S: <connect>
R: 220 receive.net ESMTP
S: EHLO sender.org
R: 250-8BITMIME
R: 250-**SMTPUTF8**
R: 250 PIPELINING
S: MAIL FROM: <猫王@普遍接受-测试.世界> **SMTPUTF8**
R: 250 Sender accepted
S: RCPT TO: <ray@receive.net>
R: 250 Recipient accepted

Specific SMTPUTF8 Signaling (EAI support)
SMTPUTF8 Example

S:DATA
R:354 Send your message
S:From: 猫王 <猫王@普遍接受-测试.世界>
S:To: ray@receive.net
S:Subject: 我们要吃午饭吗？
S:
S:How about lunch at 12:30?
S:
R:250 Message accepted 389dck343fg34
S:QUIT
R:221 Sayonara

Note: example from UASG-012 but corrected.
EAI IMAP/POP Protocol Changes

- **POP:**
  - UTF8 command

- **IMAP**
  - ENABLE UTF8=ACCEPT command
IMAP Example

S: * OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT QUOTA AUTH=PLAIN IDLE ACL ACL2=UNION STARTTLS ENABLE UTF8=ACCEPT]


C: a1 login "kévin" $s#WQw4M-a9%fd124
S: a1 OK LOGIN Ok.

S: * OK
C: a2 ENABLE UTF8=ACCEPT
S: * ENABLED UTF8=ACCEPT
S: a2 OK Options enabled

N.B. The username used for IMAP login does not necessarily need to be UTF8 nor be the actual mailbox name.
IMAP Example

C: a3 select "INBOX"
S: * FLAGS (Draft Answered Flagged Deleted Seen Recent)
S: * OK [PERMANENTFLAGS (Draft Answered Flagged Deleted Seen)] Limited
S: * 5 EXISTS
S: * 0 RECENT
S: * OK [UIDVALIDITY 571338720] Ok
S: * OK [MYRIGHTS "acdilrsw"] ACL
S: a3 OK [READ-WRITE] Ok
IMAP Example

C: a4 fetch 1 body[header]
S: * 1 FETCH (BODY[HEADER] {746}
S: Delivered-To: kévin@xn--exmple-xta.com
S: Return-Path: <user@example.com>
S: Received: from postfix.xn--exmple-xta.com (ip47.ip-144-217-171.net [::ffff:144.217.171.47]) by courier with UTF8ESMTP id 000000000010008C.000000005DA8AB35.0000503F; Thu, 17 Oct 2019 17:56:05 +0000
S: Received: from exâmple.com (localhost [IPv6:::1]) by postfix.xn--exmple-xta.com (Postfix) with UTF8SMTP id CD2B4100377 for <kévin@exâmple.com>; Thu, 17 Oct 2019 17:55:58 +0000 (UTC)
IMAP Example

S: From: user@example.com
S: Reply-to: user@example.com
S: Subject: UTF8 email
S: Message-Id: <20191017175558.CD2B4100377@postfix.xn--example-xta.com>
S: Date: Thu, 17 Oct 2019 17:55:58 +0000 (UTC)
S: To: kévin@example.com
S: ...
S: a4 OK FETCH completed.
C: a5 LOGOUT
S: * BYE Courier-IMAP server shutting down
S: a5 OK LOGOUT completed
RFC6855 (IMAP UTF-8) says that user login is done by AUTHENTICATE command instead of LOGIN.

Some implementations updated LOGIN to support UTF-8, so LOGIN can also be used for UTF-8 in some implementations.
To send and receive an email with EAI:
- All email parties involved in the delivery path have to be updated for EAI support.
- If a single SMTP server in the path does not support EAI, then the email is not delivered.
What happens when one email (SMTP) server in the path does not support EAI?

• The last server trying to send to the next hop:
  • Sends back to the sender user a report of unable to deliver.
  • Drops the email.

• Similar to reports that a sender receives when an email address does not exist.
What happens when the receiver client software (IMAP/POP) does not support EAI?

- The IMAP/POP server can be “nice”:
  - By providing a downgraded version of the email.
    - Changing the EAI to some non-EAI version of the local part.
- If IMAP/POP server can not be “nice”, then should send a report back to the sender…
  - But that is not always possible as the “mail server” may just be an IMAP/POP server, not SMTP.
Protocol Changes, Delivery Path Considerations

- IMAP/POP “downgrading”
- RFC6857
- UTF8 values are encoded as MIME: “=?UTF-8?Q?...?= ;; form
- While it may look interesting, downgrading may cause many issues for the users and the sysadmin to debug issues. Try to avoid using the downgrading mechanism if you can.
Demonstration: Setup
○ EAI email path between two (2) users.
Demonstration: Setup

- User 1:
  - kévin@example.com
  - Using MailMate on MacOSX with SMTPUTF8 enabled.
  - Using his own mail server infrastructure.

- User 2:
  - peter@example.ca
  - Using Gmail web interface: no configuration necessary apart from the domain and user.
  - Using Google mail server infrastructure.
Demonstration: Setup

- For User 1 (kévin@example.com),
  - Its inbound email server is Courier as IMAP server.
  - Its outbound email server is Courier as SMTP server.
  - Courier uses a Postfix relay server acting as a pure MTA.

- For User 2 (peter@example.ca),
  - Its inbound/outbound email servers are Google Gmail servers.
Demonstration: Setup

- Confirming advertised mail server (Postfix) for kévin@example.com:
  - `dig xn--example-xta.com mx`
    - `xn--example-xta.com. 300 IN MX 10 postfix.xn--example-xta.com.`
Demonstration: Setup

- Confirming advertised mail server (Gmail) for peter@example.ca:
  - `dig xn--exmple-xta.ca mx`
    - `xn--exmple-xta.ca. 2878 IN MX 5 ALT1.ASPMX.L.GOOGLE.COM.`
    - `xn--exmple-xta.ca. 2878 IN MX 10 ALT4.ASPMX.L.GOOGLE.COM.`
    - `xn--exmple-xta.ca. 2878 IN MX 1 ASPMX.L.GOOGLE.COM.`
    - `xn--exmple-xta.ca. 2878 IN MX 5 ALT2.ASPMX.L.GOOGLE.COM.`
    - `xn--exmple-xta.ca. 2878 IN MX 10 ALT3.ASPMX.L.GOOGLE.COM.`
Demonstration: Setup
Postfix: Configuration

These are the specific EAI configuration requirements.

/etc/postfix/main.cf

...

# enable SMTPUTF8
smtputf8_enable = yes

#defines the hostname for SMTP. Does not need to be an IDN.
myhostname = postfix.xn--exmple-xta.com

#defines the domain of the host. Does not need to be an IDN.
mydomain = xn--exmple-xta.com

#domains of the user mailboxes. With smtputf8_enable, this is the key config.
virtual_mailbox_domains = exåmple.com
These are the specific EAI configuration requirements.

/etc/courier/defaultdomain
# defines the default domain used by Courier.
exâmple.com

/etc/courier/locals
# defines the local domain Courier will deliver mail to.
exâmple.com
localhost
Configuration MailMate

- **MailMate**
  - Commercial email client on MacOSX.

- For user kévin@example.com

- Using courier.xn--exemple-xta.com running Courier software as its IMAP and SMTP server.

- Set support for SMTPUTF8:
  - Defaults write `com.freron.MailMate MmSMTPUTF8Enabled -bool YES`
Demonstration: Action!
Email sent from kévin@example.com to peter@example.ca

From: Kévin <kévin@example.com>
Subject: test EAI
Date: October 31, 2019 at 15:40
To: peter@example.ca

Ceci est un test EAI.
Kévin

Message sent from MailMate.

Message received on Gmail.
The next slides show the raw message received on Gmail.

Specific EAI data is shown in bold.
Delivered-To: peter@example.ca
Received: by 2002:a1c:acc7:0:0:0:0:0 with SMTP id v190csp1632204wme; Thu, 31 Oct 2019 12:40:05 -0700 (PDT)
X-Google-Smtp-Source: APXvYqw6KEBvljPGOjeAeXkOk9CdEAsbKkwwNEqP71nJu509fNW/d154WhKl/qvl6gghoKoR3bpA
X-Received: by 2002:aa7:9156:: with SMTP id 22mr8654416pfi.246.1572550805097; Thu, 31 Oct 2019 12:40:05 -0700 (PDT)
ARC-Seal: i=1; a=rsa-sha256; t=1572550805; cv=none; d=google.com; s=arc-20160816;
b=yBPcElip4mz7ptqUCo+wIIp2uqcMEHv6ghwwNswV9hxg9Wzm7MUI076oW/UOq7I GnR ...
ARC-Message-Signature: i=1; a=rsa-sha256; c=relaxed/relaxed; d=google.com; s=arc-20160816; h=content-transfer-encoding:mime-version:message-id:date:subject:to:from; bh=MvG608jUZtntMhp8AedaXRMGgGznkDYwJKIKR2exBnA=; b=k8szchRBbaOrsTHIgmLqtBWgCz7hKMzrBxb6BYD1h0uyTRm/mpmnvMU+InofYbN8wS ...
ARC-Authentication-Results: i=1; mx.google.com; spf=pass (google.com: domain of kévin@example.com designates 2607:5300:60:1ecd::47 as permitted sender)
smtplib.mailfrom=kévin@example.com
Return-Path: <kévin@example.com>
Gmail Received Raw Message

Received: from postfix.xn--exmples-xta.com (postfix.xn--exmples-xta.com.
[2607:5300:60:1ecd::47])
  by mx.google.com with UTF8SMTP id 2si8923989pld.6.2019.10.31.12.40.03
  for <peter@exâmple.ca>;
  Thu, 31 Oct 2019 12:40:04 -0700 (PDT)
Received-SPF: pass (google.com: domain of kévin@exâmple.com designates
2607:5300:60:1ecd::47 as permitted sender) client-ip=2607:5300:60:1ecd::47;
Authentication-Results: mx.google.com; spf=pass (google.com: domain of
kévin@exâmple.com designates 2607:5300:60:1ecd::47 as permitted sender)
smtplib.mailfrom=kévin@exâmple.com
Received: from courier (courier.xn--exmples-xta.com [144.217.171.46]) by
postfix.xn--exmples-xta.com (Postfix) with UTF8SMTP id 72C72100366 for
<peter@exâmple.ca>; Thu, 31 Oct 2019 19:40:02 +0000 (UTC)
Received: from [192.168.1.59] (modemcable138.218-70-69.provider.ca
[::ffff:69.70.218.138])
  (AUTH: PLAIN kévin)
  by courier with UTF8ESMTPA
  id 0000000000100340.000000005DBB3892.00006017; Thu, 31 Oct 2019
  19:40:02 +0000
This is an EAI test email.

Kevin.
Considerations Deploying EAI: Case Mapping and Matching
Trying Uppercase

- Mail sent from peter@viagenie.ca to KÉVIN@example.com
  - Email was NOT delivered.

Why?
- kevin and KEVIN as local parts are typically handled as same user.
  - i.e. case folding is automatic for ASCII local parts.
- But kévin and KÉVIN are by default not the same user. Unicode case folding is typically not done by servers.
Postfix log (postfix.xn--exemple-xta.com is the email server receiving emails for exâmple.com).

Oct 31 20:30:32 postfix postfix/smtp[11643]: C6531100366: to=<KÉVIN@exâmple.com>, relay=144.217.171.46[144.217.171.46]:25, delay=8.5, delays=0.01/0.01/0.18/8.3, dsn=5.0.0, status=bounced (host 144.217.171.46[144.217.171.46] said: 550 User <KÉVIN> unknown (in reply to RCPT TO command))
Oct 31 20:30:48 postfix postfix/cleanup[11642]: 503A21003EF: message-id=<20191031203048.503A21003EF@postfix.xn--exemple-xta.com>
Oct 31 20:30:48 postfix postfix/qmgr[8907]: 503A21003EF: from=<> size=4724 nrcpt=1 (queue active)
Oct 31 20:30:48 postfix postfix/qmgr[8907]: C6531100366: removed
Oct 31 20:30:49 postfix postfix/smtp[11643]: 503A21003EF: to=<peter@viagenie.ca>, relay=ASPMX.L.GOOGLE.COM[209.85.201.26]:25, delay=0.77, delays=0/0/0.35/0.42, dsn=2.0.0, status=sent (250 2.0.0 OK 1572553849 g54si5822082qtk.32 - gsmtp

Supporting case mapping is complicated to set up; creates a very larger number of combinations for local part and domain part.
Matching EAI

- As previously discussed, kévin@example.com may be represented in Unicode in various ways.

- Since one does not know if the other party (user, other server or process) is sending normalized strings, then everyone should normalize before any matching.
  - Usually not an issue as input methods generally normalize but...

- Typically in ASCII email addresses, the local part can be uppercase or lowercase. In UTF-8, case folding is not simple.
Matching EAI

- EAI with IDNs have the side effect of multiple representations of an email address:
  - kévin@example.com
  - kévin@xn--example-xta.com
- Some M*A may decide to convert from one representation to another.
- From a receiving M*A perspective, users should be defined for _both_ domains in the M*A configuration.
  - Postfix:
    - mydestination = $alabel.tld, $ulabel.tld
Considerations Deploying EAI: Spam
Avoiding Interpretation as Spam

- Not specifically related to EAI, email domains should now have various DNS entries to tell other servers about the legitimacy of the server’s sending email.

- SPF record is a TXT DNS record at the anchor of a domain that specifies which email servers are authorized to send email from that domain.

- Therefore, as with any other email domain, the SPF record should be properly configured in the DNS under the email domain.
Avoiding Interpretation as Spam

- Similarly, DKIM record is a TXT DNS record at the anchor of a domain that contains the public key used to sign email headers.

- Therefore, as with any other email domain, if your email service uses DKIM, the DKIM record should be properly configured in the DNS under the email domain.
Avoiding Interpretation as Spam

- Some spam filtering software attempt to do DNS reverse lookups on the IP address of the server sending the email. They often also verify the Received-by headers.

- If the reverse lookup is not properly mapped, it may result in classifying these emails as potential spam. Note this has nothing to do with EAI or IDNs.

- To mitigate this, make sure that your server’s IP addresses are properly mapped in the DNS to their primary name using the DNS PTR records.
Avoiding Interpretation as Spam

- Spam filtering may think EAI is spam even with SPF, DKIM, etc.
  - This is because the spam filtering does not know about EAI.

- New TLDs and IDNs may also be viewed as spam.

- Some Mail User Agent software uses your contacts database to determine if an email is known to you or not. This will make it more likely to be considered spam if the email of the sender is not in your contacts database.

- Make sure the contacts database contains the EAI email of the contacts.
Considerations Deploying EAI: MX Consistency, Mailing Lists, Delivery Notifications, and User Accounts
Delivery Path Considerations: MX Consistency

- If a domain has multiple email servers all identified by an MX record, then all email servers MUST support EAI.

- This is because any one of these servers can be used to receive emails.
Other Considerations

- EAI end users may be sending emails to others (non-EAI) that claimed they never received the emails, even if their email infrastructure supports EAI. This can happen because:
  - Somewhere in the path: wrong configuration, dropping email, unable to send delivery notification messages.
  - Spam filtering.

- In server and OS configurations, it is preferred to use A-labels instead of U-labels.
Mailing Lists

- Have a set of both EAI and non-EAI addresses and users.
- Mailing list reflector must deal with both.
Delivery Notifications

- Delivery notifications are used by end users to request a notification when an email is delivered. This may not be supported by the destination because of configuration or policy.
  - Given that the notification is taking the reverse path, it may not take the same path as the forward one.
  - Therefore, the delivery notification itself may encounter non-EAI supporting SMTP servers while the forward path was working fine.
User Accounts

- Typically, mail server software has two ways to set up mail user accounts:
  - Using system (i.e. OS) accounts.
  - Using a table/database/file of email user accounts (often called virtual users in the mail server software) that are not system accounts.
    - This is the preferred way as it is much more flexible and secure to remove/add/change email users without any change in the OS.

- Underlying OS may have limitations on UTF-8 system accounts.

- Preferable to use virtual users.
Mail Software and Services

- EAI Support:
  - MailMate (MUA) on MacOSX: v.1.9.4 minimum
  - Postfix (SMTP) v3.0 minimum
    - http://www.postfix.org/SMTPUTF8_README.html
  - Courier (IMAP, POP, SMTP) v1.0 minimum (IMAP version 5.0.8)

- EAI NOT supported (as of October 2019):
  - Dovecot (IMAP, POP)
  - Zimbra
  - Mozilla Thunderbird

- Services supporting EAI:
  - Gmail
  - Outlook

- More info in UASG021A
Conclusion

- EAI is essentially supporting UTF-8 local parts of an email address.
  - Which also means supporting in the headers.
  - Requires changes to mail servers and mail clients.

- All SMTP servers in the path must be EAI-enabled to deliver the mail to the destination.
Conclusion

- Courier and Postfix are two open-source software that support EAI with very limited changes in configuration.

- In configuration files, be careful to use A-labels, U-labels, or both, depending on the situation.
EAI RFCs


Additional Information for EAI, IDNs, UA

- Universal Acceptance Steering Group (UASG)
  - [https://uasg.tech](https://uasg.tech)
  - [https://www.icann.org/ua](https://www.icann.org/ua)

- Internationalized Domain Names (IDNs)
  - [http://icann.org/idn](http://icann.org/idn)
Engage with ICANN

Thank You and Questions
Visit us at icann.org and uasg.tech
Email: UAProgram@icann.org and info@uasg.tech

@icann linkedin/company/icann
facebook.com/icannorg slideshare/icannpresentations
youtube.com/icannnews soundcloud/icann
flickr.com/icann instagram.com/icannorg