



Global Registry Services

IDN Manual for .GLOBAL

Prepared by Afilias

March, 2017



Copyright © 2015 Afilias Limited. All rights reserved. Afilias has made efforts to ensure the accuracy and completeness of the information in this document. However, Afilias makes no warranties of any kind (express or implied) with respect to the information contained herein. Afilias assumes no liability to any party for any loss or damage (whether direct or indirect) caused by any errors, omissions, or statements of any kind contained in this document. Afilias may change details of its service over time.

The information contained herein is proprietary and confidential information. It may be used by the recipient only for the purpose for which it was transmitted and shall be returned upon request or when no longer needed by recipient. This information is submitted in confidence and its disclosure to you is not intended to constitute public disclosure or authorization for disclosure to other parties. This document and the information contained herein may not be copied or communicated without the prior written consent of Afilias.

Table of Contents

- A. ICANN guidelines 1
- B. A-Label & Punycode 1
- A. A-Label Conversion of IDNs 4
- B. Registrar Tools 6
 - 1. IDN EPP Registrar Tool-Kit (RTK) Package 7
 - 2. EPP OT&E Environment 7
- C. EPP Command IDN Extensions 7
- D. Whois 24
- E. DNS Resolution 25
 - 1. IDNA Compliant Browsers 25
 - 2. Non-IDNA Compliant Browsers 25
- A. Simplified Chinese & Traditional Chinese 26
- B. Bosnian, Bulgargian, Belarusian, Macedonian, Montenegrin, Russian, Serbian, & Ukrainian 27
- C. Danish, Hungarian, Icelandic, Latvian, Lithuanian & Swedish 28
- D. Korean 29
- E. Spanish 29
- F. German 30
- G. Arabic 30
- H. Hindi 31
- I. Polish 31
- J. Portuguese 32
- K. Italian 32
- L. French 32
- M. Finnish 33



I. IDN Overview

A. ICANN guidelines

In September of 2011, ICANN released IDN Implementation Guidelines. The ICANN guidelines require that a registry operator develops its IDN solution to be compliant with RFCs 5890, 5891, 5892, 5893 and 5895.

In addition, it is required that the launch of the solution be tailored to each language or set of languages based on work with local stakeholders and experts. These guidelines also require a registry operator's IDN solution to utilize the adopted A-Label & Punycode standard for registration.

The ICANN guidelines prohibit:

- a. Line symbol-drawing characters,
- b. Symbols and icons that are neither alphanumeric nor ideographic language characters, such as typographical and pictographical dingbats,
- c. Punctuation characters, and
- d. Spacing characters. Appendix B.1 of RFC 5892 provides the full list of "DISALLOWED" and "UNASSIGNED" code points.

B. A-Label & Punycode

A-Label specifies the syntax and punycode in the encoding algorithm designed for use with Internationalized Domain Names in Applications (IDNA—RFC 5890). It translates a Unicode string into an ASCII string that can be registered by a domain name registry and resolved through the DNS.

To convert IDNs, labels containing non-ASCII characters can be represented by ACE (ASCII Compatible Encoding) labels which begin with a special ACE prefix and contain only ASCII characters. The prefix is "xn—" and the remainder of the label following the prefix is the encoded Unicode string by punycode encoding as specified under IDNA 2008 specifications. The Unicode string must be a valid U-Label. If the string contains any of the following characteristics (not a valid U-Label), then it MUST be rejected prior to DNS lookup:

- The label is not in Unicode Normalization Form C (NFC).
- The label contains a hyphen (-) in the third and the fourth character positions "--".
- The label whose first character is a combining mark.
- The label contains prohibited code points, i.e., those that are assigned to the "DISALLOWED" category of the Tables document [[RFC5892](#)].
- The label contains code points that are identified in the Tables document as "CONTEXTJ", i.e., requiring exceptional contextual rule processing on lookup, but that do not conform to those rules. Note that this implies that a rule must be defined, not null: a character that requires a contextual rule but for which the rule is null is treated in this step as having failed to conform to the rule.
- The label contain code points that are identified in the Tables document as "CONTEXTO", but for which no such rule appears in the table of rules. Applications resolving DNS names or carrying out equivalent operations are not required to test contextual



rules for "CONTEXTO" characters, only to verify that a rule is defined (although they MAY make such tests to provide better protection or give better information to the user).

- The label contains code points that are unassigned in the version of Unicode being used by the application, i.e., in the UNASSIGNED category of the Tables document. This requirement means that the application must use a list of unassigned characters that is matched to the version of Unicode that is being used for the other requirements in this section. It is not required that the application know which version of Unicode is being used; that information might be part of the operating environment in which the application is running.
- It is verified that the string is compliant with the requirements for right-to-left characters specified in the Bi-directional (bidi) document [[RFC5893](#)].

The string that has now been validated for lookup is converted to A-Label by applying the Punycode algorithm to the string and then adding the prefix ("xn--").

The general algorithm that applications ought to implement in order to produce Unicode code points that will be valid under the IDNA protocol:

- Uppercase characters are mapped to their lowercase equivalents by using the algorithm for mapping case in Unicode characters. This step was chosen because the output will behave more like ASCII host names behave.
- Fullwidth and halfwidth characters (those defined with Decomposition Types <wide> and <narrow>) are mapped to their decomposition mappings as shown in the Unicode character database. This step was chosen because many input mechanisms, particularly in Asia, do not allow you to easily enter characters in the form used by IDNA2008. Even if they do allow the correct character form, the user might not know which form they are entering.
- All characters are mapped using Unicode Normalization Form C (NFC). This step was chosen because it maps combinations of combining characters into canonical composed form. As with the fullwidth/halfwidth mapping, users are not generally aware of the particular form of characters that they are entering, and IDNA2008 requires that only the canonical composed forms from NFC be used.
- [[IDNA2008protocol](#)] is specified such that the protocol acts on the individual labels of the domain name. If an implementation of this mapping is also performing the step of separation of the parts of a domain name into labels by using the FULL STOP character (U+002E), the IDEOGRAPHIC FULL STOP character (U+3002) can be mapped to the FULL STOP before label separation occurs. There are other characters that are used as "full stops" that one could consider mapping as label separators, but their use as such has not been investigated thoroughly. This step was chosen because some input mechanisms do not allow the user to easily enter proper label separators. Only the IDEOGRAPHIC FULL STOP character (U+3002) is added in this mapping because the authors have not fully investigated the applicability of other characters and the environments where they should and should not be considered domain name label separators.

Please see the following guidelines for more information:

- ICANN guidelines <http://www.icann.org/en/resources/idn/implementation-guidelines>
- RFC 5890 <http://tools.ietf.org/html/rfc5890>
- RFC 5891 <http://tools.ietf.org/html/rfc5891>



- RFC 5892 <http://tools.ietf.org/html/rfc5892>
- RFC 5893 <http://tools.ietf.org/html/rfc5893>
- RFC 5894 <http://tools.ietf.org/html/rfc5894>
- RFC 5895 <http://tools.ietf.org/html/rfc5895>
- RFC 3492 <http://tools.ietf.org/html/rfc3492>

Internationalized Domain Names Registration and Administration Guidelines for IDNs

- RFC 4713 (Chinese) <http://www.rfc-editor.org/rfc/rfc4713.txt>
- RFC 5564 (Arabic) <http://tools.ietf.org/html/rfc5564>
- RFC 5992 (Cyrillic) <http://www.rfc-editor.org/rfc/rfc5992.txt>

II. Registration process

A. A-Label Conversion of IDNs

In order for a registrar to process an IDN registration, first the registrar accepts the desired domain in their preferred encoding – such as UTF-8, Unicode, etc. - from their customer. In some cases, this might require some data transformation functions to be written by registrars for their front-end web-based applications.

The registry operator has released an IDN EPP Registrar Tool-Kit (RTK) Package to assist registrars in understanding the additional components of the IDN registration process. There are two new components of the registration process that registrars should note:

1. Conversion of encoded registration to A-Label using the conversion engine contained in the IDN EPP RTK.
2. Use of new XML extensions to process IDN registrations, including the use of a Language Tag to identify the script of the IDN name to be registered. The conversion engine from IDN EPP RTK will enable registrars to convert the encoded names received from their customers into A-Label domain names that can be registered with the Registry. Please see the IDN EPP RTK and associated README files for details on this tool.

Upon receipt of the IDN registration request, use the `easyEncodeDomain` function, which is part of the IDN EPP RTK, to convert the request with the correct A-Label domain. Once this is complete, the A-Label domain must be declared with the proper language tag (for example: ZH-CN for Simplified Chinese).

Registrars will be able to use the `easyDecodeDomain` function in the conversion engine to similarly convert a registered A-Label domain into the corresponding IDN (into their native form for display back to the registrant). It is the registrar's responsibility to ensure that a registrant's desired IDN is processed correctly. This requires ensuring that the original encoded name is processed through `easyEncodeDomain` functions in order to generate the correct A-Label string.

Illegal characters in the A-Label string (std3) will be rejected as per normal registry operations.

A language tag is defined as:

Language Tag: The script identifier that categorizes the domain name by language script. Registrars must enter a language tag (e.g. RU or UK for Cyrillic IDNs) when registering an IDN.

The following is the current list of IDNs that could be supported by a registry, availability subject to each TLD's application to ICANN.

Language	Tag
Arabic	AR
Hindi	HI
Bosnian	BS
Bulgarian	BG
Belarusian	BE
Macedonian	MK



Language	Tag
Montenegrin	ME ¹
Russian	RU
Serbian	SR
Ukrainian	UK
Chinese (Traditional)	ZH-TW
Chinese (Simplified)	ZH-CN
Danish	DA
German	DE
Hungarian	HU
Icelandic	IS
Korean (Hangul)	KO
Latvian	LV
Lithuanian	LT
Polish	PL
Spanish	ES
Swedish	SV
French	FR
Italian	IT
Portuguese	PT
Finnish	FI

Registrars will use the XML extensions available in the IDN EPP RTK in order to insert the appropriate language tag and declare a name as an IDN. Please see the IDN EPP RTK and its associated README files for additional details on this process. Please note that IDN names that do not have a language tag associated with them will be rejected.

Registrars will use the EPP-Check command to determine if this name is available for registration. Following confirmation that the domain is not registered, the registrar may then submit the name for creation in the registry.

¹ The IANA repository refers to SR-ME as the language tag for the Montenegrin.

Registrars will be able to use the 'To Unicode' function in the mapping engine to similarly convert registered A-Label strings into the corresponding IDN (into their native form for display back to the registrant).

It is the registrar's responsibility to ensure that a registrant's desired IDN is processed correctly. This requires ensuring the original encoded name is processed through a valid conversion (such as 'ToASCII' function in RTK) in order to generate the correct A-Label.

B. Registrar Tools

In order to assist registrars with registration of IDNs, a number of tools have been prepared. These include the IDN EPP Registrar Tool-Kit (RTK) Package, OT&E testing environment, and assistance from Technical Support to address key issues.

1. IDN EPP Registrar Tool-Kit (RTK) Package

The IDN EPP RTK Package is downloadable from Sourceforge (<http://sourceforge.net/projects/epp-rtk/>). The IDN EPP RTK is available in C++ (no longer supported nor updated as of 2011) and Java versions and contains technical explanations, examples and code that registrars will need to process and transmit IDNs to the registry.

Included with the IDN EPP RTK Package are a number of README files that explain elements of the RTK including the mapping engine, as well as, describe the XML extensions that registrars may use to register IDNs. The mapping engine contains 'To ASCII' and 'To Unicode' functions to convert IDN registrations into and out of A-Label, as well as, XML extensions for handling IDN registrations.

Registrars should download the IDN EPP RTK and associated README files immediately and begin testing in the OT&E environment to ensure that they have a sufficient understanding of the conversion process and XML extensions before the launch of live IDN registrations.

2. EPP OT&E Environment

To assist registrars with understanding and testing the A-Label conversion process, the registry operator has made available in the EPP OT&E environment the optional testing of IDNs. It is recommended that all registrars (both in Ramp up stage and Live) should use the OT&E environment to verify their EPP client code with IDNs registrations. Registrars may use the IDN EPP RTK Package to make the necessary adjustments to their EPP client code when testing in the OT&E environment.

Please connect using your OT&E environment usernames and passwords.

Technical Support

Registrars may contact Technical Support at +1.855.723.0999, +1.416.619.3033 or techsupport@afilias-srs.net should you require any assistance with accessing the OT&E environment.

Technical Support will be available to registrars to assist them in implementing the IDN EPP RTK and in utilizing the IDN OT&E environment to ensure that their clients are adequately configured to process IDN registrations.

Technical support is available 24/7/365 and can be reached at: +1.855.723.0999, +1.416.619.3033 or techsupport@afilias-srs.net.

C. EPP Command IDN Extensions

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [RFC5730]. The command mappings described here are specifically for use in implementing internationalized domain name processes via EPP.

EPP <check> Command

This extension defines additional elements to extend the EPP <check> command described in the EPP domain mapping [RFC5731] for IDN check. No additional elements are defined for EPP <check> response.

Domain name validation logic must allow A-Label names.



If the <check> command contains at least one <domain:name> element with an IDN value, then in addition to the standard EPP command elements, the <check> command MUST contain an <idn:check> extension element.

The <idn:check> element contains the following child elements:

- An <idn:script> element that contains the script name.

Example <check> command:

```
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0">
C: <command>
C: <check>
C:   <domain:check
C:     xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'>
C:     <domain:name>xn--bq-uia.info</domain:name>
C:   </domain:check>
C: </check>
C: <clTRID>CLI-1097596300572</clTRID>
C: <extension>
C:   <idn:check xmlns:idn='urn:afllias:params:xml:ns:idn-1.0'>
C:     <idn:script>de</idn:script>
C:   </idn:check>
C: </extension>
C: </command>
C:</epp>
```

A <check> command can contain more than one IDN name to check, in such cases, only a single <idn:script> element should be specified.

Example <check> command specifying multiple IDN names.

```
C:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
C: <command>
C: <check>
C:   <domain:check
C:     xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'>
C:     <domain:name>xn--oq-xka.info</domain:name>
C:     <domain:name>xn--bq-uia.info</domain:name>
C:   </domain:check>
C: </check>
C: <clTRID>CLI-1065207438144</clTRID>
```

```

C: <extension>
C: <idn:check xmlns:idn='urn:afiliass:params:xml:ns:idn-1.0'>
C:   <idn:script>de</idn:script>
C: </idn:check>
C: </extension>
C: </command>
C:</epp>

```

An EPP error response will return for a <check> command that contains at least one <domain:name> element with an IDN value that does not contain <idn:check> or <idn:script> extensions.

Example <check> response when the <idn:check> element is omitted:

```

S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S: <response>
S: <result code='2003'>
S:   <msg lang='en-US'>Required parameter missing</msg>
S:   <value xmlns:oxrs='urn:afiliass:params:xml:ns:oxrs-1.1'>
S:     <oxrs:xcp>2003:Required parameter missing
S:(idn:check)</oxrs:xcp>
S:   </value>
S: </result>
S: <trID>
S:   <clTRID>CLI-1065207438144</clTRID>
S:   <svTRID>SRO-1097598449989</svTRID>
S: </trID>
S: </response>
S:</epp>

```

Example <check> response when the <idn:script> element is omitted:

```

S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S: <response>
S: <result code='2003'>
S:   <msg lang='en-US'>Required parameter missing</msg>
S:   <value xmlns:oxrs='urn:afiliass:params:xml:ns:oxrs-1.1'>
S:     <oxrs:xcp>2003:Required parameter missing
S:(epp.command.extension.idn:check.idn:script)</oxrs:xcp>

```

```

S: </value>
S: </result>
S: <trID>
S: <clTRID>CLI-1065207438144</clTRID>
S: <svTRID>SRO-1097598809161</svTRID>
S: </trID>
S: </response>
S:</epp>

```

When a <check> request contains an IDN whose translated A-Label <domain:name> value contains character(s) that are not in the script table specified in the <idn:script> element value, the <domain:cd> element in response will indicate that the IDN is not available.

Example <check> response when <domain:name> value has a conflict with the value of <idn:script> element:

```

S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S: <response>
S: <result code='1000'>
S: <msg lang='en-US'>Command completed successfully</msg>
S: </result>
S: <resData>
S: <domain:chkData
S:   xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'>
S: <domain:cd>
S: <domain:name avail='0'>xn--dn-mja.info</domain:name>
S: <domain:reason>Invalid code point - U+ed</domain:reason>
S: </domain:cd>
S: </domain:chkData>
S: </resData>
S: <trID>
S: <clTRID>CLI-1065207438144</clTRID>
S: <svTRID>SRO-1097599157873</svTRID>
S: </trID>
S: </response>
S:</epp>

```


EPP <info> Command

XML syntax for domain info response will be modified for IDN domain names. Extension element <idn:infData> will be added. The element contains the following elements:

- An <idn:script> element that contains a script name.

Example <info> response for an IDN name:

```
S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S: <response>
S:   <result code='1000'>
S:     <msg lang='en-US'>Command completed successfully</msg>
S:   </result>
S:   <resData>
S:     <domain:infData
S:       xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'>
S:       <domain:name>xn--bq-uia.info</domain:name>
S:       <domain:roid>D224-LRMS</domain:roid>
S:       <domain:status s='ok'/>
S:       <domain:registrant>OTNE-C1</domain:registrant>
S:       <domain:contact type='tech'>OTNE-C2</domain:contact>
S:       <domain:contact type='admin'>OTNE-C3</domain:contact>
S:       <domain:contact type='billing'>OTNE-C4</domain:contact>
S:       <domain:ns>
S:         <domain:hostObj>ns1.valid.info</domain:hostObj>
S:         <domain:hostObj>ns2.valid.info</domain:hostObj>
S:       </domain:ns>
S:       <domain:clID>ClientA</domain:clID>
S:       <domain:crID>ClientA</domain:crID>
S:       <domain:crDate>2004-10-12T17:57:41.0Z</domain:crDate>
S:       <domain:exDate>2006-10-12T17:57:41.0Z</domain:exDate>
S:       <domain:authInfo>
S:         <domain:pw>foo-BAR</domain:pw>
S:       </domain:authInfo>
S:     </domain:infData>
S:   </resData>
S:   <extension>
S:     <idn:infData xmlns:idn='urn:afllias:params:xml:ns:idn-1.0'>
S:       <idn:script>de</idn:script>
S:     </idn:infData>
S:   </extension>
S: </trID>
```

	<pre> S: <clTRID>CLI-1097604691520</clTRID> S: <svTRID>SRO-1097604691524</svTRID> S: </trID> S: </response> S:</epp> </pre>
--	---

EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response.

EPP <create> Command

The extension defines additional elements for the EPP <create> command but no additional elements for the EPP <create> response.

The EPP <create> command must contain an <extension> element, and the <extension> element must contain a child <idn:create> element that identifies the extension namespace if the client wants to associate data defined in this extension to the domain object. The <idn:create> element contains the following child elements:

- An <idn:script> element that indicates an IDN script name.

EPP <create> request for a non-IDN name that contains an IDN extension will be accepted. The IDN extension will be ignored in such case.

Example <create> command and its response:

	<pre> C:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'> C: <command> C: <create> C: <domain:create C: xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'> C: <domain:name>xn--bq-ua.info</domain:name> C: <domain:period unit='y'>2</domain:period> C: <domain:authInfo> C: <domain:pw>foo-BAR</domain:pw> C: </domain:authInfo> C: <domain:ns> C: <domain:hostObj>ns1.valid.info</domain:hostObj> C: <domain:hostObj>ns2.valid.info</domain:hostObj> C: </domain:ns> C: <domain:registrant>OTNE-C1</domain:registrant> C: <domain:contact type='tech'>OTNE-C2</domain:contact> </pre>
--	---

	<pre> C: <domain:contact type='billing'>OTNE-C4</domain:contact> C: <domain:contact type='admin'>OTNE-C3</domain:contact> C: </domain:create> C: </create> C: <clTRID>CLI-1097602657409</clTRID> C: <extension> C: <idn:create xmlns:idn='urn:afiliass:params:xml:ns:idn-1.0'> C: <idn:script>de</idn:script> C: </idn:create> C: </extension> C: </command> C:</epp> </pre>
	<pre> S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'> S: <response> S: <result code='1000'> S: <msg lang='en-US'>Command completed successfully</msg> S: </result> S: <resData> S: <domain:creData S: <xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'> S: <domain:name>xn--bq-ua.info</domain:name> S: <domain:crDate>2004-10-12T17:57:41.0Z</domain:crDate> S: <domain:exDate>2006-10-12T17:57:41.0Z</domain:exDate> S: </domain:creData> S: </resData> S: <trID> S: <clTRID>CLI-1097603861301</clTRID> S: <svTRID>SRW-6011</svTRID> S: </trID> S: </response> S:</epp> </pre>

EPP domain create request for an IDN name that has a missing `<idn:create>` extension will return an error response with response code 2003 ("Required parameter missing").

Example `<create>` command with the `<idn:create>` element missing:

	<pre>C:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'></pre>
--	---

```

C: <command>
C: <create>
C: <domain:create
C:   xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'>
C:   <domain:name>xn--bq-uia.info</domain:name>
C:   <domain:period unit='y'>2</domain:period>
C:   <domain:authInfo>
C:     <domain:pw>foo-BAR</domain:pw>
C:   </domain:authInfo>
C:   <domain:ns>
C:     <domain:hostObj>ns1.valid.info</domain:hostObj>
C:     <domain:hostObj>ns2.valid.info</domain:hostObj>
C:   </domain:ns>
C:   <domain:registrant>OTNE-C1</domain:registrant>
C:   <domain:contact type='admin'>OTNE-C2</domain:contact>
C:   <domain:contact type='billing'>OTNE-C3</domain:contact>
C:   <domain:contact type='tech'>OTNE-C4</domain:contact>
C: </domain:create>
C: </create>
C: <cITRID>CLI-1088191770366</cITRID>
C: </command>
C:</epp>

```

Example of corresponding <create> response for a request with the <idn:create> element missing:

```

S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S: <response>
S: <result code='2003'>
S: <msg lang='en-US'>Required parameter missing</msg>
S: <value xmlns:oxrs='urn:afilias:params:xml:ns:oxrs-1.1'>
S: <oxrs:xcp>2003:Required parameter missing
S:(idn:create)</oxrs:xcp>
S: </value>
S: </result>
S: <trID>
S: <cITRID>CLI-1088191770366</cITRID>
S: <svTRID>SRW-6015</svTRID>
S: </trID>
S: </response>

```

```
S:</epp>
```

EPP domain create request for an IDN name that has a missing `<idn:script>` element in `<idn:create>` extension will return an error response with response code 2003 ("Required parameter missing").

Example `<create>` command with the `<idn:script>` element missing:

```
C:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
C: <command>
C: <create>
C: <domain:create
C:   xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'>
C: <domain:name>xn--bq-ua.info</domain:name>
C: <domain:period unit='y'>2</domain:period>
C: <domain:authInfo>
C: <domain:pw>foo-BAR</domain:pw>
C: </domain:authInfo>
C: <domain:ns>
C: <domain:hostObj>ns1.valid.info</domain:hostObj>
C: <domain:hostObj>ns2.valid.info</domain:hostObj>
C: </domain:ns>
C: <domain:registrant>OTNE-C1</domain:registrant>
C: <domain:contact type='admin'>OTNE-C2</domain:contact>
C: <domain:contact type='billing'>OTNE-C3</domain:contact>
C: <domain:contact type='tech'>OTNE-C4</domain:contact>
C: </domain:create>
C: </create>
C: <clTRID>CLI-1088191770366</clTRID>
C: <extension>
C: <idn:create xmlns:idn='urn:afiliass:params:xml:ns:idn-1.0'>
C: </idn:create>
C: </extension>
C: </command>
C:</epp>
```

Example of corresponding `<create>` response for a request with the element `<idn:script>` missing:

```
S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
```

```

S: <response>
S: <result code='2003'>
S: <msg lang='en-US'>Required parameter missing</msg>
S: <value xmlns:oxrs='urn:afiliass:params:xml:ns:oxrs-1.1'>
S: <oxrs:xcp>2003:Required parameter missing
S:(epp.command.extension.idn:create.idn:script)</oxrs:xcp>
S: </value>
S: </result>
S: <trID>
S: <clTRID>CLI-1088191770366</clTRID>
S: <svTRID>SRW-6016</svTRID>
S: </trID>
S: </response>
S:</epp>

```

For EPP domain create request with an invalid value for <idn:script>, the error response will return with error code 2306 ("Parameter value policy error"). A detailed error message is also returned in the response.

Example <create> response for an invalid script name:

```

S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S: <response>
S: <result code='2306'>
S: <msg lang='en-US'>Parameter value policy error</msg>
S: <value xmlns:oxrs='urn:afiliass:params:xml:ns:oxrs-1.1'>
S: <oxrs:xcp>
S: 2306:Parameter value policy error (ScriptType:de-at)
S: </oxrs:xcp>
S: </value>
S: </result>
S: <trID>
S: <clTRID>CLI-1088191770366</clTRID>
S: <svTRID>SRW-6020</svTRID>
S: </trID>
S: </response>
S:</epp>

```

EPP <delete> Command

This extension does not add any elements to the EPP <delete> command or <delete> response.

EPP <renew> Command

This extension does not add any elements to the EPP <renew> command or <renew> response.

EPP <update> Command

The extension defines additional elements to extend the EPP <update> command but no additional elements to <update> response.

The EPP <update> command provides a transform operation that allows a client to change the state of a domain object. The registry IDN extension modifies base update processing to support defining language script for IDN names that don't have the parameter defined yet. Such a situation may arise when an IDN name exists in the system as a result of a data conversion.

Section 3.2.5 of the EPP domain mapping in RFC5731 describes the elements that have to be specified within an <update> command. The requirement to provide at least one <domain:add>, <domain:rem>, or <domain:chg> element is updated by this extension such that one empty <domain:chg> element must be present if this extension is specified within an <update> command. This requirement is updated to disallow the possibility of modifying a domain object as part of legacy IDN name processing.

In addition to the EPP command elements described in the EPP domain mapping in RFC5731, the <update> command MUST contain an <extension> element. The <extension> element MUST contain a child <idn:update> element that identifies the registry idn namespace and the location of the registry idn schema. The <idn:update> element contains a single <idn:chg> element that contains a single <idn:script> element that SHOULD be used to provide a language script for updating an existing already in the registry international domain name.

Example <update> command with an idn script and its response:

```
C:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
C: <command>
C: <update>
C:   <domain:update
C:     xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'>
C:   <domain:name>xn--bc-ua.info</domain:name>
C:   <domain:chg>
C:   </domain:chg>
C: </domain:update>
C: </update>
C: <cITRID>CLI-1102528331230</cITRID>
C: <extension>
C:   <idn:update xmlns:idn='urn:afiliass:params:xml:ns:idn-1.0'>
C:   <idn:chg>
```

	<pre> C: <idn:script>de</idn:script> C: </idn:chg> C: </idn:update> C: </extension> C: </command> C:</epp> </pre>
	<pre> S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'> S: <response> S: <result code='1000'> S: <msg lang='en-US'>Command completed successfully</msg> S: </result> S: <trID> S: <clTRID>CLI-1102528331230</clTRID> S: <svTRID>SRW-14003</svTRID> S: </trID> S: </response> S:</epp> </pre>

EPP domain update request for an IDN name that has a missing `<idn:script>` element in the `<idn:update>` extension will return an error response with response code 2003 ("Required parameter missing").

Example `<update>` command with `<idn:script>` element missing:

	<pre> C:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'> C: <command> C: <update> C: <domain:update C: xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'> C: <domain:name>xn--bc-ua.org</domain:name> C: <domain:chg> C: </domain:chg> C: </domain:update> C: </update> C: <clTRID>CLI-1102528331230</clTRID> C: <extension> C: <idn:update xmlns:idn='urn:afilias:params:xml:ns:idn-1.0'> C: <idn:chg> C: </idn:chg> </pre>
--	---


```
C: </idn:update>
C: </extension>
C: </command>
C:</epp>
```

Example <update> response for a request with the element <idn:script> missing:

```
S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S: <response>
S: <result code='2003'>
S: <msg lang='en-US'>Required parameter missing</msg>
S: <value xmlns:oxrs='urn:afilias:params:xml:ns:oxrs-1.1'>
S: <oxrs:xcp>2003:Required parameter missing
S:(epp.command.extension.idn:update.idn:chg.idn:script)</oxrs:xcp>
S: </value>
S: </result>
S: <trID>
S: <clTRID>CLI-1102528331230</clTRID>
S: <svTRID>SRW-14004</svTRID>
S: </trID>
S: </response>
S:</epp>
```

For EPP domain update request with invalid value for <idn:script>, the error response will return with error code 2306 ("Parameter value policy error"). A detailed error message is also returned in the response.

Example <update> response for an invalid script name:

```
S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
S: <response>
S: <result code='2306'>
S: <msg lang='en-US'>Parameter value policy error</msg>
S: <value xmlns:oxrs='urn:afilias:params:xml:ns:oxrs-1.1'>
S: <oxrs:xcp>
S: 2306:Parameter value policy error (Invalid script name:de-AT)
S: </oxrs:xcp>
S: </value>
S: </result>
```

	<pre> S: <trID> S: <clTRID>CLI-1102528331230</clTRID> S: <svTRID>SRW-14005</svTRID> S: </trID> S: </response> S:</epp> </pre>
--	---

For an EPP domain update request with conflicting <domain:name> and <idn:script> values, the error response will return with error code 2306 (“Parameter value policy error”). A detailed error message is also returned in the response.

Example of update command with conflicting <domain:name> and <idn:script> values, and its corresponding response:

	<pre> C:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'> C: <command> C: <update> C: <domain:update C: xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'> C: <domain:name>xn--bc-ua.info</domain:name> C: <domain:chg> C: </domain:chg> C: </domain:update> C: </update> C: <clTRID>CLI-1102528331230</clTRID> C: <extension> C: <idn:update xmlns:idn='urn:afiliass:params:xml:ns:idn-1.0'> C: <idn:chg> C: <idn:script>pl</idn:script> C: </idn:chg> C: </idn:update> C: </extension> C: </command> C:</epp> </pre>
	<pre> S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'> S: <response> S: <result code='2306'> S: <msg lang='en-US'>Parameter value policy error</msg> </pre>

```

S: <value xmlns:oxrs='urn:afiliass:params:xml:ns:oxrs-1.1'>
S:   <oxrs:xcp>
S:     2306:Parameter value policy error (Invalid code point - U+e4)
S:   </oxrs:xcp>
S: </value>
S: </result>
S: <trID>
S:   <clTRID>CLI-1102528331230</clTRID>
S:   <svTRID>SRW-14006</svTRID>
S: </trID>
S: </response>
S:</epp>

```

For an EPP domain update request to change an IDN script and the information other than an IDN script, the error response will return with the error code 2306 (“Parameter value policy error”). A detailed error message is also returned in the response.

Example of such <update> command and its response:

```

C:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'>
C: <command>
C:   <update>
C:     <domain:update xmlns:domain='urn:ietf:params:xml:ns:domain-1.0'>
C:       <domain:name>xn--bc-ua.info</domain:name>
C:       <domain:add>
C:         <domain:ns>
C:           <domain:hostObj>ns1.valid.info</domain:hostObj>
C:         </domain:ns>
C:       </domain:add>
C:       <domain:chg>
C:     </domain:chg>
C:   </domain:update>
C: </update>
C: <clTRID>CLI-1102528331230</clTRID>
C: <extension>
C:   <idn:update xmlns:idn='urn:afiliass:params:xml:ns:idn-1.0'>
C:     <idn:chg>
C:       <idn:script>de</idn:script>
C:     </idn:chg>
C:   </idn:update>

```

	<pre> C: </extension> C: </command> C:</epp> </pre>
	<pre> S:<epp xmlns='urn:ietf:params:xml:ns:epp-1.0'> S: <response> S: <result code='2306'> S: <msg lang='en-US'>Parameter value policy error</msg> S: <value xmlns:oxrs='urn:afilias:params:xml:ns:oxrs-1.1'> S: <oxrs:xcp>2306:Parameter value policy error (General information S:and idn script cannot be updated in one domain update request) S: </oxrs:xcp> S: </value> S: </result> S: <trID> S: <clTRID>CLI-1102528331230</clTRID> S: <svTRID>SRW-14007</svTRID> S: </trID> S: </response> S:</epp> </pre>

Formal Syntax

<pre> <?xml version="1.0" encoding="UTF-8"?> <schema targetNamespace="urn:afilias:params:xml:ns:idn-1.0" xmlns:idn="urn:afilias:params:xml:ns:idn-1.0" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"> <annotation> <documentation> Extensible Provisioning Protocol v1.0 domain name extension schema for Internationalized Domain Names (IDN) processing. </documentation> </annotation> </pre>
--

```

<!--
  Child elements found in EPP commands.
-->
<element name="check" type="idn:cmdType"/>
<element name="info" type="idn:cmdType"/>
<element name="create" type="idn:cmdType"/>
<element name="update" type="idn:updateType"/>

<!--
  Utility Types
-->

<complexType name="cmdType">
  <sequence>
    <element name="script" type="idn:scriptType"
      minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>

<complexType name="updateType">
  <sequence>
    <element name="chg" type="idn:chgType"
      minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>

<simpleType name="scriptType">
  <restriction base="token">
    <minLength value="2"/>
    <maxLength value="17"/>
  </restriction>
</simpleType>

<complexType name="chgType">
  <sequence>
    <element name="script" type="idn:scriptType"
      minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>

```

```

<!--
  Child response elements
-->

<element name="infData" type="idn:infDataType"/>

<!--
  Response elements
-->

<complexType name="infDataType">
  <sequence>
    <element name="script" type="idn:scriptType"
      minOccurs="1" maxOccurs="1"/>
  </sequence>
</complexType>

<!--
  End of schema.
-->

</schema>

```

D. Whois

All registered IDNs will be available both in the Port 43 and Web-based Whois, just as regular ASCII domain registrations are.

Both the Port 43 and Web-based Whois will display the A-Label that has been registered as the domain name, a Unicode HEX representation of the domain name and an HTML representation of the Unicode equivalent. Registrars that want to display the Unicode/UTF-8 equivalent for the IDN domain name can integrate the output from the "Unicode HTML" field into their Whois applications.

Whois example (without contact data):

Domain ID: D434-LROR

Domain Name: XN--VAREPRVE-B5A.TLD

Created On: 20-Dec-2001 19:48:51 UTC

Expiration Date: 15-Jan-2006 19:48:51 UTC

Status: OK

IDN Script: DA

Unicode Hex: U+76;U+61;U+72;U+65;U+70;U+72;U+F8;U+76;U+65;



Example of IDN Display in WHOIS with reserved variants:

```
Domain ID:D137-LRMS
Domain Name:ZN---638A490M.TLD
Created On:01-Oct-2008 19:36:02 UTC
Expiration Date:01-Oct-2010 19:36:02 UTC
Sponsoring Registrar:LibertyRMS Co. (R2-LRMS)
Status:TRANSFER PROHIBITED
Status:INACTIVE
Status:ADDPERIOD
Name Server:
...
IDN Script:zh-tw
Unicode Hex:8499;2d;53e4;.tld
Unicode HTML:&#33945;-&#21476;.tld
Variant reserved by Registry Operator: XN----638A851I.TLD
Variant reserved by Registry Operator: XN----638AZ27F.TLD
Variant reserved by Registry Operator: XN----638A624B.TLD
```

E. DNS Resolution

DNS will resolve IDNs in A-Label format. For example, the successful registration of vareprøve.tld will be registered as xn--vareprve-b5a.tld. Assuming the registration meets the usual criteria for resolution, there will be an entry in the zone for xn--vareprve-b5a.tld.

1. IDNA Compliant Browsers

If an end-user enters the desired IDN in an application that is IDNA-compliant, the application will perform 'To ASCII' on the relevant IDN labels and, if successful, will produce a valid A-Label. This A-Label is sent through the resolver by the usual means, and returns according to the customary resolution rules. Resolution is no different from the resolution of a plain ASCII domain except it includes the 'To ASCII' and 'To Unicode' steps performed by the application. A request for a domain in A-Label format will receive an NXDOMAIN response if the domain does not exist in the Zone File.

Browsers anticipated to provide this functionality are Internet Explorer 7.0, Firefox 0.6, Safari 1.2, Netscape 7.1, Mozilla 1.4, and Opera 7. Subsequent versions of these browsers are also expected to contain this same functionality.

2. Non-IDNA Compliant Browsers

Non-IDNA compliant applications will only be capable of handling IDNs in their A-Label form, as with any ASCII string. If the application is not UTF-8 capable, it will not be IDN capable.

In these circumstances the Internet end-user is encouraged to upgrade to an IDNA-compliant browser.



III. Policies

The following tables reflect the various IDN policies applying to designated languages.

A. Simplified Chinese & Traditional Chinese

Language:	Simplified and Traditional Chinese			
IDN-Tag:	ZH-CN and ZH-TW			
Minimum Length:	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length:	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	Domain label for registration must have at least one non-LDH (Letter, Digit, Hyphen) character			
Special Remarks:	<ul style="list-style-type: none"> • Numbers (0-9) and hyphen (-) are allowed • Mixed Chinese scripts and language tags will not be permitted • ASCII characters (A-Z same as a-z) are not permitted 			
Language Table(s):	<ul style="list-style-type: none"> • The Traditional Chinese and Simplified Chinese language table are compliant to Chinese Domain Name Consortium (CDNC) • (Simplified Chinese) https://www.iana.org/domains/idn-tables/tables/global_zh-cn_1.txt • (Traditional Chinese) https://www.iana.org/domains/idn-tables/tables/global_zh-tw_1.txt 			

B. Bosnian, Bulgargian, Belarusian, Macedonian, Montenegrin, Russian, Serbian, & Ukrainian

Language:	Bosnian, Bulgarian, Belarusian, Macedonian, Montenegrin, Russian, Serbian, Ukrainian			
IDN-Tag:	BS, BG, BE, MK, ME, RU, SR-ME, UK			
Minimum Length:	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length:	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> • Domain label for registration must have at least one non-LDH character • ASCII characters (A-Z same as a-z) are not permitted 			
Special Remarks:	<ul style="list-style-type: none"> • Numbers (0-9) and hyphen (-) are allowed • The Montenegrin language table is listed on IANA repository with the language tag "SR-ME". When provision a .GLOBAL label, the language tag ME" must be used. 			
Language Table(s):	<ul style="list-style-type: none"> • RFC5992 (http://tools.ietf.org/html/rfc5992) covers these languages. • (Bosnian) https://www.iana.org/domains/idn-tables/tables/global_bs_1.txt • (Bulgarian) https://www.iana.org/domains/idn-tables/tables/global_bg_1.txt • (Belarusian) https://www.iana.org/domains/idn-tables/tables/global_be_1.txt • (Macedonian) https://www.iana.org/domains/idn-tables/tables/global_mk_1.txt • (Montenegrin) https://www.iana.org/domains/idn-tables/tables/global_sr-me_1.txt • (Russian) https://www.iana.org/domains/idn-tables/tables/global_ru_1.txt • (Serbian) https://www.iana.org/domains/idn-tables/tables/global_sr-me_1.txt • (Ukrainian) https://www.iana.org/domains/idn-tables/tables/global_uk_1.txt 			

C. Danish, Hungarian, Icelandic, Latvian, Lithuanian & Swedish

Language:	Danish, Hungarian, Icelandic, Latvian, Lithuanian, Swedish			
IDN-Tag:	DA, HU, IS, LV, LT, PL, SV			
Minimum Length:	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length:	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> Domain label for registration must have at least one non-LDH character 			
Special Remarks:	<ul style="list-style-type: none"> None 			
Language Table(s):	<ul style="list-style-type: none"> (Danish) https://www.iana.org/domains/idn-tables/tables/global_da_1.txt (Hungarian) https://www.iana.org/domains/idn-tables/tables/global_hu_1.txt (Icelandic) https://www.iana.org/domains/idn-tables/tables/global_is_1.txt (Lithuanian) https://www.iana.org/domains/idn-tables/tables/global_lt_1.txt (Latvian) https://www.iana.org/domains/idn-tables/tables/global_lv_1.txt (Swedish) https://www.iana.org/domains/idn-tables/tables/global_sv_1.txt 			

D. Korean

Language	Korean			
IDN-Tag:	KO			
Minimum Length:	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length:	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none">• Domain label for registration must have at least one non-LDH character			
Language Table(s):	<ul style="list-style-type: none">• The Korean policy conforms to NIDA (National Internet Development Agency of Korea)• https://www.iana.org/domains/idn-tables/tables/global_ko_1.txt			

E. Spanish

Language	Spanish			
IDN-Tag:	ES			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none">• Domain label for registration must have at least one non-LDH character			
Language Table(s):	<ul style="list-style-type: none">• https://www.iana.org/domains/idn-tables/tables/global_es_1.txt			

F. German

Language	German			
IDN-Tag:	DE			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> Domain label for registration must have at least one non-LDH character 			
Special Remarks:	<ul style="list-style-type: none"> The Eszet Character (ß) is not allowed in registration 			
Language Table(s):	<ul style="list-style-type: none"> https://www.iana.org/domains/idn-tables/tables/global_de_1.txt 			

G. Arabic

Language	Arabic			
IDN-Tag:	AR			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> Domain label for registration must have at least one non-LDH character Zero Width Non Joiner (ZWNJ) U+200C and Zero Width Joiner (ZWJ) U+200D are not allowed in registration 			
Special Remarks:	<ul style="list-style-type: none"> Letters (0-9) and Hyphen (-) are allowed in Arabic label, but not Alphabets (a-z) 			
Language Table(s):	<ul style="list-style-type: none"> The Arabic policy conforms to Saudi Arabia's IDN policy https://www.iana.org/domains/idn-tables/tables/global_ar_1.txt 			

H. Hindi

Language	Hindi			
IDN-Tag:	HI			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> Domain label for registration must have at least one non-LDH character Zero Width Non Joiner (ZWNJ) U+200C and Zero Width Joiner (ZWJ) U+200D are not allowed in registration 			
Special Remarks:	<ul style="list-style-type: none"> Letters (0-9) and Hyphen (-) are allowed in Hindi label, but not Alphabets (a-z) 			
Language Table(s):	<ul style="list-style-type: none"> The Hindi policy conforms to NIXI (National Internet Exchange of India) & C-DAC (Centre for Development of Advanced Computing) standard https://www.iana.org/domains/idn-tables/tables/global_hi_1.txt 			

I. Polish

Language	Polish			
IDN-Tag:	PL			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> Domain label for registration must have at least one non-LDH character 			
Language Table(s):	<ul style="list-style-type: none"> https://www.iana.org/domains/idn-tables/tables/global_pl_1.txt 			

J. Portuguese

Language	Portuguese			
IDN-Tag:	PT			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> Domain label for registration must have at least one non-LDH character 			
Language Table(s):	<ul style="list-style-type: none"> https://www.iana.org/domains/idn-tables/tables/global_pt_1.txt 			

K. Italian

Language	Italian			
IDN-Tag:	IT			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> Domain label for registration must have at least one non-LDH character 			
Language Table(s):	<ul style="list-style-type: none"> https://www.iana.org/domains/idn-tables/tables/global_it_1.txt 			

L. French

Language	French			
IDN-Tag:	FR			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	3
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none"> Domain label for registration must have at least one non-LDH character 			
Special Remarks:	<ul style="list-style-type: none"> None 			
Language Table(s):	<ul style="list-style-type: none"> https://www.iana.org/domains/idn-tables/tables/global_fr_1.txt 			

M. Finnish

Language	Finnish			
IDN-Tag:	FI			
Minimum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	1
Maximum Length	<input checked="" type="checkbox"/> A-Label Length	<input type="checkbox"/> Native Length	Length:	63
Special Contextual Rules:	<ul style="list-style-type: none">• Domain label for registration must have at least one non-LDH character			
Language Table(s):	<ul style="list-style-type: none">• https://www.iana.org/domains/idn-tables/tables/global_fi_1.txt			