

IDNABIS  
Internet-Draft  
Intended status: Standards Track  
Expires: November 26, 2009

P. Resnick, Ed.  
Qualcomm Incorporated  
May 25, 2009

Mapping Characters in IDNA  
draft-ietf-idnabis-mappings-00

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#). This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on November 26, 2009.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal

---

Internet-Draft

IDNA Mapping

May 2009

Provisions Relating to IETF Documents in effect on the date of publication of this document (<http://trustee.ietf.org/license-info>). Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## Abstract

In the original version of the Internationalized Domain Names in Applications (IDNA) protocol, any Unicode code points taken from user input were mapped into a set of Unicode code points that "make sense", which were then encoded and passed to the domain name system (DNS). The current version of IDNA presumes that the input to the protocol comes from a set of "permitted" code points, which it then encodes and passes to the DNS, but does not specify what to do with the result of user input. This document specifies the actions taken by an implementation between user input and passing permitted code points to the new IDNA protocol.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">4</a>
<a href="#">1.1.</a>	Requirements Language . . . . .	<a href="#">4</a>
<a href="#">2.</a>	Architectural Principles . . . . .	<a href="#">4</a>
<a href="#">3.</a>	The General Procedure . . . . .	<a href="#">6</a>
<a href="#">4.</a>	IANA Considerations . . . . .	<a href="#">6</a>
<a href="#">5.</a>	Security Considerations . . . . .	<a href="#">7</a>
<a href="#">Appendix A.</a>	Backwards-compatible Mapping Algorithm . . . . .	<a href="#">7</a>
<a href="#">Appendix B.</a>	Acknowledgements . . . . .	<a href="#">7</a>
<a href="#">6.</a>	Normative References . . . . .	<a href="#">7</a>
	Author's Address . . . . .	<a href="#">8</a>

## 1. Introduction

This document specifies the operations that applications apply to user input in order to get it into a form acceptable by the Internationalized Domain Names in Applications (IDNA) protocol [[I-D.ietf-idnabis-protocol](#)]. The document describes the architectural principles that underly this function in [section 2](#), describes a general procedure that an application SHOULD implement in [section 3](#), and specifies an algorithm and mapping that an application MAY implement in order to remain reasonably backward compatible with the original version of the IDNA protocol in [appendix A](#).

It should be noted that this document is NOT specifying the behavior of a protocol that appears "on the wire". It specifies an operation that is to be applied to user input in order to prepare that user input for use in an "on the network" protocol. As unusual as this may be for an IETF protocol document, it is a necessary operation to maintain interoperability.

### 1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[RFC2119](#)].

## 2. Architectural Principles

An application that implements the IDNA protocol

[[I-D.ietf-idnabis-protocol](#)] must take a set of user input and convert that input to a set of Unicode code points. That user input might be typed on a keyboard, written by hand onto some sort of digitizer, spoken into a microphone and interpreted by a speech-to-text engine, or otherwise. The process of taking any particular user input and mapping it into a Unicode code point may be a simple one: If a user strikes the "A" key on a US English keyboard, without any modifiers such as the "Shift" key held down, in order to draw a Latin small letter A ("a"), many (perhaps most) modern operating system input methods will produce to the calling application the code point U+0061, encoded in a single octet. Sometimes the process is somewhat more complicated: A user might strike a particular set of keys to represent a combining macron followed by striking the "A" key in order to draw a Latin small letter A with a macron above it. Depending on the operating system, the input method chosen by the user, and even the parameters with which the application communicates with the input method, the result might be the code point U+0101 (encoded as two octets in UTF-8 or UTF-16, four octets in UTF-32, etc.), the code point U+0061 followed by the code point U+0304

(again, encoded in three or more octets, depending upon the encoding used) or even the code point U+FF41 followed by the code point U+0304 (and encoded in some form). And these examples leave aside the issue of operating systems and input methods that do not use Unicode code points for their character set. In every case, applications (with the help of the operating systems on which they run and the input methods used) MUST perform a mapping from user input into Unicode code points.

The original version of the IDNA protocol [[RFC3490](#)] used a model whereby input was taken from the user, mapped (via whatever input method mechanisms were used) to a set of Unicode code points, and then further mapped to a set of Unicode code points using the Nameprep profile specified in [[RFC3491](#)]. In this procedure, there are two separate mapping steps: First, a mapping done by the input method (which might be controlled by the operating system, the application, or some combination) and then a second mapping performed by the Nameprep portion of the IDNA protocol. The mapping done in Nameprep includes a particular mapping table to re-map some characters to other characters, a particular normalization, and a set of prohibited characters.

Note that the result of the two step mapping process means that the mapping chosen by the operating system or application in the first step might differ significantly from the mapping supplied by the Nameprep profile in the second step. This has advantages and disadvantages. Of course, the second mapping regularizes what gets looked up in the DNS, making for better interoperability between implementations which use the Nameprep mapping. However, the application or operating system may choose mappings in their input methods, which when passed through the second (Nameprep) mapping result in characters that are "surprising" to the end user.

The other important feature of the original version of the IDNA protocol is that, with very few exceptions, it assumes that any set of Unicode code points provided to the Nameprep mapping can be mapped into a string of Unicode code points that are "sensible", even if that means mapping some code points to nothing (that is, removing the code points from the string). This allowed maximum flexibility in input strings.

The present version of IDNA differs significantly in approach from the original version. First and foremost, it does not provide explicit mapping instructions. Instead, it assumes that the application (perhaps via an operating system input method) will do whatever mapping it requires to convert input into Unicode code points. This has the advantage of giving flexibility to the application to choose a mapping that is suitable for its user given

specific user requirements, and avoids the two-step mapping of the original protocol. Instead of a mapping, the current version of IDNA provides a set of categories that can be used to specify the valid code points allowed in a domain name.

In principle, an application ought to take user input of a domain name and convert it to the set of Unicode code points that represent the domain name the user *\_intends\_*. As a practical matter, of course, determining user desires is a tricky business, so an application needs to choose a reasonable mapping from user input. That may differ based on the particular circumstances of a user, depending on locale, language, type of input method, etc. It is up to the application to make a reasonable choice.

In the next section, this document specifies a general algorithm that

applications SHOULD implement in order produce Unicode code points that will be valid under the IDNA protocol. Then, in [appendix A](#), a full mapping is specified that is substantially compatible with the original IDNA protocol. An application MAY implement the full mapping or MAY choose a different mapping.

### [3.](#) The General Procedure

The general algorithm that an application (or the input method provided by an operating system) should use is relatively straightforward and generally follows section 5 of [\[I-D.ietf-idnabis-protocol\]](#):

1. All characters are mapped using Unicode Normalization Form C (NFC). [\[Unicode51\]](#)
2. Capital (upper case) characters are mapped to their small (lower case) equivalents. [\[\[anchor2: Need reference to "toLowerCase"\]\]](#)
3. Full-width and half-width CJK characters are mapped to their equivalents. [\[\[anchor3: Handwaving for how that's supposed to happen\]\]](#)

These are the minimal mappings that an application SHOULD do. Of course, there are many others that MAY be done. In particular, a mapping that is substantially compatible with [\[RFC3490\]](#) appears below in [appendix A](#).

### [4.](#) IANA Considerations

This memo includes no request to IANA.

Resnick

Expires November 26, 2009

[Page 6]

---

Internet-Draft

IDNA Mapping

May 2009

### [5.](#) Security Considerations

### [Appendix A.](#) Backwards-compatible Mapping Algorithm

The following mapping is mostly backwards-compatible with the original version of the IDNA protocol [\[RFC3490\]](#). One important change is that the original IDNA specification mapped some characters

to nothing that the current IDNA specification permit. Those characters are not re-mapped in this algorithm.

[[anchor4](#): This is filler; needs to be completed.]]

1. Map using table B.1 and B.2 from [[RFC3454](#)].
2. Normalize using Unicode Normalization Form KC. [[Unicode51](#)]
3. Prohibit using tables C.1.2, C.3, C.4, C.5, C.6, C.7, C.8, and C.9 from [[RFC3454](#)].

## [Appendix B](#). Acknowledgements

### [6](#). Normative References

- [I-D.ietf-idnabis-protocol]  
Klensin, J., "Internationalized Domain Names in Applications (IDNA): Protocol",  
[draft-ietf-idnabis-protocol-12](#) (work in progress),  
May 2009.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", [RFC 3454](#), December 2002.
- [RFC3490] Faltstrom, P., Hoffman, P., and A. Costello, "Internationalizing Domain Names in Applications (IDNA)", [RFC 3490](#), March 2003.
- [RFC3491] Hoffman, P. and M. Blanchet, "Nameprep: A Stringprep Profile for Internationalized Domain Names (IDN)", [RFC 3491](#), March 2003.
- [Unicode51]



5.1.0", 2008.

defined by: The Unicode Standard, Version 5.0, Boston, MA,  
Addison-Wesley, 2007, ISBN 0-321-48091-0, as amended by  
Unicode 5.1.0  
(<http://www.unicode.org/versions/Unicode5.1.0/>).

#### Author's Address

Peter W. Resnick (editor)  
Qualcomm Incorporated  
5775 Morehouse Drive  
San Diego, CA 92121-1714  
US

Phone: +1 858 651 4478  
Email: [presnick@qualcomm.com](mailto:presnick@qualcomm.com)  
URI: <http://www.qualcomm.com/~presnick/>