

Internet Draft

Marc Blanchet

[draft-ietf-idn-version-00.txt](#)

Viagenie

October 26, 2000

Expires in six  
months

Handling versions of internationalized domain names protocols

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>.

Abstract

This document describes some issues related to handling changes in the codepoints and other features in the chars space of an internationalized domain name as well as changes in the protocol that supports it (whatever that be). It also presents some solutions to these issues.

## **1. Rationale**

The nameprep draft [[NAMEPREP](#)] is defining prohibited chars, normalization algorithms, etc.. The current consensus is to take the safer approach of not accepting new chars if they are not listed, since the new characters that can be included in the subsequent releases of the universal character set can have an impact on the domain name (for example, a new variation of

the dot . will mostly be non-compatible and being excluded).

The Unicode and ISO-10646 versioning is designed not for the same purpose as the idn work: for example, some chars or properties can be changed or added to those repertoires that cannot be taken as is by the idn protocol. In essence, the idn nameprep versions cannot be in sync with unicode/iso versions. idn need its own versioning of the accepted character set, which can or cannot be synchronized with the two others. While saying that, there is no intent at all to define new codepoints inside this work and any attempt to do that must be rejected.

Since internationalization and specifically internationalization of the domain names is new, we don't really know if the chosen algorithm, protocol and codepoints will not have any problem in the future. We need to handle versions of the protocol and to have a specific table for idn accepted chars.

## **2. Versioning of the protocol**

Since the idn table of chars will be changed in the future, and possibly the algorithms and the processing, then there is a need to handle these situations in a controlled way. A version of the idn protocol should be defined and included as part of the exchange between the parties so that they can handle smoothly the new revisions.

### **2.1 Implementing the version in the protocol**

Depending on which way the idn protocol will be, a different versioning will have to be defined. We will discuss the current proposals and identify where a versioning system can be handled.

#### **2.1.1 Proposals based on extensions of the DNS protocol**

**Proposals based on extensions of the DNS protocol should include in the bits the version number and a way to exchange version numbers between the parties.** [IDNE] already defined a version number as part of the use of EDNS. Similar versioning should be defined in the other proposed protocols.

#### **2.1.2 Proposals based on ACE and application**

**Since ACEs (for example [RACE]) in applications [IDNA] do not change the DNS protocol but only encode the idn in a ascii compatible way, it looks more difficult to include a version number in the ACE encoding, since it will change the domain name.** The proposal is to use a different prefix/suffix for each version, by using one of the chars used in the prefix as a version number, beginning with the lowest possible ascii char available and increase the ascii codepoint of the char by 1 for each version. For example, if the prefix is "ra--", then the first version of idn will be "ra--", the second version will be "rb--", the third "rc--". While this would be more elegant, one can handle versioning by having different prefixes for each version, while choosing prefixes randomly (i.e. 1st version: rq--, 2nd version: wz--, ...). IANA should block all possible prefixes so that no pre-registration is permitted.

## **2.2 Major and minor version numbers**

A <major>.<minor> version number is proposed (i.e. 1.0, 1.1, 2.0). A minor revision is defined to be as new chars or small changes in the table to be handled without any modification of algorithm. The idea here is to enable easy upgrading of idn processors when only new chars will be

handled. In this case, the binary software do not have to be changed and only the table containing the chars has to be changed to enable a new version. A major revision means that the software has to be upgraded since it implies new ways of handling idn which are not identified in the table.

### **2.3 Processing of the version numbers**

Any idn processor MUST verify the version number before processing. When a major version number is higher than what the processor currently handle is detected, processing MUST be stopped and rejected. When a minor version number is higher than what the processor currently handle, then any intermediate processors MUST forward the idn but the end processor (i.e. the dns server authoritative for that zone that is handling the request) MUST stop and reject the request. Specific handling of rejecting the request should be defined in the protocol document.

### **2.4 Version numbers**

**Version numbers of the idn protocol will be handled by IANA.** A IESG-reviewed document should be used as the basis for IANA to define a new protocol version number.

## **3. Idn table**

**Since the unicode consortium and ISO will issue new versions not at the same time as the idn protocol versioning, the IETF need to have its own registered table of accepted idn chars.** This will also enable specific data only useful for idn. The intent is not to duplicate the unicode/iso effort, it is to support the specific needs of the idn group. For example, it is possible that specific chars will have a different behavior than the normal Unicode way: the special casing for final or non-final letters in the Unicode SpecialCasing.txt file can be merged (ie not totally identical to the unicode processing) to only one casing since final and non-final letters have less meaning in a domain name. Simpler processing for idn is also useful: for example, the Lud property is probably not useful in the idn context and can be considered equivalent to Lu. Combining those properties together means much simple table and simpler processing and less errors in implementations. Added chars, codepoint, properties MUST NOT be done in this file, but must be done within the Unicode/ISO process.

This document proposes a table contained in an ascii file handled by IANA and available in the IANA public directory. The source of the table will be the Unicode table, with a similar format, but a simpler, and perhaps different, content based on the needs of the idn protocol. Proposed filename is: idntab.txt.

This file format will also help implementors to have the same input description and exhaustive list of which chars and processing to handle idn. This is easier for implementors to have a complete list of accepted chars instead a list of non-accepted chars. The hope is also to help decrease the different behaviors between the different implementations. This table can be considered as an implementation of [[NAMEPREP](#)].

### **3.1 File format**

The file is divided in two parts. The first part contains variables. The second part contains all the chars accepted for idn. The two parts are separated by a empty line.

The format of the first part is:

```
tag=value<CR><LF>
```

Currently, only the version tag is defined. The "version" tag defines the highest idn version number that can be found in this table. The intent is to have only one file that is kept current but where the beginning of the file can be used by parsers to identify the latest version of the file.

The first idntab.txt file will define version=1.0:

```
version=1.0<CR><LF>
```

The format of the second part is:

```
one codepoint per line  
lines separated by CR/LF  
each field in the line separated by ";"
```

The fields are (in order, from left to right):

1. codepoint in hexadecimal (as in unicode table): HHHH
2. first version of idn table that this char is supported

It is possible that new fields will be added in the future. Parsers should ignore the fields that they don't understand.

Spaces (ascii 0x20) are ignored. Comments are identified by "#". All text following the comment char up to the end of line is ignored. Any codepoint not in the table is considered prohibited. Codepoints that are prohibited may be included in the table inside commented lines in order to help a reader to see explicitly which ones are prohibited.

Example of the file idntab10.txt:

```
version=1.0<CR><LF>  
<CR><LF>  
0041;1.0;
```

#### **4. Security Considerations**

Changing the way a software react about domain names is clearly something that have security impacts. While the actual table doesn't present any security threat by itself, if there is someways by a intruder to reload a new table into a idn processor software without the knowledge of the user, then it can completly disables name resolution or have other possible threats. In conclusion, care must be taken by software developers on how to manage the insertion of a new idn table in a idn processor software.

#### **5. IANA Considerations**

**This document describes versions of the idn file called idntab.txt. The file should be kept in the IANA public directory for references purposes. New versions of the file should be made available after IESG review of a document. Old revisions of the file (idntab-xy.txt) should be kept for documentation purposes.**

#### **6. Acknowledgements**

**The author would like to acknowledge the comments and idea of the following people: Fran ois Yergeau and Paul Hoffman.**

#### **7. Author**

Marc Blanchet  
Viagenie  
**2875** boul. Laurier, bur. 300  
Ste-Foy, Quebec, Canada  
G1V 2M2  
Marc.Blanchet@viagenie.qc.ca  
tel: 418-656-9254

## **8. References**

[NAMEPREP] Preparation of Internationalized Host Names, Paul Hoffman, Marc Blanchet. Work in progress.

[RACE] RACE: Row-based ASCII Compatible Encoding for IDN, Paul Hoffman. Work in progress.

[IDNA] Internationalizing Host Names In Applications (IDNA), Paul Hoffman, Patrick Falstrom. Work in progress.

Marc Blanchet  
Viagenie inc.  
tel: 418-656-9254  
<http://www.viagenie.qc.ca>

-----  
Normos (<http://www.normos.org>): Internet standards portal:  
IETF RFC, drafts, IANA, W3C, ATMForum, ISO, ... all in one place.