IETF IDN Working Group
Internet Draft
23 May 2001

Editors Zita Wenzel, James Seng

draft-ietf-idn-requirements-07.txt
Expires 23 November 2001

Requirements of Internationalized Domain Names

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.

Intended Scope

The intended scope of this document is to explore requirements for the internationalization of domain names on the Internet. It is not intended to document user requirements. It is recommended that solutions not necessarily be within the DNS itself, but could be a layer interjected between the application and the DNS. Proposals SHOULD fulfill most, if not all, of the requirements. This document MAY be updated based on clinical trials.

Abstract

This document describes the requirement for encoding international characters into DNS names and records. This document is guidance for developing protocols for internationalized domain names.

1. Introduction

At present, the encoding of Internet domain names is restricted to a subset of 7-bit ASCII (ISO/IEC 646). HTML, XML, IMAP, FTP, and many other text based items on the Internet have already been at least partially internationalized. It is important for domain names to be similarly internationalized or for an equivalent solution to be found. This document assumes that the most effective solution involves putting

non-ASCII names inside some parts of the overall DNS system although such assumption may not be the consensus of the IETF community.

This document is being discussed on the "idn" mailing list. To join the list, send a message to <majordomo@ops.ietf.org> with the words "subscribe idn" in the body of the message. Archives of the mailing list can also be found at ftp://ops.ietf.org/pub/lists/idn*.

1.1 Definitions and Conventions

A language is a way that humans interact. In computerised form, a text in a written language can be expressed as a string of characters. The same set of characters can often be used for many written languages, and many written languages can be expressed using different scripts. The same characters are often shown with somewhat different glyphs (shapes) for display of a text depending on the font used, the automatic shaping applied, or the automatic formation of ligatures. In addition, the same characters can be shown with somewhat different glyphs (shapes) for display of a text depending on the language being used, even within the same font or through automatic font change.

A character is a member of a set of elements used for organization, control, or representation of textual data.

A graphic character is a character, other than a control function, that has a visual representation normally handwritten, printed, or displayed.

Characters mentioned in this document are identified by their position in the Unicode [UNICODE] character set. This character set is also known as the UCS [ISO10646]. The notation U+12AB, for example, indicates the character at position 12AB (hexadecimal) in the Unicode character set. Note that the use of this notation is not an indication of a requirement to use Unicode.

Examples quoted in this document should be considered as a method to further explain the meanings and principles adopted by the document. It is not a requirement for the protocol to satisfy the examples.

Unicode Technical Report 17 [UTR17] defines a character encoding model in several levels (much of the text below is quoted from Unicode Technical Report 17 [UTR17]):

1. **A abstract character repertoire (ACR) is defined as the set of abstract characters to be encoded, normally a familiar alphabet** or symbol set. The word abstract just means that these objects are defined by convention (such as the 26 letters of the English alphabet, uppercase and lowercase forms). Examples: the ASCII repertoire, the Latin-15 repertoire, the JIS X 0208 repertoire, the UCS repertiore (of a particular version).

2. **A coded character set (CCS) is defined to be a mapping from a**

set of abstract characters to the set of non-negative integers. This range of integers need not be contiguous. An abstract character is defined to be in a coded character set if the coded character set maps from it to an integer. That integer is said to be the code point for the abstract character. That abstract character is then an encoded character. Examples: ASCII, Latin-15, JIS X 0208, the UCS.

3. **A character encoding form (CEF) is a mapping from the set of integers used in a CCS to the set of sequences of code units. A code unit** is an integer occupying a specified binary width in a computer architecture, such as a septet, an octet, or a 16-bit unit. The encoding form enables character representation as actual data in a computer. The sequences of code units do not necessarily have the same length. Examples: ASCII, Latin-15, Shift-JIS, UTF-16, UTF-8.

4. **A character encoding scheme (CES) is a mapping of code units into serialized octet sequences. Character encoding schemes are relevant** to the issue of cross-platform persistent data involving code units wider than a byte, where byte-swapping may be required to put data into the byte polarity canonical for a particular platform.

   The CES may involve two or more CCS's, and may include code units (e.g. single shifts, SI/SO, or escape sequences) that are not part of the CCS per se, but which are defined by the character encoding architecture and which may require an external registry of particular values (as for the ISO 2022 escape sequences). In such a case, the CES is called a compound CES. (A CES that only involves a single CCS is called a simple CES.)

   Examples: ASCII, Latin-15, Shift-JIS, UTF-16BE, UTF-16LE, UTF-8.

5. **The mapping from an abstract character repertoire (ACR) to a serialised sequence of octets is called a Character Map (CM). A simple** character map thus implicitly includes a CCS, a CEF, and a CES, mapping from abstract characters to code units to octets. A compound character map includes a compound CES, and thus includes more than one CCS and CEF. In that case, the abstract character repertoire for the character map is the union of the repertoires covered by the coded character sets involved.

   Character Maps are the things that in the IAB architecture get IANA charset identifiers. A sequence of encoded characters must be unambiguously mapped onto a sequence of octets by the charset. The charset must be specified in all instances, as in Internet protocols, where textual content is treated as a ordered sequence of octets, and where the textual content must be reconstructible from that sequence of octets. Charset names are registered by the IANA according to procedures documented in [RFC2278]. In many cases, the same name is used for both a character map and for a character encoding scheme, such as UTF-16BE. Typically this is done for simple

character maps when such usage is clear from context.

6. **A transfer encoding syntax (TES) is a reversible transform of encoded data which may (or may not) include textual data represented in** one or more character encoding schemes. Examples: 8bit, Quoted-Printable, BASE64, UTF-7 (defunct), (UTF-5, and RACE).

1.2 Description of the Domain Name System

The Domain Name System is defined by [RFC1034] and [RFC1035], with clarifications, extensions and modifications given in [RFC1123], [RFC1996], [RFC2181], and others. Of special importance here is the security extensions described in [RFC2535] and companions.

Over the years, many different words have been used to describe the components of resource naming on the Internet (e.g., URI, URN); to make certain that the set of terms used in this document are well-defined and non-ambiguous, the definitions are given here.

A master server for a zone holds the main copy of that zone. This copy is sometimes stored in a zone file. A slave server for a zone holds a complete copy of the records for that zone. Slave servers MAY be either authorized by the zone owner (secondary servers) or unauthorized (so-called "stealth secondaries"). Master and authorized slave servers are listed in the NS records for the zone, and are termed "authoritative" servers. In many contexts outside this document, the term "primary" is used interchangeably with "master" and "secondary" is used interchangeably with "slave".

A caching server holds temporary copies of DNS records; it uses records to answer queries about domain names. Further explanation of these terms can be found in [RFC1034] and [RFC1996].

DNS names can be represented in multiple forms, with different properties for internationalization. The most important ones are:

- Domain name: The binary representation of a name used internally in the DNS protocol. This consists of a series of components of 1-63 octets, with an overall length limited to 255 octets (including the length fields).

- Master file format domain name: This is a representation of the name as a sequence of characters in some character sets; the common convention (derived from [RFC1035] section 5.1) is to represent the octets of the name as ASCII characters where the octet is in the set corresponding to the ASCII values for [a-zA-Z0-9-], using an escape mechanism (\x or \NNN) where not, and separating the components of the name by the dot character (".").

The form specified for most protocols using the DNS is a limited form of the master file format domain name. This limited form is defined in [RFC1034] Section 3.5 and [RFC1123]. In most implementations of

applications today, domain names in the Internet have been limited to the much more restricted forms used, e.g., in email. Those names are limited to the upper- and lower-case letters a-z (interpreted in a case-independent fashion), the digits, and the hyphen-minus, all in ASCII.

1.3 Definition of "hostname" and "Internationalized Domain Name"

In the DNS protocols, a name is referred to as a sequence of octets. However, when discussing requirements for internationalized domain names, what we are looking for are ways to represent characters that are meaningful for humans.

In this document, this is referred to as a "hostname". While this term has been used for many different purposes over the years, it is used here in the sense of sequence of characters (not octets) representing a domain name conforming to the limited hostname syntax [RFC952].

This document attempts to define the requirements for an "Internationalized Domain Name" (IDN). This is defined as a sequence of characters that can be used in the context of functions where a hostname is used today, but contains one or more characters that are outside the set of characters specified as legal characters for host names [RFC1123].

1.4 A multilayer model of the DNS function

The DNS can be seen as a multilayer function:

- The bottom layer is where the packets are passed across the Internet in a DNS query and a DNS response. At this level, what matters is the format and meaning of bits and octets in a DNS packet.

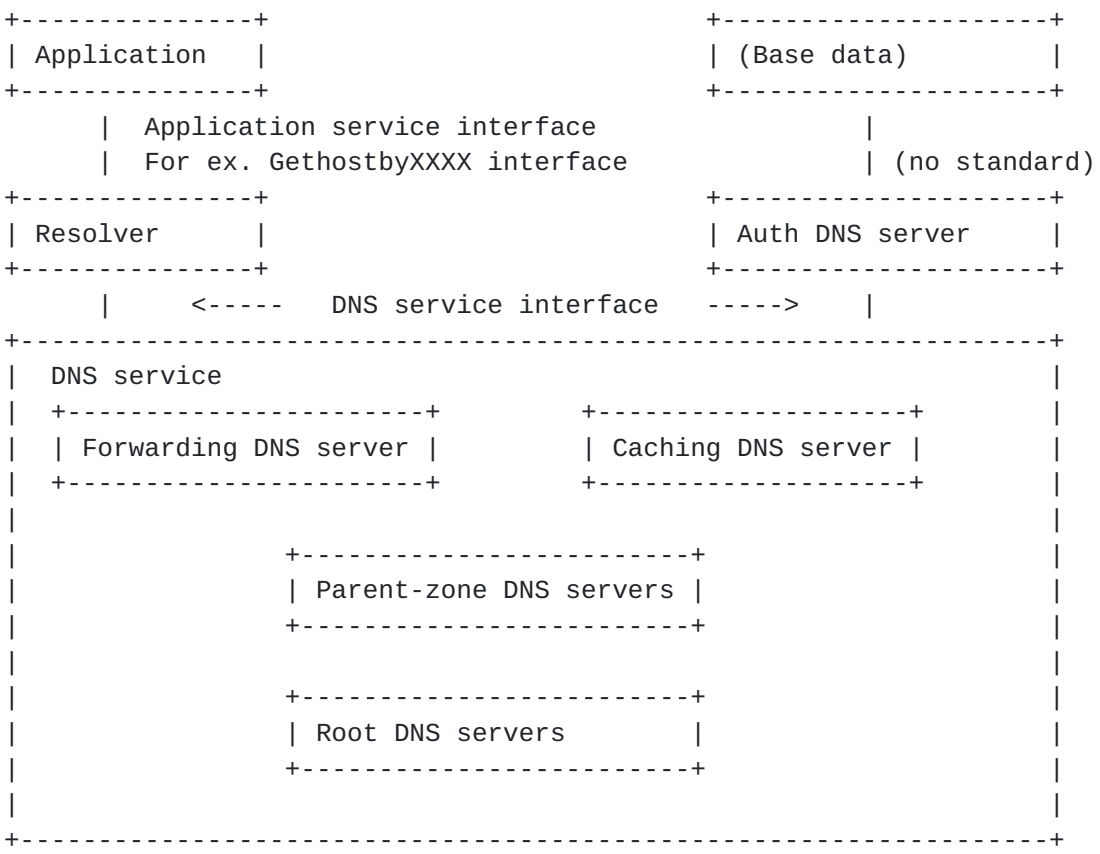
- Above that is the "DNS service", created by an infrastructure of DNS servers, NS records that point to those DNS servers, that is pointed to by the root servers (listed in the "root cache file" on each DNS server often called "named.cache"). It is at this level that the statement "the DNS has a single root" [RFC2826] makes sense, but still, what are being transferred are octets, not characters.

- Interfacing to the user is a service layer, often called "the resolver library", and often embedded in the operating system or system libraries of the client machines. It is at the top of this layer that the API calls commonly known as "gethostbyname" and "gethostbyaddress" reside. These calls are modified to support IPv6 [RFC2553]. A conceptually similar layer exists in authoritative DNS servers, comprising the parts that generate "meaningful" strings in DNS files. Due to the popularity of the "master file" format, this layer often exists only in the administrative routines of the service maintainers.

- The user of this layer (resolver library) is the application programs

that use the DNS, such as mailers, mail servers, Web clients, Web servers, Web caches, IRC clients, FTP clients, distributed file systems, distributed databases, and almost all other applications on TCP/IP.

Graphically, one can illustrate it like this:



1.5 Service model of the DNS

The Domain Name Service is used for multiple purposes, each of which is characterized by what it puts into the system (the query) and what it expects as a result (the reply).

The most used ones in the current DNS are:

- Hostname-to-address service (A, AAAA, A6): Enter a hostname, and get back an IPv4 or IPv6 address.

- Hostname-to-Mail server service (MX): As above, but the expected return value is a hostname and a priority for SMTP servers.

- Address-to-hostname service (PTR): Enter an IPv4 or IPv6 address (in in-addr.arpa or ip6.arpa form respectively) and get back a hostname.

- Domain delegation service (NS). Enter a domain name and get back nameserver records (designated hosts which provide authoritive

nameservice) for the domain.

New services are being defined, either as entirely new services (IPv6 to hostname mapping using binary labels) or as embellishments to other services (DNSSEC returning information about whether a given DNS service is performed securely or not).

These services exist, conceptually, at the Application/Resolver interface, NOT at the DNS-service interface. This document attempts to set requirements for an equivalent of the "used services" given above, where "hostname" is replaced by "Internationalized Domain Name". This doesn't preclude the fact that IDN should work with any kind of DNS queries. IDN is a new service. Since existing protocols like SMTP or HTTP use the old service, it is a matter of great concern how the new and old services work together, and how other protocols can take advantage of the new service.

2. General Requirements

These requirements address two concerns: The service offered to the users (the application service), and the protocol extensions, if needed, added to support this service.

In the requirements, we attempt to use the term "service" whenever a requirement concerns the service, and "protocol" whenever a requirement is believed to constrain the possible implementation.

2.1 Compatibility and Interoperability

[1] The DNS is essential to the entire Internet. Therefore, the service MUST NOT damage present DNS protocol interoperability. It MUST make the minimum number of changes to existing protocols on all layers of the stack. It MUST continue to allow any system anywhere that implements the IDN specification to resolve any internationalized domain name.

[2] The service MUST preserve the basic concept and facilities of domain names as described in [RFC1034]. It MUST maintain a single, global, universal, and consistent hierarchical namespace.

[3] The DNS protocol (the packet formats that go on the wire) MUST NOT limit the codepoints that can be used. A service defined on top of the DNS, for instance the IDN-to-address function, MAY limit the codepoints that can be used. The service descriptions MUST describe what limitations are imposed.

[4] The protocol MUST work for all features of DNS, IPv4, and IPv6. The protocol MUST NOT allow an IDN to be returned to a requestor that requests the IP-to-(old)-domain-name mapping service.

[5] The same name resolution request MUST generate the same response, regardless of the location or localization settings in the resolver, in the master server, and in any slave servers involved in the resolution

process.

[6] The protocol MUST NOT require that the current DNS cache servers be modified to support IDN. If a cache server can have additional functionality to support IDN better, this additional functionality MUST NOT cause problems for resolving correctly functioning current domain names.

[7] A caching server MUST NOT return data in response to a query that would not have been returned if the same query had been presented to an authoritative server. This applies fully for the cases when:

- The caching server does not know about IDN
- The caching server implements the whole specification
- The caching server implements a valid subset of the specification

[8] The service MAY modify the DNS protocol [RFC1035] and other related work undertaken by the [DNSEXT] WG. However, these changes SHOULD be as small as possible and any changes SHOULD be coordinated with the [DNSEXT] WG.

[9] The protocol supporting the service SHOULD be as simple as possible from the user's perspective. Ideally, users SHOULD NOT realize that IDN was added on to the existing DNS.

[10] The best solution is one that maintains maximum feasible compatibility with current DNS standards as long as it meets the other requirements in this document.

[11] The protocol should handle with care new revisions of the CCS. Undefined codepoints should not be allowed unless a new revision of the protocol can handle it. Protocol revisions should be tagged.

2.2 Internationalization

[12] Internationalized characters MUST be allowed to be represented and used in DNS names and records. The protocol MUST specify what charset is used when resolving domain names and how characters are encoded in DNS records.

[13] Codepoints SHOULD be from the Universal Set as defined in ISO-10646 or Unicode. The specifics of versions MUST be defined in the proposed solution. If multiple charsets are allowed, each charset MUST be tagged and conform to [RFC2277].

[14] The protocol MUST NOT reject any non-IDN characters (to be defined) in any DNS queries or responses.

[15] The protocol SHOULD NOT invent a new CCS for the purpose of IDN only and SHOULD use existing CES. The charset(s) chosen SHOULD also be non-ambiguous.

[16] The protocol SHOULD NOT make any assumptions about the location in a domain name where internationalization might appear. In other words, it SHOULD NOT differentiate between any part of a domain name because this MAY impose restrictions on future internationalization efforts. For example, the TLDs can be internationalized.

[17] The protocol also SHOULD NOT make any localized restrictions in the protocol. For example, an IDN implementation which only allows domain names to use a single local script would immediately restrict multinational organization.

[18] While there are a wide range of devices that use the DNS and a wide range of characteristics of international scripts and methods of domain name input and display, IDN is only concerned with the protocol. Therefore, there MUST be a single way of encoding an internationalized domain name within the DNS.

2.3 Canonicalization

Matching rules are a complicated process for IDN. Canonicalization of characters MUST follow precise and predictable rules to ensure consistency. [CHARREQ] is RECOMMENDED as a guide on canonicalization.

The DNS has to match a host name in a request with a host name held in one or more zones. It also needs to sort names into order. It is expected that some sort of canonicalization algorithm will be used as the first step of this process. This section discusses some of the properties which will be REQUIRED of that algorithm.

[19] To achieve interoperability, canonicalization MUST be done at a single well-defined place in the DNS resolution process. The protocol MUST specify canonicalization; it MUST specify exactly where in the DNS that canonicalization happens and does not happen; it MUST specify how additions to ISO 10646 will affect the stability of the DNS and the amount of work done on the root DNS servers.

[20] The canonicalization algorithm MAY specify operations for case, ligature, and punctuation folding.

[21] In order to retain backwards compatibility with the current DNS, the service MUST retain the case-insensitive comparison for [US-ASCII] as specified in [RFC1035]. For example, Latin capital letter A (U+0041) MUST match Latin small letter a (U+0061). [UTR21] describes some of the issues with case mapping. Case-insensitivity for non [US-ASCII] MUST be discussed in the protocol proposal.

[22] Case folding MUST be locale independent. If it were locale-dependent, then different clients would get different results. For example, Latin capital letter I (U+0049) case folded to lower case in the Turkish context will become Latin small letter dotless i (U+0131). But in the English context, it will become Latin small

letter i (U+0069).

[23] If other canonicalization is done, it MUST be done before the domain name is resolved. Further, the canonicalization MUST be easily upgradable as new languages and writing systems are added.

[24] Any conversion (case, ligature folding, punctuation folding, etc) from what the user enters into a client to what the client asks for resolution MUST be done identically on any request from any client.

[25] If the charset can be normalized, then it SHOULD be normalized before it is used in IDN. Normalization SHOULD follow [UTR15].

[26] The protocol SHOULD avoid inventing a new normalization form provided a technically sufficient one is available.

2.4 Operational Issues

[27] Zone files SHOULD remain easily editable.

[28] An IDN-capable resolver or server SHALL NOT generate more traffic than a non-IDN-capable resolver or server would when resolving an ASCII-only domain name. The amount of traffic generated when resolving an IDN SHALL be similar to that generated when resolving an ASCII-only name.

[29] The service SHOULD NOT add new centralized administration for the DNS. A domain administrator SHOULD be able to create internationalized names as easily as adding current domain names.

[30] The protocol MUST work with DNSSEC. The protocol MAY break language sort order.

3. Security Considerations

Any solution that meets the requirements in this document MUST NOT be less secure than the current DNS. Specifically, the mapping of internationalized host names to and from IP addresses MUST have the same characteristics as the mapping of today's host names.

Specifying requirements for internationalized domain names does not itself raise any new security issues. However, any change to the DNS MAY affect the security of any protocol that relies on the DNS or on DNS names. A thorough evaluation of those protocols for security concerns will be needed when they are developed. In particular, IDNs MUST be compatible with DNSSEC and, if multiple charsets or representation forms are permitted, the implications of this name-spoof MUST be throughly understood.

4. References

[CHARREQ] "Requirements for string identity matching and String

Indexing", http://www.w3.org/TR/WD-charreq, July 1998, World Wide Web Consortium.

[DNSEXT] "IETF DNS Extensions Working Group", namedroppers@ops.ietf.org, Olafur Gudmundson, Randy Bush.

[RFC952] "DoD Internet Host Table Specification", rfc952.txt, October 1985, K. Harrenstien, M.K. Stahl, E.J. Feinler.

[RFC1034] "Domain Names - Concepts and Facilities", rfc1034.txt, November 1987, P. Mockapetris.

[RFC1035] "Domain Names - Implementation and Specification", rfc1035.txt, November 1987, P. Mockapetris.

[RFC1123] "Requirements for Internet Hosts -- Application and Support", rfc1123.txt, October 1989, R. Braden.

[RFC1996] "A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY)", rfc1996.txt, August 1996, P. Vixie.

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", rfc2119.txt, March 1997, S. Bradner.

[RFC2181] "Clarifications to the DNS Specification", rfc2181.txt, July 1997, R. Elz, R. Bush.

[RFC2277] "IETF Policy on Character Sets and Languages", rfc2277.txt, January 1998, H. Alvestrand.

[RFC2278] "IANA Charset Registration Procedures", rfc2278.txt, January 1998, N. Freed and J. Postel.

[RFC2279] "UTF-8, a transformation format of ISO 10646", rfc2279.txt, F. Yergeau, January 1998.

[RFC2535] "Domain Name System Security Extensions", rfc2535.txt, March 1999, D. Eastlake.

[RFC2553] "Basic Socket Interface Extensions for IPv6", rfc2553.txt, March 1999, R. Gilligan et al.

[RFC2825] "A Tangled Web: Issues of I18N, Domain Names, and the Other Internet protocols", rfc2825.txt, May 2000, L. Daigle et al.

[RFC2826] "IAB Technical Comment on the Unique DNS Root", rfc2826.txt, May 2000, Internet Architecture Board.

[IDNCOMP] "Comparison of Internationalized Domain Name Proposals", draft-ietf-idn-compare-00.txt, June 2000, P. Hoffman.

[ISO10646] ISO/IEC 10646-1:2000 (note that an amendment 1 is in preparation), ISO/IEC 10646-2 (in preparation), plus corrigenda and amendments to these standards.

[UNICODE] The Unicode Consortium, "The Unicode Standard". Described at http://www.unicode.org/unicode/standard/versions/.

[UNICODE30] The Unicode Consortium, "The Unicode Standard -- Version 3.0", ISBN 0-201-61633-5. Same repertoire as ISO/IEC 10646-1:2000. Described at http://www.unicode.org/unicode/standard/versions/Unicode3.0.html.

[US-ASCII] Coded Character Set -- 7-bit American Standard Code for Information Interchange, ANSI X3.4-1986; also: ISO/IEC 646 (IRV).

[UAX15] "Unicode Normalization Forms", Unicode Standard Annex #15, http://www.unicode.org/unicode/reports/tr15/, 2000-08-31, M. Davis and M. Duerst, Unicode Consortium.

[UTR17] "Character Encoding Model", Unicode Technical Report #17, http://www.unicode.org/unicode/reports/tr17/, 2000-08-31, K. Whistler and M. Davis, Unicode Consortium.

[UTR21] "Case Mappings", Unicode Technical Report #21, http://www.unicode.org/unicode/reports/tr21/, 2000-09-12, M. Davis, Unicode Consortium.

5. Editors' Contact

Zita Wenzel, Ph.D.
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA
90292 USA
Tel: +1 310 448 8462
Fax: +1 310 823 6714
zita@isi.edu

James Seng
i-DNS.net International Pte Ltd.
8 Temesek Boulevand
#24-02 Suntec Tower 3
Singapore 038988
Tel: +65 248 6208
Fax: +65 248 6198
Email: jseng@pobox.org.sg

6. Acknowledgements

The editors gratefully acknowledge the contributions of:

Harald Tveit Alvestrand <Harald@Alvestrand.no>
Mark Andrews <Mark.Andrews@nominum.com>
RJ Atkinson <request not to have email>
Alan Barret <apb@cequrux.com>
Marc Blanchet <blanchet@mailviagenie.qc.ca>
Randy Bush <randy@psg.com>
Andrew Draper <ADRAPER@altera.com>
Martin Duerst <duerst@w3.org>
Patrik Faltstrom <paf@swip.net>
Ned Freed <ned.freed@innosoft.com>
Olafur Gudmundsson <ogud@tislabs.com>
Paul Hoffman <phoffman@imc.org>
Simon Josefsson <jas+idn@pdc.kth.se>
Kent Karlsson <keka@im.se>
John Klensin <klensin+idn@jck.com>
Tan Juay Kwang <tanjk@i-dns.net>
Dongman Lee <dlee@icu.ac.kr>
Bill Manning <bmanning@ISI.EDU>
Dan Oscarsson <Dan.Oscarsson@trab.se>
J. William Semich <bill@mail.nic.nu>
Yoshiro Yoneda <yone@nic.ad.jp>