Network Working Group
Internet-Draft
Intended Status: Proposed Standard

Chris Newman
Sun Microsystems
Arnt Gulbrandsen
Oryx Mail Systems GmhH
August 2007

Internet Message Access Protocol Internationalization
draft-ietf-imapext-i18n-12.txt

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

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.

This Internet-Draft expires in February 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

Internet Message Access Protocol (IMAP) version 4rev1 has basic support for non-ASCII characters in mailbox names and search substrings. It also supports non-ASCII message headers and content encoded as specified by Multipurpose Internet Mail Extensions (MIME). This specification defines a collection of IMAP extensions which improve international support including comparator negotiation for search, sort and thread, language negotiation for international

error text, and translations for namespace prefixes.

Table of Contents

1. Conventions Used in this Document 2
2. Introduction 3
3. LANGUAGE Extension 3
3.1 LANGUAGE Extension Requirements 3
3.2 LANGUAGE Command 4
3.3 LANGUAGE Response 6
3.4 TRANSLATION Extension to the NAMESPACE Response 6
3.5 Formal Syntax 6
4. COMPARATOR Extension 7
4.1 COMPARATOR Extension Requirements 8
4.2 Comparators and Charsets 9
4.3 COMPARATOR Command 9
4.4 COMPARATOR Response 10
4.5 Formal Syntax 10
5. Other IMAP Internationalization Issues 11
5.1 UTF-8 Userids and Passwords 11
5.2 UTF-8 Mailbox Names 11
5.3 UTF-8 Domains, Addresses and Mail Headers 11
6. IANA Considerations 12
7. Security Considerations 12
8. Acknowledgements 12
9. Relevant Standards for i18n IMAP Implementations 13
Normative References 13
Informative References 14
Authors' Addresses 15
Intellectual Property and Copyright Statements 16

Conventions Used in This Document

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 [RFC2119].

The formal syntax use the Augmented Backus-Naur Form (ABNF) [RFC4234] notation including the core rules defined in Appendix A. The UTF8-related productions are defined in [RFC3629].

In examples, "C:" and "S:" indicate lines sent by the client and server respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

2. Introduction

This specification defines two IMAP4rev1 [RFC3501] extensions to enhance international support. These extensions can be advertised and implemented separately.

The LANGUAGE extension allows the client to request a suitable language for protocol error messages and in combination with the NAMESPACE extension [RFC2342] enables namespace translations.

The COMPARATOR extension allows the client to request a suitable collation which will modify the behavior of the base specification's SEARCH command as well as the SORT and THREAD extensions [SORT]. This leverages the collation registry [RFC4790].

3. LANGUAGE Extension

IMAP allows server responses to include human-readable text that in many cases needs to be presented to the user. But that text is limited to US-ASCII by the IMAP specification [RFC3501] in order to preserve backwards compatibility with deployed IMAP implementations. This section specifies a way for an IMAP client to negotiate which language the server should use when sending human-readable text.

The LANGUAGE extension only provides a mechanism for altering fixed server strings such as response text and NAMESPACE folder names. Assigning localized language aliases to shared mailboxes would be done with a separate mechanism such as the proposed METADATA extension (see [METADATA]).

3.1 LANGUAGE Extension Requirements

IMAP servers that support this extension MUST list the keyword LANGUAGE in their CAPABILITY response as well as in the greeting CAPABILITY data.

A server that advertises this extension MUST use the language "i-default" as described in [RFC2277] as its default language until another supported language is negotiated by the client. A server MUST include "i-default" as one of its supported languages.

Clients and servers that support this extension MUST also support the NAMESPACE extension [RFC2342].

The LANGUAGE command is valid in all states. Clients are urged to issue LANGUAGE before authentication, since some servers send

valuable user information as part of authentication (e.g. "password is correct, but expired").

3.2 LANGUAGE Command

Arguments: Optional language range arguments.

Response: A possible LANGUAGE response (see section 3.3).
A possible NAMESPACE response (see section 3.4).

Result: OK - Command completed
NO - Could not complete command
BAD - arguments invalid

The LANGUAGE command requests that human-readable text emitted by the server be localized to a language matching one of the language range argument as described by section 2 of [RFC4647].

If the command succeeds, the server will return human-readable responses in the first supported language specified. These responses will be in UTF-8 [RFC3629]. The server MUST send a LANGUAGE response specifying the language used, and the change takes effect immediately after the LANGUAGE response.

If the command fails, the server continues to return human-readable responses in the language it was previously using.

The special "default" language range argument indicates a request to use a language designated as preferred by the server administrator. The preferred language MAY vary based on the currently active user.

If a language range does not match a known language tag exactly but does match a language by the rules of [RFC4647], the server MUST send an untagged LANGUAGE response indicating the language selected.

If there aren't any arguments, the server SHOULD send an untagged LANGUAGE response listing the languages it supports. If the server is unable to enumerate the list of languages it supports it MAY return a tagged NO response to the enumeration request.

```
< The server defaults to using English i-default responses until
  the user explicitly changes the language. >

C: A001 LOGIN KAREN PASSWORD
S: A001 OK LOGIN completed

< Client requested MUL language, which no server supports. >
```

```
C: A002 LANGUAGE MUL
S: A002 NO Unsupported language MUL

< A LANGUAGE command with no arguments is a request to enumerate
  the list of languages the server supports. >

C: A003 LANGUAGE
S: * LANGUAGE (EN DE IT i-default)
S: A003 OK Supported languages have been enumerated

C: B001 LANGUAGE
S: B001 NO Server is unable to enumerate supported languages

< Once the client changes the language, all responses will be in
  that language starting after the LANGUAGE response. Note that
  this includes the NAMESPACE response. Because RFCs are in US-
  ASCII, this document uses an ASCII transcription rather than
  UTF-8 text, e.g. ue in the word "ausgefuehrt" >

C: C001 LANGUAGE DE
S: * LANGUAGE (DE)
S: * NAMESPACE (("" "/")) (("Other Users/" "/" "TRANSLATION"
      ("Andere Ben&APw-tzer/"))) (("Public Folders/" "/"
      "TRANSLATION" ("Gemeinsame Mailboxen/")))
S: C001 OK Sprachwechsel durch LANGUAGE-Befehl ausgefuehrt

< If a server does not support the requested primary language,
  responses will continue to be returned in the current language
  the server is using. >

C: D001 LANGUAGE FR
S: D001 NO Diese Sprache ist nicht unterstuetzt
C: D002 LANGUAGE DE-IT
S: * LANGUAGE (DE-IT)
S: * NAMESPACE (("" "/"))(("Other Users/" "/" "TRANSLATION"
      ("Andere Ben&APw-tzer/"))) (("Public Folders/" "/"
      "TRANSLATION" ("Gemeinsame Mailboxen/")))
S: D002 OK Sprachwechsel durch LANGUAGE-Befehl ausgefuehrt
C: D003 LANGUAGE "default"
S: * LANGUAGE (DE)
S: D003 OK Sprachwechsel durch LANGUAGE-Befehl ausgefuehrt

< Server does not speak French, but does speak English. User
  speaks Canadian French and Canadian English. >

C: E001 LANGUAGE FR-CA EN-CA
S: * LANGUAGE (EN)
S: E001 OK Now speaking English
```

3.3 LANGUAGE Response

Contents: A list of one or more language tags.

The LANGUAGE response occurs as a result of a LANGUAGE command. A LANGUAGE response with a list containing a single language tag indicates that the server is now using that language. A LANGUAGE response with a list containing multiple language tags indicates the server is communicating a list of available languages to the client, and no change in the active language has been made.

3.4 TRANSLATION Extension to the NAMESPACE Response

If localized representations of the namespace prefixes are available in the selected language, the server SHOULD include these in the TRANSLATION extension to the NAMESPACE response.

The TRANSLATION extension to the NAMESPACE response returns a single string, containing the modified UTF-7 [RFC3501] encoded translation of the namespace prefix. It is the responsibility of the client to convert between the namespace prefix and the translation of the namespace prefix when presenting mailbox names to the user.

In this example a server supports the IMAP4 NAMESPACE command. It uses no prefix to the user's Personal Namespace, a prefix of "Other Users" to its Other Users' Namespace and a prefix of "Public Folders" to its only Shared Namespace. Since a client will often display these prefixes to the user, the server includes a translation of them that can be presented to the user.

```
C: A001 LANGUAGE DE-IT
S: * NAMESPACE (("" "/")) (("Other Users/" "/" "TRANSLATION"
      ("Andere Ben&APw-tzer/"))) (("Public Folders/" "/"
      "TRANSLATION" ("Gemeinsame Mailboxen/")))
S: A001 OK LANGUAGE-Befehl ausgefuehrt
```

3.5 Formal Syntax

The following syntax specification inherits ABNF [RFC4234] rules from IMAP4rev1 [RFC3501], IMAP4 Namespace [RFC2342], Tags for the Identifying Languages [RFC4646], UTF-8 [RFC3629] and Collected Extensions to IMAP4 ABNF [RFC4466].

```
command-any     =/ language-cmd
    ; LANGUAGE command is valid in all states
```

```
language-cmd    = "LANGUAGE" *(SP lang-range-quoted)

response-payload  =/ language-data

language-data     = "LANGUAGE" SP "(" lang-tag-quoted *(SP
                    lang-tag-quoted) ")"

namespace-trans   = SP DQUOTE "TRANSLATION" DQUOTE SP "(" string ")"
    ; the string is encoded in Modified UTF-7.
    ; this is a subset of the syntax permitted by
    ; the Namespace-Response-Extension rule in [RFC4466]

lang-range-quoted = astring
    ; Once any literal wrapper or quoting is removed, this
    ; follows the language-range rule in [RFC4647]

lang-tag-quoted = astring
    ; Once any literal wrapper or quoting is removed, this follows
    ; the Language-Tag rule in [RFC4646]

resp-text       = ["[" resp-text-code "]" SP ] UTF8-TEXT-CHAR
                  *(UTF8-TEXT-CHAR / "[")
    ; After the server is changed to a language other than
    ; i-default, this resp-text rule replaces the resp-text
    ; rule from [RFC3501].

UTF8-TEXT-CHAR  = %x20-5A / %x5C-7E / UTF8-2 / UTF8-3 / UTF8-4
    ; UTF-8 excluding 7-bit control characters and "["
```

4. COMPARATOR Extension

IMAP4rev1 [RFC3501] includes the SEARCH command which can be used to locate messages matching criteria including human-readable text. The SORT extension [SORT] to IMAP allows the client to ask the server to determine the order of messages based on criteria including human-readable text. These mechanisms require the ability to support non-English search and sort functions.

This section defines an IMAP extension to negotiate use of comparators [RFC4790] to internationalize IMAP SEARCH, SORT and THREAD. The IMAP extension consists of a new command to determine or change the active comparator and a new response to indicate the active comparator and possibly other available comparators.

The term "default comparator" refers to the comparator which is used by SEARCH and SORT absent any negotiation using the COMPARATOR command. The term "active comparator" refers to the comparator

which will be used within a session e.g. by SEARCH and SORT. The COMPARATOR command is used to change the active comparator.

The active comparator applies to the following SEARCH keys: "BCC", "BODY", "CC", "FROM", "SUBJECT", "TEXT", "TO" and "HEADER". If the server also advertises the "SORT" extension, then the active comparator applies to the following SORT keys: "CC", "FROM", "SUBJECT" and "TO". If the server advertises THREAD=ORDEREDSUBJECT, then the active comparator applies to the ORDEREDSUBJECT threading algorithm. If the server advertises THREAD=REFERENCES, then the active comparator applies to the subject field comparisons done by REFERENCES threading algorithm. Future extensions may choose to apply the active comparator to their SEARCH keys.

For SORT and THREAD, the pre-processing necessary to extract the base subject text from a Subject header occurs prior to the application of a comparator.

4.1 COMPARATOR Extension Requirements

IMAP servers that support this extension MUST list the keyword COMPARATOR in their CAPABILITY data once IMAP enters authenticated state, and MAY list that keyword in other states.

A server that advertises this extension MUST implement the i;ascii-casemap and i;octet comparators, as defined in [RFC4790]. A server intended to be deployed globally MUST implement the i;unicode-casemap comparator, as defined in [UCM].

A server that advertises this extension SHOULD use i;ascii-casemap as the default comparator. The selection of the default comparator MAY be adjustable by the server administrator, and MAY be sensitive to the current user. Once the IMAP connection enters authenticated state, the default comparator MUST remain static for the remainder of that connection.

A server that advertises this extension MUST support UTF-8 as a SEARCH charset.

The COMPARATOR command is valid in authenticated and selected states.

Note that since SEARCH uses the substring operation, IMAP servers can only implement collations that offer the substring operation (see [RFC4790 section 4.2.2). Since SORT uses ordering operation (and by implication equality), IMAP servers which advertise the SORT extension can only implement collations that offer all three

operations (see [RFC4790] sections 4.2.2-4).

If the active collation does not provide the operations needed by an IMAP command, the server MUST respond with a tagged BAD.

4.2 Comparators and Character Encodings

When SEARCH, SORT, THREAD or another command needs to perform collation operations on messages (or on the command's arguments), the server MUST remove MIME encoding (see [RFC2047] for headers and [RFC2045] for bodyparts) and convert character encodings compatibly before doing the collation operation.

Strings encoded using unknown character encodings should sort together with invalid input (as defined by the active collation) for the SORT and THREAD commands.

4.3 COMPARATOR Command

Arguments: Optional comparator order arguments.

Response: A possible COMPARATOR response (see Section 4.4).

Result: OK - Command completed
NO - No matching comparator found
BAD - arguments invalid

The COMPARATOR command is used to determine or change the active comparator. When issued with no arguments, it results in a COMPARATOR response indicating the currently active comparator.

When issued with one or more comparator argument, it changes the active comparator as directed. (If more than one installed comparator is matched by an argument, the first argument wins.) The COMPARATOR response lists all matching comparators if more than one matches the specified patterns.

The argument "default" refers to the server's default comparator. Otherwise each argument is an collation specification as defined in the Internet Application Protocol Comparator Registry [RFC4790].

< The client requests activating a Czech comparator if possible, or else a generic international comparator which it considers suitable for Czech. The server picks the first supported comparator. >

```
C: A001 COMPARATOR "cz;*" i;basic
S: * COMPARATOR i;basic
S: A001 OK Will use i;basic for collation
```

4.4 COMPARATOR Response

Contents: The active comparator.
An optional list of available matching comparators

The COMPARATOR response occurs as a result of a COMPARATOR command. The first argument in the comparator response is the name of the active comparator. The second argument is a list of comparators which matched any of the arguments to the COMPARATOR command and is present only if more than one match is found.

4.5 Formal Syntax

The following syntax specification inherits ABNF [RFC4234] rules from IMAP4rev1 [RFC3501], and Internet Application Protocol Comparator Registry [RFC4790].

```
command-auth      =/ comparator-cmd

resp-text-code    =/ "BADCOMPARATOR" / "BADMATCH"

comparator-cmd    = "COMPARATOR" *(SP comp-order-quoted)

response-payload  =/ comparator-data

comparator-data   = "COMPARATOR" SP comp-sel-quoted [SP "("
                    comp-id-quoted *(SP comp-id-quoted) ")"]

comp-id-quoted  = astring
    ; Once any literal wrapper or quoting is removed, this
    ; follows the collation-id rule from [RFC4790]

comp-order-quoted = astring
    ; Once any literal wrapper or quoting is removed, this
    ; follows the collation-order rule from [RFC4790]

comp-sel-quoted   = astring
    ; Once any literal wrapper or quoting is removed, this
    ; follows the collation-selected rule from [RFC4790]
```

5. Other IMAP Internationalization Issues

The following sections provide an overview of various other IMAP internationalization issues. These issues are not resolved by this specification, but could be resolved by other standards work, such as that being done by the EAI group (see [IMAP-EAI]).

5.1 Unicode Userids and Passwords

IMAP4rev1 presently restricts the userid and password fields of the LOGIN command to US-ASCII. The "userid" and "password" fields of the IMAP LOGIN command are restricted to US-ASCII only until a future standards track RFC states otherwise. Servers are encouraged to validate both fields to make sure they conform to the formal syntax of UTF-8 and to reject the LOGIN command if that syntax is violated. Servers MAY reject the use of any 8-bit in the "userid" or "password" field.

When AUTHENTICATE is used, some servers may support userids and passwords in Unicode [RFC3490] since SASL (see [RFC4422]) allows that. However, such userids cannot be used as part of email addresses.

5.2 UTF-8 Mailbox Names

The modified UTF-7 mailbox naming convention described in section 5.1.3 of RFC 3501 is best viewed as an transition from the status quo in 1996 when modified UTF-7 was first specified. At that time, there was widespread unofficial use of local character sets such as ISO-8859-1 and Shift-JIS for non-ASCII mailbox names, with resultant non-interoperability.

The requirements in section 5.1 of RFC 3501 are very important if we're ever going to be able to deploy UTF-8 mailbox names. Servers are encouraged to enforce them.

5.3 UTF-8 Domains, Addresses and Mail Headers

There is now an IETF standard for Internationalizing Domain Names in Applications [RFC3490]. While IMAP clients are free to support this standard, an argument can be made that it would be helpful to simple clients if the IMAP server could perform this conversion (the same argument would apply to MIME header encoding [RFC2047]). However,

it would be unwise to move forward with such work until the work in progress to define the format of international email addresses is complete.

6. IANA Considerations

The IANA is requested to add LANGUAGE and COMPARATOR to the IMAP4 Capabilities Registry. [Note to IANA: http://www.iana.org/assignments/imap4-capabilities]

7. Security Considerations

The LANGUAGE extension makes a new command available in "Not Authenticated" state in IMAP. Some IMAP implementations run with root privilege when the server is in "Not Authenticated" state and do not revoke that privilege until after authentication is complete. Such implementations are particularly vulnerable to buffer overflow security errors at this stage and need to implement parsing of this command with extra care.

A LANGUAGE command issued prior to activation of a security layer is subject to an active attack which suppresses or modifies the negotiation and thus makes STARTTLS or authentication error messages more difficult to interpret. This is not a new attack as the error messages themselves are subject to active attack. Clients MUST re-issue the LANGUAGE command once a security layer is active, so this does not impact subsequent protocol operations.

Both the LANGUAGE and COMPARATOR extensions use the UTF-8 charset, thus the security considerations for UTF-8 [RFC3629] are relevent. However, neither uses UTF-8 for identifiers so the most serious concerns do not apply.

8. Acknowledgements

The LANGUAGE extension is based on a previous Internet draft by Mike Gahrns and Alexey Melnikov, a substantial portion of the text in that section was written by them. Many people have participated in discussions about an IMAP Language extension in the various fora of the IETF and Internet working groups, so any list of contributors is bound to be incomplete. However, the authors would like to thank Andrew McCown for early work on the original proposal, John Myers for suggestions regarding the namespace issue, along with Jutta Degener, Mark Crispin, Mark Pustilnik, Larry Osterman, Cyrus Daboo and Martin Duerst for their many suggestions that have been

incorporated into this document.

Initial discussion of the COMPARATOR extension involved input from Mark Crispin and other participants of the IMAP Extensions WG.

9. Relevant Standards for i18n IMAP Implementations

This is a non-normative list of standards to consider when implementing i18n aware IMAP software.

- The LANGUAGE and COMPARATOR extensions to IMAP (this specification).
- The 8-bit rules for mailbox naming in section 5.1 of RFC 3501.
- The Mailbox International Naming Convention in section 5.1.3 of RFC 3501.
- MIME [RFC2045] for message bodies.
- MIME header encoding [RFC2047] for message headers.
- The IETF EAI working group.
- MIME Parameter Value and Encoded Word Extensions [RFC2231] for filenames. Quality IMAP server implementations will automatically combine multipart parameters when generating the BODYSTRUCTURE. There is also some deployed non-standard use of MIME header encoding inside double-quotes for filenames.
- IDNA [RFC3490] and punycode [RFC3492] for domain names (presently only relevant to IMAP clients).
- The UTF-8 charset [RFC3629].
- The IETF policy on Character Sets and Languages [RFC2277].

Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2277] Alvestrand, "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

[RFC2342] Gahrns, Newman, "IMAP4 Namespace", RFC 2342, May 1998.

[RFC3501] Crispin, "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC3629] Yergeau, "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC4234] Crocker, Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, Brandenburg

Internetworking, Demon Internet Ltd, October 2005.

[RFC4422] Melnikov, Zeilenga, "Simple Authentication and Security Layer (SASL)", RFC 4422, June 2006.

[RFC4466] Melnikov, Daboo, "Collected Extensions to IMAP4 ABNF", RFC 4466, Isode Ltd., April 2006.

[RFC4646] Philips, Davis, "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006.

[RFC4647] Philips, Davis, "Matching of Language Tags", BCP 47, RFC 4647, September 2006.

[RFC4790] Newman, Duerst, Gulbrandsen, "Internet Application Protocol Comparator Registry", RFC 4790, February 2007

[SORT] Crispin, M. and K. Murchison, "INTERNET MESSAGE ACCESS PROTOCOL - SORT AND THREAD EXTENSION", draft-ietf-imapext-sort-18 (work in progress), November 2006.

[UCM] Crispin, "i;unicode-casemap - Simple Unicode Collation Algorithm", draft-crispin-collation-unicasemap-04.txt, May 2007.

Informative References

[RFC2045] Freed, Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2047] Moore, "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

[RFC2231] Freed, Moore, "MIME Parameter Value and Encoded Word Extensions: Character Sets, Languages, and Continuations", RFC 2231, November 1997.

[RFC3490] Faltstrom, Hoffman, Costello, "Internationalizing Domain Names in Applications (IDNA)", RFC 3490, March 2003.

[RFC3492] Costello, "Punycode: A Bootstring encoding of Unicode for Internationalized Domain Names in Applications (IDNA)", RFC 3492, March 2003.

[METADATA] Daboo, C., "IMAP METADATA Extension", draft-daboo-imap-

annotatemore-10 (work in progress), November 2006.

[IMAP-EAI] Resnick, Newman, "IMAP Support for UTF-8", draft-ietf-eai-imap-utf8 (work in progress), May 2006.

Authors' Addresses

Chris Newman
Sun Microsystems
3401 Centrelake Dr., Suite 410
Ontario, CA 91761
US

Email: chris.newman@sun.com

Arnt Gulbrandsen
Oryx Mail Systems GmbH
Schweppermannstr. 8
D-81671 Muenchen
Germany

Email: arnt@oryx.com

Fax: +49 89 4502 9758

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Full Copyright Statement

Copyright (C) The IETF Trust (2007). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.