ENUM Working Group   L. Conroy
Internet Draft   Roke Manor Research
INFORMATIONAL   K. Fujiwara
                                                                  JPRS

Expires: February 2005   July 2004

ENUM Implementation Issues and Experiences
<draft-ietf-enum-experiences-00.txt>

Status of this Memo

Copyright Notice

Abstract

This document captures experience in implementing systems based on the ENUM protocol, and experience of ENUM data that have been created by others. As such, it is informational only, and produced as a help to others in reporting what is "out there" and the potential pitfalls in interpreting the set of documents that specify the protocol.

1. Introduction

The ENUM protocol [1] and the Dynamic Delegation Discovery System (DDDS) [2] [9] [10] [11] are defined elsewhere, and those documents alone form the normative definition of the ENUM system. Unfortunately, this document cannot provide an overview of the specifications, so the reader is assumed to have read and understood the complete set of ENUM normative documents.

>From experience of creating ENUM data and of developing client systems to process that data it is apparent that there are some subtleties in the specifications that have led to different interpretations; in addition there are common syntactic mistakes in data currently "out there" on the Internet.

This document is intended to help others avoid the potential pitfalls in interpreting the set of documents that specify the protocol. It also reports the kind of data they will "find" and so how to process the intent of the publisher of that ENUM data, regardless of the syntax used. As such, it is in keeping with the principle evinced in RFC791 that "In general, an implementation must be conservative in its sending behavior, and liberal in its receiving behavior".

Note that the DDDS system is intricate and so in some places there are several potential interpretations of the specifications. This document proposes a suggested interpretation for some of these points, but they are just that; suggestions.

This draft covers 9 issues in four areas, making 28 recommendations and giving 3 clarifications.

Any ENUM implementation issue has two sides:

- the "Server" side covering the expected behaviour of the ENUM zone population system and expectations that the Registrant may make, and
- the "Client" side covering behaviour that has been observed and that can be expected of the Client, together with the expectations that an end user who requests an ENUM lookup may make.

For each of the issues, we have split the recommendations into "Client" and "Server" proposals. In three cases, we have indicated proposals that relate to ENUMservice specifications, rather than implementations; these are labelled as "Spec".

Note that these recommendations contain the words "MUST", "REQUIRE", "SHOULD", and "MAY". In this particular document, these do not form a standard, and so do not hold their normative definitions. The proposals include these terms from observation of behaviour and for internal consistency, where Client and Server recommendations have to match.

There are undoubtedly other issues, and developers are asked to raise any others they find on the IETF ENUM Working group's mailing list and/or by mail to the authors (see later for contact information).

Note that the authors are not aware of any IPR issues that are involved in the suggestions made in this document.

2. Character Sets and ENUM

2.1. Character Sets - Non-ASCII considered harmful

RFC3761 [1] and RFC3403 [2] specify that ENUM (and NAPTRs) support Unicode using the UTF-8 [5] encoding. This raises an issue where implementations use "single byte" string processing routines. If there are multi-byte characters within an ENUM NAPTR, incorrect processing may well result from these non "UTF-8 aware" systems.

The UTF-8 encoding has a "US-ASCII equivalent range", so that all characters in US-ASCII from 0x00 to 0x7F hexadecimal have an identity map to the UTF-8 encoding; the encodings are the same. In UTF-8, characters with Unicode code points above this range will be encoded using more than one byte, all of which will be in the range 0x80 to 0xFF hexadecimal. Thus it is important to consider the different fields of a NAPTR and whether or not multi-byte characters can or should appear in them.

In addition, characters in the "non-printable" portion of US-ASCII (0x00 to 0x1F hexadecimal, plus 0x7F hexadecimal) are "difficult". Although NAPTRs are processed by machine, they may sometimes need to be written in a "human readable" form. Similarly, if NAPTR content is shown to an end user so that they may choose, it is important that the content is "human readable". Thus it is unwise to use non-printable characters within the US-ASCII range; the Client may have good reason to reject NAPTRs that include these characters as they cannot be shown.

There are two numeric fields in a NAPTR; the ORDER and PREFERENCE fields. As these contain binary values, no risk is involved as string processing should not be applied to them. The "string based" fields are the flags, services, and RegExp fields. The Replacement field holding a domain name encoded according to the standard DNS mechanism [3] [4]. With the introduction of Internationalized Domain Name (IDN) support, this domain name MUST be further encoded using Punycode [6]. As this holds a domain name that is not subject to replacement or modification (other than Punycode processing), it is not of concern here.

Taking the "string" fields in turn, the flags field contains characters that indicate the disposition of the NAPTR. This may be empty, in which case the NAPTR is "non-final", or it may include a flag character as specified in RFC3761. These characters all fall into the US-ASCII equivalent range, so multi-byte characters cannot occur.

The services field includes the DDDS Application identifier ("E2U") used for ENUM, the '+' character used to separate tokens, and a set of ENUMservice identifiers, any of which may include the ':' separator character. In section 2.4.2 of RFC3761 these identifiers are specified as 1*32 ALPHA/DIGIT, so there is no possibility of non-ASCII characters in the services field.

The RegExp field is more complex. It forms a SED-like substitution expression, defined in [2], and consists of two sub-fields:
* the POSIX Extended Regular Expression (ERE) sub-field [7]
* a replacement (repl) sub-field [2].

Additionally, RFC3403 specifies that a flag character may be appended, but the only flag currently defined there (the 'i' case insensitivity flag) is not appropriate for ENUM - see later in this document.

The ERE sub-field matches against the "Application Unique String"; for ENUM, this is defined in RFC3761 to consist of digit characters, with an initial '+' character. All of these fall into the US-ASCII equivalent range of UTF-8 encoding, as do the characters significant to the ERE processing. Thus, for ENUM, there will be no multi-byte characters within this sub-field.

The repl sub-field can include a mixture of explicit text used to construct a URI and characters significant to the substitution expression, as defined in RFC3403. Whilst the latter set all fall into the US-ASCII equivalent range of UTF-8 encoding, this might not be the case for all conceivable text used to construct a URI.

The current URIs use the overall URI character "escaping" rules [8], and so any multi-byte characters will be pre-processed; they will not occur in the explicit text used to construct a URI within the repl sub-field. However, a future URI scheme might be developed that allows characters with multi-byte UTF-8 encoding to exist. If this happens, and if an ENUMservice specification is created that refers to this scheme, then multi-byte characters could then occur in a NAPTR. The presence of these characters could complicate URI generation and processing routines.

Given that this is the only place within an ENUM NAPTR where such multi-byte encodings might reasonably be found, a simple solution is for no ENUMservice definition to include a reference to a URI that allows UTF-8 characters outside the US-ASCII equivalent range to be included, or to require that such characters be "escape encoded" before inclusion in the repl sub-field. Note that the domain part of a URI MUST be processed using Punycode if it has a non-ASCII domain name, so there will be no non-ASCII characters in this part of any URI.

Thus, it is RECOMMENDED that, taking into account the existing client base:

```
Spec   - ENUMservice registrations SHOULD REQUIRE that any static text
         in the repl sub-field is encoded using only characters in the
         US-ASCII equivalent range that are "printable".
         If any of the static text characters do fall outside this
         range then they MUST be pre-processed using a URI-specific
         "escape" mechanism to re-encode them only using US-ASCII
         equivalent printable characters ( in the range 0x20 to 0x7E).
```

At the least, it is RECOMMENDED that:

```
Spec   - Any ENUMservice registration that allows characters requiring
         multi-byte UTF-8 encoding to be present in the repl sub-field
         MUST have a clear indication that there may be characters
         outside of the US-ASCII equivalent range.
```

Finally, the majority of ENUM Clients in use today do not support multi-byte encodings of UCS. This is a reasonable choice, particularly for "small footprint" implementations, and may not be able to support NAPTR content that is non-printable as they need to present the content to an end user for selection. Thus, it is RECOMMENDED that:

```
Client - Clients MAY discard NAPTRs in which they detect characters not
         in the US-ASCII "printable" range (0x20 to 0x7E hexadecimal).
```

ENUM zone population systems should consider this. It is RECOMMENDED that:

```
Server - ENUM zone content population systems SHOULD NOT use non-ASCII
         characters in the NAPTRs they generate unless they are sure
         that all Clients they intend to support will be able correctly
         to process them.
```

2.2. Case Sensitivity

The only place where NAPTR field content is case sensitive is in any static text in the repl sub-field of the RegExp field. Everywhere else, case insensitive processing can be used.

The case insensitivity flag ('i') may be added at the end of the RegExp field. However, in ENUM, the ERE sub-field operates on a string defined as the '+' character, followed by a sequence of digit characters. Thus this flag is redundant for E2U NAPTRs, as it does not act on the repl sub-field contents.

It is RECOMMENDED that:

Server - When populating ENUM zones with NAPTRs, population systems SHOULD NOT use the 'i' flag, as it has no effect and some Clients don't expect it.

Client - Clients SHOULD NOT assume that the field delimiter is the last character.

2.3. Regexp field delimiter

It is not possible to select a delimiter character that cannot appear in one of the sub-fields. Some old clients are "hardwired" to expect the character '!' as a delimiter.
This is used in an example in RFC3403.

It is RECOMMENDED that:

Server - ENUM zone population systems SHOULD use '!' (U+0021) as their RegExp delimiter character.

Client - Clients MAY discard NAPTRs that do not use '!' as a RegExp delimiter.

This cannot appear in the ERE sub-field. It may appear in the content of some URIs, as it is a valid character (e.g. in http URLs). Thus, it is further RECOMMENDED that:

Server - ENUM zone population systems MUST ensure that, if the RegExp delimiter is a character in the static text of the repl sub-field, it MUST be "escaped" using the escaped-delimiter production of the BNF specification shown in section 3.2 of RFC3402 (i.e. "\!", U+005C U+0021).

Client - Clients SHOULD discard NAPTRs that have more or less than 3 "unescaped" instances of the delimiter character within the RegExp field.

2.4. Regexp meta-character issue

In ENUM, the ERE sub-field may include a literal character '+', as the Application Unique String on which it operates includes this. However, if it is present, then '+' must be "escaped" using a backslash character as '+' is a meta-character in POSIX Extended Regular Expression syntax.

The following NAPTR example is incorrect:

```
* IN NAPTR 100 10 "u" "E2U+sip" "!^+46555(.*)$!sip:\1@sipcsp.se!" .
```

This example MUST be written as:

```
* IN NAPTR 100 10 "u" "E2U+sip" "!^\+46555(.*)$!sip:\1@sipcsp.se!" .
```

Server - If present in the ERE sub-field of an ENUM NAPTR, '+' MUST be written as "\+" (i.e. U+005C U+002B).

3. ORDER/PRIORITY Processing

3.1. Order/Priority values - general processing

RFC3761 and RFC3403 state that the client MUST sort the NAPTRs using the ORDER field value ("lowest value is first") and SHOULD order the

NAPTRs using the PRIORITY field value as the minor sort term (again, lowest value first). The NAPTRs in the sorted list must be processed in order. Subsequent NAPTRs with less preferred ORDER values must only be dealt with once the current ones with a "winning" ORDER value have been processed.

However, this expected behaviour is a simplification; clients may not behave this way in practice, and so there is a conflict between the specification and practice. For example, ENUM clients will be incapable of using most NAPTRs as they don't support the ENUMservice (and the URI generated by those NAPTRs). As such, they will discard the "unusable" NAPTRs and continue with processing the "next best" NAPTR in the list.

The end user may have pre-specified their own preference for services to be used. Thus, an end user may specify that they would prefer to use contacts with a "sip" ENUMservice, and then those with "msg:email" service, and are not interested in any other options. Thus the sorted list as proposed by the Registrant (and published via ENUM) may be re-ordered. For example, a NAPTR with a "sip" ENUMservice may have a low ORDER field value, and yet is chosen before a NAPTR with an "h323" ENUMservice and a high ORDER value. This may occur even if the node the end user controls is capable of handling other ENUMservices.

ENUM Clients may also include the end user "in the decision loop", offering the end user the choice from a list of possible NAPTRs. Given that the ORDER field value is the major sort term, one would expect a conforming ENUM Client to present only those NAPTRs with a "winning" ORDER field value as choices. However, if all the options presented had been rejected, then the ENUM Client might offer those with the "next best" ORDER field value, and so on. As this may be inconvenient for the end user, some clients simply offer all of the available NAPTRs as options to the end user for their selection "in one go".

In summary, some clients will take into account the service field value along with the ORDER and PRIORITY field values, and may consider the preferences of the end user.

The Registrant and the ENUM zone population system they use MUST be aware of this and SHOULD NOT rely on Clients taking account of the value of the ORDER and the PRIORITY fields.

Specifically, it is unsafe to assume that a Client WILL NOT consider another NAPTR until they have discarded one with a "winning" ORDER value. The instruction (in RFC3403 section 4.1 and section 8) may or may not be followed strictly by different ENUM clients for perfectly justifiable reasons.

To avoid the risk of variable Client behaviour, it is RECOMMENDED that:

Server - An ENUM zone population system SHOULD NOT use different ORDER values for NAPTRs within a zone.

In our experience, incorrect ORDER values in ENUM zones is a major source of problems. Although it is by no means required, it is further RECOMMENDED that:

Server - An ENUM zone population systems SHOULD use a value of 100 as the default ORDER value to be used with all NAPTRs.

As such, when populating a zone with NAPTRS, it is RECOMMEDED that:

```
Server - A Registrant SHOULD NOT expect the client to ignore NAPTRs
         with higher ORDER field values - the "winning" ones may have
         been discarded.

Server - A Registrant SHOULD NOT expect the client to conform to the
         ORDER and PRIORITY sort order they have specified for their
         NAPTRs; the end user may have their own preferences for
         ENUMservices.

Client - Clients MAY re-order the NAPTRs only to match an explicit
         preference pre-specified by their end user.

Client - Clients that offer a list of contacts to the end user for
         their choice MAY present all NAPTRs, not just the ones with
         the highest currently unprocessed ORDER field value.

Server - A Registrant SHOULD NOT assume which NAPTR choices will be
         presented "at once".
```

The impact of this is that a Registrant should place into their zone only contacts that they are willing to support; even those with the "least preferred" ORDER/PRIORITY values may be selected by an end user.

Finally, we have noticed a number of ENUM zones with NAPTRs that have identical PRIORITY field values and different ORDER values. This may be the result of a zone population system "bug" or a misunderstanding over the uses of the two fields.

To clarify, the ORDER field value is the major sort term, and the PRIORITY field value is the minor sort term. Thus one should expect to have a set of NAPTRs in a zone with identical ORDER field values and different PRIORITY field values.

3.2. Treatment of NAPTRs with identical ORDER/PRIORITY values

>From experience, there are zones that hold discrete NAPTRs with identical ORDER and PRIORITY field values. This is an error and so should not occur. However, in the spirit of being liberal in what is allowed:

It is RECOMMENDED that:

```
Client - Clients should accept all NAPTRs with identical ORDER/PRIORITY
         field values, and process them in the order in which they
         appear in the DNS response.
(There is no benefit in further randomizing the order in which these
 are processed, as intervening DNS servers may do this already).
```

Conversely, populating the records with these identical values is a mistake, and so it is RECOMMENDED that:

```
Server - When populating ENUM zones with NAPTRs, ENUM zone population
         systems SHOULD NOT have more than one with the same ORDER and
         the same PRIORITY field values, as Clients MAY reject the
         response.
```

There is a special case in which one could derive a set of NAPTRs with identical ORDER/PRIORITY fields. With RFC3761, it is possible to have more than one ENUMservice associated with a single NAPTR. Of course, the different ENUMservices share the same RegExp field and so generate the same URI. Such a "compound ENUMservice" NAPTR could well be used to indicate, for example, a mobile phone that supports both voice:tel and msg:sms ENUMservices.

This compound NAPTR may be deconstructed into a set of NAPTRs each holding a single ENUMservice. However, in this case the members of this set all hold the same ORDER and PRIORITY field values.

In this case, it is RECOMMENDED that:

Client - Clients receiving "compound" NAPTRs (i.e ones with more than one ENUMservice) should process these ENUMservices using a "left-to-right" sort ordering, so that the first ENUMservice to be processed will be the leftmost one, and the last will be the rightmost one.

Server - An ENUM zone population system SHOULD generate compound NAPTRs under the assumption that the ENUMservices will be processed in "left to right" order within such NAPTRs.

4. Non-Final NAPTR Processing

4.1. Non-final NAPTRs - general issues

To clarify, a non-final NAPTR should include an empty services field. Such a NAPTR will be significant regardless of the DDDS Application processing NAPTRs in a zone. Clients should ignore the service field contents when encountering a NAPTR with an empty flags field. As the RegExp field should be empty in this case, the client should also ignore this field.

To clarify, if all NAPTRs in a domain traversed as a result of a reference in a non-final NAPTR have been discarded, then the Client should continue its processing with the next NAPTR in the zone including the non-final NAPTR that caused the traversal.

To clarify, most ENUM Clients appear not to support non-final NAPTRs, and ignore them if received. This should be taken into account when populating a zone. Thus, it is RECOMMENDED that:

Server - ENUM zone population systems SHOULD NOT generate non-final NAPTRs (i.e. NAPTRs with an empty flags field).

Client - ENUM clients MAY discard non-final NAPTRs (i.e. they MAY only support ENUM NAPTRs with a flags field of "u").

4.2. Non-final NAPTRs - loop detection and response

Where a "chain" of Non-final NAPTRs refers back to a domain already traversed in the current query, this implies a "non-final loop".

It is RECOMMENDED that:

Client - Clients SHOULD consider processing more than 5 "non-final" NAPTRs in a single ENUM query to indicate that a loop may have been detected, and act accordingly.

Server - When populating a set of domains with NAPTRs, one should not configure non-final NAPTRs so that more than 5 such NAPTRs will be processed in an ENUM query.

Client - Where a domain is about to be entered as the result of a reference in a non-final NAPTR, and the client has detected a potential "non-final loop", then the Client should discard the non-final NAPTR from its processing and continue with the next NAPTR in its list.

5. Backwards Compatibility

5.1. Service field syntax

RFC3761 is the current standard for the syntax for NAPTRs supporting the ENUM DDDS application. This obsoletes the original specification that was given in RFC2916. There has been a change to the syntax of the services field of the NAPTR that reflects a refinement of the concept of ENUM processing.

As defined in RFC3403, there is now single identifier that indicates the DDDS Application. In the obsolete specification (RFC2915), there were zero or more "Resolution Service" identifiers (the equivalent of the DDDS Application). The same identifier string is defined in both RFC3761 and in the old RFC2916 specifications for the DDDS identifier or the Resolution Service; "E2U".

Also, RFC3761 defines at least one but potentially several ENUMservice sub-fields; in the obsolete specification, only one "protocol" sub-field was allowed.

In many ways, the most important change for implementations is that the order of the sub-fields has been reversed. RFC3761 specifies that the DDDS Application identifier is the leftmost sub-field, followed by one or more ENUMservice sub-fields, each separated by the '+' character delimiter. RFC2916 specified that the protocol sub-field was the leftmost, followed by the '+' delimiter, in turn followed by the "E2U" resolution service tag.

RFC2915 and RFC2916 have been obsoleted by RFC3401-RFC3404 and by RFC3761. Thus it is RECOMMENDED that:

Server - ENUM zone population systems MUST NOT generate NAPTRs according to the syntax defined in RFC2916. All zones MUST hold ENUM NAPTRs according to RFC3761 (and ENUMservice specifications according to the framework specified there).

However, RFC3834 suggests that clients SHOULD be prepared to accept NAPTRs with the obsolete syntax. Thus, a Client implementation may have to deal with both forms.

It is RECOMMENDED that:

Client - Clients MUST support ENUM NAPTRs according to RFC3761 syntax Clients SHOULD also support ENUM NAPTRs according to the obsolete syntax of RFC2916; there are still zones that hold "old" syntax NAPTRs.

This need not be difficult. For example, an implementation could process the services field into a set of tokens, and expect exactly one of these tokens to be "E2U". In this way, the client might be designed to handle both the old and the current forms without added complexity.

There is one subtle implication of this scheme. It is RECOMMENDED that:

```
Spec   - Registrations for an ENUMservice with the type string of
         "E2U" and an empty sub-type string MUST NOT be accepted.
```

6. Security Considerations

This document does not specify any standard. It does however make some recommendations, and so the implications of following those suggestions have to be considered.

In addition to these issues, those in the basic use of ENUM (and specified in the normative documents for this protocol) should be considered as well; this document does not negate those in any way.

The clarifications throughout this document are intended only as that; clarifications of text in the normative documents. They do not appear to have any security implications above those mentioned in the normative documents.

The suggestions in sections 2, 3, and 5 do not appear to have any security considerations (either positive or negative).

The suggestions in section 4.2 are a valid approach to a known security threat. It does not open an advantage to an attacker in client excess processing or memory usage. It does, however, mean that a client will traverse a "tight loop" of non-final NAPTRs in two domains 5 times before the client detects this as a loop; this does introduce slightly higher processing load that would be provided using other methods, but avoids the risks they incur.

7. IANA Considerations

This document is informational only, and does not include any IANA considerations other than the suggestion in section 5.1 that no-one should specify an enumservice with the identifying tag "E2U".

8. Acknowledgments

We would like to thank the various development teams who implemented ENUM (both creation systems and clients) and who read the normative documents differently - without these differences it would have been harder for us all to develop robust clients and suitably conservative management systems. We would also thank those who allowed us to check their implementations to explore behaviour; their trust and help were much appreciated.

In particular, thanks to Richard Stastny for his hard work on a similar task [12] under the aegis of ETSI, and for supporting some of the ENUM implementations that exist today.

Finally, thanks for the dedication of Michael Mealling in giving us such detailed DDDS specifications, without which the ENUM development effort would have had a less rigorous framework on which to build. This document reflects how complex a system it is - without the intricacy of

RFC3401-RFC3404 and the work that went into them, it could have been quite different.

9. Normative References

[1] Falstrom, P. and Mealling, M., "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery Systems (DDDS) Application (ENUM)", RFC 3761, April 2004

[2] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

[3] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[4] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

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

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

[7] IEEE, "IEEE Standard for Information Technology - Portable Operating System Interface (POSIX) - Part 2: Shell and Utilities (Vol. 1)", IEEE Std 1003.2-1992, January 1993.

[8] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[9] M. Mealling, RFC 3401, "Dynamic Delegation Discovery System (DDDS) Part One: The Comprehensive DDDS", October 2002

[10] M. Mealling, RFC 3402, "Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm", October 2002

[11] M. Mealling, RFC 3404, "Dynamic Delegation Discovery System (DDDS) Part Four: The Uniform Resource Identifiers (URI)", October 2002

10. Non-normative Reference

[12] ETSI TS 102 172, "Minimum Requirements for Interoperability of European ENUM Trials", July 2004

11. Authors' Addresses

Lawrence Conroy
Roke Manor Research

Old Salisbury Lane
Romsey
Hampshire SO51 0ZN
United Kingdom

email: lwc@roke.co.uk
URI: http://www.sienum.co.uk

Kazunori Fujiwara
Japan Registry Service Co., Ltd.
Chiyoda First Bldg. East 13F
3-8-1 Nishi-Kanda Chiyoda-ku
Tokyo 101-0065
JAPAN

email: fujiwara@jprs.co.jp
URI: http://jprs.jp/en/

12. IPR and Full Copyright Statement

Disclaimer of validity:

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

Disclaimer of Warranty

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 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

Acknowledgement

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