

Network Working Group  
Request for Comments: 4114  
Category: Standards Track

S. Hollenbeck  
VeriSign, Inc.  
June 2005

## E.164 Number Mapping for the Extensible Provisioning Protocol (EPP)

### Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2005).

### Abstract

This document describes an Extensible Provisioning Protocol (EPP) extension mapping for the provisioning and management of E.164 numbers that represent domain names stored in a shared central repository. Specified in XML, this mapping extends the EPP domain name mapping to provide additional features required for the provisioning of E.164 numbers.

## Table of Contents

1. Introduction . . . . .	2
1.1. Conventions Used in This Document . . . . .	3
2. Object Attributes . . . . .	3
2.1. E.164 Domain Names . . . . .	3
2.2. NAPTR Fields . . . . .	3
2.2.1. Order . . . . .	4
2.2.2. Preference . . . . .	4
2.2.3. Flags . . . . .	4
2.2.4. Service . . . . .	4
2.2.5. Regular Expression . . . . .	4
2.2.6. Replacement . . . . .	4
3. EPP Command Mapping . . . . .	4
3.1. EPP Query Commands . . . . .	5
3.1.1. EPP <check> Command . . . . .	5
3.1.2. EPP <info> Command . . . . .	5
3.1.3. EPP <transfer> Command . . . . .	7
3.2. EPP Transform Commands . . . . .	7
3.2.1. EPP <create> Command . . . . .	7
3.2.2. EPP <delete> Command . . . . .	9
3.2.3. EPP <renew> Command . . . . .	9
3.2.4. EPP <transfer> Command . . . . .	9
3.2.5. EPP <update> Command . . . . .	9
4. Formal Syntax . . . . .	11
5. Internationalization Considerations . . . . .	14
6. IANA Considerations . . . . .	14
7. Security Considerations . . . . .	14
8. Acknowledgements . . . . .	15
9. References . . . . .	15
9.1. Normative References . . . . .	15
9.2. Informative References . . . . .	16

## 1. Introduction

This document describes an E.164 number mapping for version 1.0 of the Extensible Provisioning Protocol (EPP). This mapping, an extension of the domain name mapping described in [1], is specified using the Extensible Markup Language (XML) 1.0, as described in [2], and XML Schema notation, as described in [3] and [4].

The EPP core protocol specification [5] provides a complete description of EPP command and response structures. A thorough understanding of the base protocol specification is necessary to understand the mapping described in this document.

ENUM [6] describes how the Domain Name System (DNS) can be used to identify services associated with an E.164 number. The EPP mapping described in this document specifies a mechanism for the provisioning and management of E.164 numbers stored in a shared central repository. Information exchanged via this mapping can be extracted from the repository and used to publish DNS resource records as described in ENUM [6]. Examples used in this document were chosen specifically to illustrate provisioning concepts for the example resource records described in the ENUM specification.

### 1.1. 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 RFC 2119 [7].

In examples, "C:" represents lines sent by a protocol client, and "S:" represents lines returned by a protocol server. Indentation and white space in examples are only provided to illustrate element relationships and are not a REQUIRED feature of this specification.

XML is case sensitive. Unless stated otherwise, XML specifications and examples provided in this document MUST be interpreted in the character case presented to develop a conforming implementation.

## 2. Object Attributes

This extension adds elements to the EPP domain name mapping [1]. Only new element descriptions are described here.

### 2.1. E.164 Domain Names

An E.164 domain name is a representation of an E.164 number that has been translated to conform to domain name syntax, as described in the ENUM specification [6]. The labels used to describe the name space of an E.164 domain name are a policy matter that is beyond the scope of this document.

### 2.2. NAPTR Fields

According to ENUM [6], Naming Authority Pointer (NAPTR) resource records are used to identify available ways for contacting a specific node identified by a domain name created from the translation of an E.164 number. The basic NAPTR record format is described in RFC 3403 [8]. Rules for structuring and using NAPTR records for use with ENUM are described in RFC 3761 [6].

Several NAPTR field values are optional per RFC 3403. RFC 3761 describes processing rules that require the presence of certain NAPTR field values. This document describes field value requirements that correspond to RFC 3761.

#### 2.2.1. Order

The NAPTR order field, a 16-bit unsigned integer, is represented in this mapping using the XML Schema "unsignedShort" data type.

#### 2.2.2. Preference

The NAPTR preference field, a 16-bit unsigned integer, is represented in this mapping using the XML Schema "unsignedShort" data type.

#### 2.2.3. Flags

The NAPTR flags field is represented in this mapping using a single character. The case of the flag character is not significant.

#### 2.2.4. Service

The NAPTR service field is represented in this mapping using a character string with an unspecified maximum length. Valid values are application-dependent.

#### 2.2.5. Regular Expression

The NAPTR regexp field is represented in this mapping using a character string with an unspecified maximum length. This field can contain numerous backslashes and should thus be treated with care.

#### 2.2.6. Replacement

The NAPTR replacement field, whose value is a domain name, is represented in this mapping using a character string with a maximum length of 255 characters.

### 3. EPP Command Mapping

A detailed description of the EPP syntax and semantics can be found in the EPP core protocol specification [5]. The command mappings described here are specifically for use in implementing ENUM provisioning processes via EPP.

### 3.1. EPP Query Commands

EPP provides three commands to retrieve object information: <check> to determine if an object is known to the server, <info> to retrieve detailed information associated with an object, and <transfer> to retrieve object transfer status information.

#### 3.1.1. EPP <check> Command

This extension does not add any elements to the EPP <check> command or <check> response described in the EPP domain mapping [1].

#### 3.1.2. EPP <info> Command

This extension does not add any elements to the EPP <info> command described in the EPP domain mapping [1]. Additional elements are defined for the <info> response.

When an <info> command has been processed successfully, the EPP <resData> element MUST contain child elements as described in the EPP domain mapping [1]. In addition, the EPP <extension> element MUST contain a child <e164:infData> element that identifies the extension namespace and the location of the extension schema. The <e164:infData> element contains one or more <e164:naptr> elements that contain the following child elements:

- An <e164:order> element that contains a NAPTR order value.
- An <e164:pref> element that contains a NAPTR preference value.
- An OPTIONAL <e164:flags> element that contains a NAPTR flags value.
- An <e164:svc> element that contains a NAPTR service value.
- An OPTIONAL <e164:regex> element that contains a NAPTR regular expression value.
- An OPTIONAL <e164:replacement> element that contains a NAPTR replacement value.

Example <info> response:

```
S:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
S:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
S:  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
S:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
S:    epp-1.0.xsd">
S:  <response>
S:    <result code="1000">
S:      <msg>Command completed successfully</msg>
S:    </result>
S:    <resData>
S:      <domain:infData
S:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
S:          domain-1.0.xsd">
S:        <domain:name>3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa</domain:name>
S:        <domain:roid>EXAMPLE1-REP</domain:roid>
S:        <domain:status s="ok"/>
S:        <domain:registrant>jd1234</domain:registrant>
S:        <domain:contact type="admin">sh8013</domain:contact>
S:        <domain:contact type="tech">sh8013</domain:contact>
S:        <domain:ns>
S:          <domain:hostObj>ns1.example.com</domain:hostObj>
S:          <domain:hostObj>ns2.example.com</domain:hostObj>
S:        </domain:ns>
S:        <domain:host>ns1.example.com</domain:host>
S:        <domain:host>ns2.example.com</domain:host>
S:        <domain:clID>ClientX</domain:clID>
S:        <domain:crID>ClientY</domain:crID>
S:        <domain:crDate>1999-04-03T22:00:00.0Z</domain:crDate>
S:        <domain:upID>ClientX</domain:upID>
S:        <domain:upDate>1999-12-03T09:00:00.0Z</domain:upDate>
S:        <domain:exDate>2005-04-03T22:00:00.0Z</domain:exDate>
S:        <domain:trDate>2000-04-08T09:00:00.0Z</domain:trDate>
S:        <domain:authInfo>
S:          <domain:pw>2fooBAR</domain:pw>
S:        </domain:authInfo>
S:      </domain:infData>
S:    </resData>
S:    <extension>
S:      <e164:infData xmlns:e164="urn:ietf:params:xml:ns:e164epp-1.0"
S:        xsi:schemaLocation="urn:ietf:params:xml:ns:e164epp-1.0
S:          e164epp-1.0.xsd">
S:        <e164:naptr>
S:          <e164:order>10</e164:order>
S:          <e164:pref>100</e164:pref>
S:          <e164:flags>u</e164:flags>
```

```
S:    <e164:svc>E2U+sip</e164:svc>
S:    <e164:regex>"!^.*$!sip:info@example.com!"</e164:regex>
S:    </e164:naptr>
S:    <e164:naptr>
S:    <e164:order>10</e164:order>
S:    <e164:pref>102</e164:pref>
S:    <e164:flags>u</e164:flags>
S:    <e164:svc>E2U+msg</e164:svc>
S:    <e164:regex>"!^.*$!mailto:info@example.com!"</e164:regex>
S:    </e164:naptr>
S:    </e164:infData>
S:    </extension>
S:    <trID>
S:    <clTRID>ABC-12345</clTRID>
S:    <svTRID>54322-XYZ</svTRID>
S:    </trID>
S: </response>
S:</epp>
```

An EPP error response MUST be returned if an extended <info> command can not be processed for any reason.

### 3.1.3. EPP <transfer> Command

This extension does not add any elements to the EPP <transfer> command or <transfer> response described in the EPP domain mapping [1].

## 3.2. EPP Transform Commands

EPP provides five commands to transform objects: <create> to create an instance of an object, <delete> to delete an instance of an object, <renew> to extend the validity period of an object, <transfer> to manage object sponsorship changes, and <update> to change information associated with an object.

### 3.2.1. EPP <create> Command

This extension defines additional elements for the EPP <create> command described in the EPP domain mapping [1]. No additional elements are defined for the EPP <create> response.

The EPP <create> command provides a transform operation that allows a client to create a domain object. In addition to the EPP command elements described in the EPP domain mapping [1], the command MUST contain an <extension> element. The <extension> element MUST contain a child <e164:create> element that identifies the extension namespace and the location of the extension schema. The <e164:create> element

contains one or more <e164:naptr> elements that contain the following child elements:

- An <e164:order> element that contains a NAPTR order value.
- An <e164:pref> element that contains a NAPTR preference value.
- An OPTIONAL <e164:flags> element that contains a NAPTR flags value.
- An <e164:svc> element that contains a NAPTR service value.
- An OPTIONAL <e164:regex> element that contains a NAPTR regular expression value.
- An OPTIONAL <e164:replacement> element that contains a NAPTR replacement value.

Example <create> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:    epp-1.0.xsd">
C:  <command>
C:    <create>
C:      <domain:create
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:          domain-1.0.xsd">
C:          <domain:name>3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa</domain:name>
C:          <domain:period unit="y">2</domain:period>
C:          <domain:ns>
C:            <domain:hostObj>ns1.example.com</domain:hostObj>
C:            <domain:hostObj>ns2.example.com</domain:hostObj>
C:          </domain:ns>
C:          <domain:registrant>jd1234</domain:registrant>
C:          <domain:contact type="admin">sh8013</domain:contact>
C:          <domain:contact type="tech">sh8013</domain:contact>
C:          <domain:authInfo>
C:            <domain:pw>2fooBAR</domain:pw>
C:          </domain:authInfo>
C:        </domain:create>
C:      </create>
C:    <extension>
C:      <e164:create
C:        xmlns:e164="urn:ietf:params:xml:ns:e164epp-1.0"
```

```
C:   xsi:schemaLocation="urn:ietf:params:xml:ns:e164epp-1.0
C:   e164epp-1.0.xsd">
C:   <e164:naptr>
C:     <e164:order>10</e164:order>
C:     <e164:pref>100</e164:pref>
C:     <e164:flags>u</e164:flags>
C:     <e164:svc>E2U+sip</e164:svc>
C:     <e164:regex>"!^.*$!sip:info@example.com!"</e164:regex>
C:   </e164:naptr>
C:   <e164:naptr>
C:     <e164:order>10</e164:order>
C:     <e164:pref>102</e164:pref>
C:     <e164:flags>u</e164:flags>
C:     <e164:svc>E2U+msg</e164:svc>
C:     <e164:regex>"!^.*$!mailto:info@example.com!"</e164:regex>
C:   </e164:naptr>
C: </e164:create>
C: </extension>
C: <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

When an extended `<create>` command has been processed successfully, the EPP response is as described in the EPP domain mapping [1].

### 3.2.2. EPP `<delete>` Command

This extension does not add any elements to the EPP `<delete>` command or `<delete>` response described in the EPP domain mapping [1].

### 3.2.3. EPP `<renew>` Command

This extension does not add any elements to the EPP `<renew>` command or `<renew>` response described in the EPP domain mapping [1].

### 3.2.4. EPP `<transfer>` Command

This extension does not add any elements to the EPP `<transfer>` command or `<transfer>` response described in the EPP domain mapping [1].

### 3.2.5. EPP `<update>` Command

This extension defines additional elements for the EPP `<update>` command described in the EPP domain mapping [1]. No additional elements are defined for the EPP `<update>` response.

The EPP <update> command provides a transform operation that allows a client to change the state of a domain object. In addition to the EPP command elements described in the EPP domain mapping [1], the <update> command MUST contain an <extension> element. The <extension> element MUST contain a child <e164:update> element that identifies the extension namespace and the location of the extension schema. The <e164:update> element contains one or more <e164:add> or <e164:rem> elements. Each <e164:add> and <e164:rem> element contains an <e164:naptr> element that contains the following child elements:

- An <e164:order> element that contains a NAPTR order value.
- An <e164:pref> element that contains a NAPTR preference value.
- An OPTIONAL <e164:flags> element that contains a NAPTR flags value.
- An <e164:svc> element that contains a NAPTR service value.
- An OPTIONAL <e164:regex> element that contains a NAPTR regular expression value.
- An OPTIONAL <e164:replacement> element that contains a NAPTR replacement value.

Example <update> command:

```
C:<?xml version="1.0" encoding="UTF-8" standalone="no"?>
C:<epp xmlns="urn:ietf:params:xml:ns:epp-1.0"
C:  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
C:  xsi:schemaLocation="urn:ietf:params:xml:ns:epp-1.0
C:    epp-1.0.xsd">
C:  <command>
C:    <update>
C:      <domain:update
C:        xmlns:domain="urn:ietf:params:xml:ns:domain-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:domain-1.0
C:          domain-1.0.xsd">
C:        <domain:name>3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa</domain:name>
C:      </domain:update>
C:    </update>
C:    <extension>
C:      <e164:update xmlns:e164="urn:ietf:params:xml:ns:e164epp-1.0"
C:        xsi:schemaLocation="urn:ietf:params:xml:ns:e164epp-1.0
C:          e164epp-1.0.xsd">
C:        <e164:rem>
C:          <e164:naptr>
C:            <e164:order>10</e164:order>
```

```
C:      <e164:pref>102</e164:pref>
C:      <e164:flags>u</e164:flags>
C:      <e164:svc>E2U+msg</e164:svc>
C:      <e164:regex>"!^.*$!mailto:info@example.com!"</e164:regex>
C:      </e164:naptr>
C:      </e164:rem>
C:      </e164:update>
C:      </extension>
C:      <clTRID>ABC-12345</clTRID>
C: </command>
C:</epp>
```

When an extended <update> command has been processed successfully, the EPP response is as described in the EPP domain mapping [1].

#### 4. Formal Syntax

An EPP object mapping is specified in XML Schema notation. The formal syntax presented here is a complete schema representation of the object mapping suitable for automated validation of EPP XML instances. The BEGIN and END tags are not part of the schema; they are used to note the beginning and ending of the schema for URI registration purposes.

BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<schema targetNamespace="urn:ietf:params:xml:ns:e164epp-1.0"
  xmlns:e164="urn:ietf:params:xml:ns:e164epp-1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified">
```

```
  <annotation>
    <documentation>
      Extensible Provisioning Protocol v1.0
      domain name extension schema for E.164 number provisioning.
    </documentation>
  </annotation>
```

```
<!--
```

Child elements found in EPP commands.

```
-->
```

```
  <element name="create" type="e164:createType"/>
  <element name="update" type="e164:updateType"/>
```

```
<!--
Global elements.
-->
  <element name="naptr" type="e164:naptrType"/>

<!--
Child elements of the <create> command.
-->
  <complexType name="createType">
    <sequence>
      <element ref="e164:naptr" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <complexType name="naptrType">
    <sequence>
      <element name="order" type="unsignedShort"/>
      <element name="pref" type="unsignedShort"/>
      <element name="flags" type="e164:flagsType"
        minOccurs="0"/>
      <element name="svc" type="e164:svcType"/>
      <element name="regex" type="e164:regexType"
        minOccurs="0"/>
      <element name="repl" type="e164:replType"
        minOccurs="0"/>
    </sequence>
  </complexType>

  <simpleType name="flagsType">
    <restriction base="token">
      <pattern value="[A-Z] | [a-z] | [0-9]"/>
      <length value="1"/>
    </restriction>
  </simpleType>

  <simpleType name="svcType">
    <restriction base="token">
      <minLength value="1"/>
    </restriction>
  </simpleType>

  <simpleType name="regexType">
    <restriction base="token">
      <minLength value="1"/>
    </restriction>
  </simpleType>
```

```
<simpleType name="replType">
  <restriction base="token">
    <minLength value="1"/>
    <maxLength value="255"/>
  </restriction>
</simpleType>

<!--
Child elements of the <update> command.
-->
<complexType name="updateType">
  <sequence>
    <element name="add" type="e164:addRemType"
      minOccurs="0"/>
    <element name="rem" type="e164:addRemType"
      minOccurs="0"/>
  </sequence>
</complexType>

<!--
Data elements that can be added or removed.
-->
<complexType name="addRemType">
  <sequence>
    <element ref="e164:naptr" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
Child response elements.
-->
<element name="infData" type="e164:infDataType"/>

<!--
<info> response elements.
-->
<complexType name="infDataType">
  <sequence>
    <element ref="e164:naptr" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!--
End of schema.
-->
</schema>
END
```

## 5. Internationalization Considerations

EPP is represented in XML, which provides native support for encoding information using the Unicode character set and its more compact representations, including UTF-8 [10]. Conformant XML processors recognize both UTF-8 and UTF-16 [11]. Though XML includes provisions to identify and use other character encodings through use of an "encoding" attribute in an `<?xml?>` declaration, use of UTF-8 is RECOMMENDED in environments where parser encoding support incompatibility exists.

As an extension of the EPP domain mapping [1], the elements, element content, attributes, and attribute values described in this document MUST inherit the internationalization conventions used to represent higher-layer domain and core protocol structures present in an XML instance that includes this extension.

## 6. IANA Considerations

This document uses URNs to describe XML namespaces and XML schemas conforming to a registry mechanism described in RFC 3688 [9]. Two URI assignments have been completed by the IANA:

Registration for the extension namespace:

URI: urn:ietf:params:xml:ns:e164epp-1.0

Registrant Contact: IESG

XML: None. Namespace URIs do not represent an XML specification.

Registration for the extension XML schema:

URI: urn:ietf:params:xml:schema:e164epp-1.0

Registrant Contact: IESG

XML: See the "Formal Syntax" section of this document.

## 7. Security Considerations

The mapping extensions described in this document do not provide any security services beyond those described by EPP [5], the EPP domain name mapping [1], and protocol layers used by EPP. Security considerations related to ENUM are described in the "Security Considerations" section of the ENUM specification [6]; security considerations related to the Dynamic Delegation Discovery System and NAPTR records are described in the "Security Considerations" section

of RFC 3403 [8]. The security considerations described in these specifications apply to this specification as well.

As with other domain object transforms, the EPP transform operations described in this document MUST be restricted to the sponsoring client as authenticated using the mechanisms described in sections 2.9.1.1 and 7 of RFC 3730 [5]. Any attempt to perform a transform operation on a domain object by any client other than the sponsoring client MUST be rejected with an appropriate EPP authorization error.

## 8. Acknowledgements

The author would like to thank the following people who have provided significant contributions to the development of this document:

Lawrence Conroy, Edward Lewis, Michael Mealling, Allison Mankin, Chip Sharp, and James Yu.

## 9. References

### 9.1. Normative References

- [1] Hollenbeck, S., "Extensible Provisioning Protocol (EPP) Domain Name Mapping", RFC 3731, March 2004.
- [2] Paoli, J., Sperberg-McQueen, C., Bray, T., and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C FirstEdition REC-xml-20001006, October 2000.
- [3] Maloney, M., Beech, D., Mendelsohn, N., and H. Thompson, "XML Schema Part 1: Structures", W3C REC REC-xmlschema-1-20010502, May 2001.
- [4] Malhotra, A. and P. Biron, "XML Schema Part 2: Datatypes", W3C REC REC-xmlschema-2-20010502, May 2001.
- [5] Hollenbeck, S., "Extensible Provisioning Protocol (EPP)", RFC 3730, March 2004.
- [6] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004.
- [7] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

- [8] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.
- [9] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

## 9.2. Informative References

- [10] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
- [11] Hoffman, P. and F. Yergeau, "UTF-16, an encoding of ISO 10646", RFC 2781, February 2000.

## Author's Address

Scott Hollenbeck  
VeriSign, Inc.  
21345 Ridgetop Circle  
Dulles, VA 20166-6503  
US

EMail: [shollenbeck@verisign.com](mailto:shollenbeck@verisign.com)

## Full Copyright Statement

Copyright (C) The Internet Society (2005).

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 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.

## Intellectual Property

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](mailto:ietf-ipr@ietf.org).

## Acknowledgement

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

