Internet Engineering Task Force (IETF)
Request for Comments: 6117
Obsoletes: 3761
Category: Standards Track
ISSN: 2070-1721

B. Hoeneisen
Ucom.ch
A. Mayrhofer
enum.at
J. Livingood
Comcast

March 2011

IANA Registration of Enumservices: Guide, Template, and IANA Considerations

Abstract

This document specifies a revision of the IANA Registration Guidelines for Enumservices, describes corresponding registration procedures, and provides a guideline for creating Enumservice Specifications.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at http://www.rfc-editor.org/info/rfc6117.

Table of Contents

1. Introduction

E.164 Number Mapping (ENUM) [RFC6116] provides an identifier mapping mechanism to map E.164 numbers [ITU.E164.2005] to Uniform Resource Identifiers (URIs) [RFC3986] using the Domain Name System (DNS) [RFC1035]. One of the primary concepts of ENUM is the definition of "Enumservices", which allows for providing different URIs for different applications of said mapping mechanism.

This document specifies a revision of the IANA registry for Enumservices, which was originally described in [RFC3761]. This document obsoletes Section 3 of RFC 3761 while RFC 6116 obsoletes RFC 3761.

The new registration processes, which are detailed in Section 6, have been specifically designed to be decoupled from the existence of the ENUM working group. Compared to RFC 3761, the main changes are as follows:

- For an Enumservice to be inserted to the IANA registry, "Specification Required", which implies the use of a Designated Expert, according to "Guidelines for Writing an IANA Considerations Section in RFCs" [RFC5226], are now sufficient.

- The IANA Registration Template has been supplemented with elements for "Enumservice Class" and "Enumservice Specification".

The IETF’s ENUM Working Group has encountered an unnecessary amount of variation in the format of Enumservice Specifications. The ENUM Working Group’s view of what particular information is required and/or recommended has also evolved, and capturing these best current practices is helpful in both the creation of new Enumservice Specifications, as well as the revision or refinement of existing Enumservice Specifications.

2. Terminology

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].

For the purpose of this document:

- "Registration Document" refers to a draft specification that defines an Enumservice and proposes its registration following the procedures outlined herein.

- "Enumservice Specification" refers to a Registration Document that has been approved by the experts and published according to "Specification Required" as defined in [RFC5226].

3. Registration Requirements

As specified in the Augmented Backus-Naur Form (ABNF, [RFC5234]) found in Section 3.4.3 of [RFC6116], an Enumservice is made up of Types and Subtypes. For any given Type, the allowable Subtypes (if any) must be defined in the Enumservice Specification. There is currently no concept of a registered Subtype outside the scope of a given Type.

While the combination of each Type and all of its Subtypes constitutes the allowed values for the "Enumservice" field, it is not sufficient to just list their allowed values. To allow for interoperability, a complete Enumservice Specification MUST document the semantics of the Type and Subtype values to be registered, and MUST contain all sections listed in Section 5 of this document.

Furthermore, in order for an Enumservice to be registered, the entire Registration Document requires approval by the experts according to "Specification Required", which implies the use of a Designated Expert, as set out in "Guidelines for Writing an IANA Considerations Section in RFCs" [RFC5226] and Section 7.2 of this document.

All Enumservice Specifications are expected to conform also to various requirements laid out in the following sections.

3.1. Functionality Requirements

A registered Enumservice must be able to function as a selection mechanism for choosing one Naming Authority Pointer (NAPTR) [RFC3403] DNS Resource Record (RR) from a set of such RRs. That means the Enumservice Specification MUST define how to use the NAPTR RR and the URI(s) the NAPTR RR resolves to.

Specifically, a registered Enumservice MUST specify the URI Scheme(s) that may be used for the Enumservice, and, when needed, other information that will have to be transferred into the URI resolution process itself.

3.2. Naming Requirements

The name of an Enumservice MUST be unique in order to be useful as a selection criteria:

- The Type MUST be unique.

- The Subtype (being dependent on the Type) MUST be unique within a given Type.

Types and Subtypes MUST conform to the ABNF specified in Section 3.4.3 of [RFC6116].

The ABNF specified in Section 3.4.3 of [RFC6116] allows the "-" (dash) character for Types and Subtypes. To avoid confusion with possible future prefixes, a "-" MUST NOT be used as the first nor as the second character of a Type nor a Subtype. Furthermore, a "-" MUST NOT be used as the last character of a Type nor a Subtype. In addition, Types and Subtypes are case insensitive and SHOULD be specified in lowercase letters.

Note: The legacy IANA registry of Enumservices contains Type and Subtype strings with uppercase letters. Implementors could be tempted to refuse handling uppercase Type or Subtype strings, which could negatively affect interoperability.

To avoid confusion with Enumservice fields using a deprecated (obsolete) syntax, Type and Subtype MUST NOT start with the string "e2u".

The Subtype for one Type MAY have the same identifier as a Subtype for another Type, but it is not sufficient to simply reference another Type's Subtype. The functionality of each Subtype MUST be fully specified in the context of the Type being registered.

Section 4 contains further naming requirements.

3.3. Security Requirements

An analysis of security issues is REQUIRED for all registered Enumservices. (This is in accordance with the basic requirements for all IETF protocols.)

All descriptions of security issues MUST be as accurate and extensive as feasible. In particular, a statement that there are "no security issues associated with this Enumservice" must not be confused with "the security issues associated with this Enumservice have not been assessed".

There is no requirement that an Enumservice must be completely free of security risks. Nevertheless, all known security risks MUST be identified in an Enumservice Specification.

Some of the issues to be looked at in a security analysis of an Enumservice are:

1. Complex Enumservices may include provisions for directives that institute actions on a user's resources. In many cases provision can be made to specify arbitrary actions in an unrestricted fashion which may then have devastating results (especially if there is a risk for a new ENUM look-up, and because of that an infinite loop in the overall resolution process of the E.164 number).

2. Complex Enumservices may include provisions for directives that institute actions which, while not directly harmful, may result in disclosure of information that either facilitates a subsequent attack or else violates the users' privacy in some way.

3. An Enumservice might be targeted for applications that require some sort of security assurance but do not provide the necessary security mechanisms themselves. For example, an Enumservice could be defined for storage of confidential security services information such as alarm systems or message service passcodes, which in turn require an external confidentiality service.

3.4. Publication Requirements

Enumservices Specifications MUST be published according to the requirements for "Specification Required" set out in "Guidelines for Writing an IANA Considerations Section in RFCs" [RFC5226]. RFCs fulfill these requirements. Therefore, it is strongly RECOMMENDED to publish Enumservice Specifications as RFCs.

In case the Enumservice Specification is not published as an RFC, sufficient information that allows unique identification of the Enumservice Specification MUST be provided.

4. Enumservice Creation Cookbook

4.1. General Enumservice Considerations

ENUM is an extremely flexible identifier mapping mechanism, using E.164 (phone) numbers as input identifiers, and returning URIs as output identifiers. Because of this flexibility, almost every use case for ENUM could be implemented in several ways.

Section 2 of "Guidelines for Writing an IANA Considerations Section in RFCs" [RFC5226] provides motivation for why management of a namespace might be necessary. Even though the namespace for Enumservices is rather large (up to 32 alphanumeric characters), there are reasons to manage this in accordance with Section 2 of [RFC5226]. The following is a list of motivations applying to Enumservices:

- Prevent hoarding or wasting of values: Enumservice Types are not an opaque identifier to prevent collisions in the namespace, but rather identify the use of a certain technology in the context of ENUM. Service Types might also be displayed to end users in implementations, so meaningful Type strings having a clear relation to the protocols and applications used are strongly RECOMMENDED. Therefore, preventing hoarding, wasting, or "hijacking" of Enumservice Type strings is important.

- Sanity check to ensure sensible or necessary requests: This applies to Enumservices, since especially various Enumservices for the same purpose would reduce the chance of successful interoperability, and unnecessarily increase confusion among implementers.

- Delegation of namespace portions: Theoretically, the Type and/or Subtype structure of Enumservices would allow for delegations of Type values, and self-supporting management of Subtype values by a delegate within the Type value. Such delegates could, for example, be other standardization bodies. However, this would require clear policies regarding publication and use of such Subtypes. Delegation of Enumservice namespace portions is therefore currently not supported.

- Interoperability: Since the benefit of an Enumservice rises with the number of supporting clients, the registration and use of several services for a similar or identical purpose clearly reduces interoperability. Operational circumstances suggest to keep the space occupied by all services published in the NAPTR RRSet at any owner in the e164.arpa domain bounded. Registration of nearly identical services and subsequent competing or parallel use could easily increase the DNS operational complexity.

Generally, before commencing work on a new Enumservice registration, the following should be considered:

- Is there an existing Enumservice that could fulfill the desired functionality without overloading it? Check the IANA Enumservice Registry at <http://www.iana.org/assignments/enum-services>.

- Is there work in progress, or previous work, on a similar Enumservice? Check the <enum@ietf.org> mailing list archives at <http://www.ietf.org/mail-archive/web/enum/index.html>, and search the Internet-Drafts Archive at <http://tools.ietf.org/>. Some Internet-Drafts may have expired and no longer be available in the Internet-Drafts Archive, or some work on Enumservices may have been considered outside the IETF; therefore, we also recommend a web search.

- Section 4.2 provides three general categories for Enumservice classification. In some cases, there might be several options for designing an Enumservice. For example, a mapping service using HTTP could be considered a "protocol Type" Enumservice (using HTTP as the protocol), while it could also be viewed as an "application Type" Enumservice, with the application providing access to mapping services. In such a case where several options are available, defining use cases before commencing work on the Enumservice itself might be useful before making a decision on which aspect of the Enumservice is more important.

4.2. Classification, Type and Subtype

Because of their flexibility, Enumservices can be and are used in a lot of different ways. This section contains a classification of Enumservices, and provides guidance for choosing suitable Type and Subtype strings for each individual Enumservice Class.

The Classification of each Enumservice MUST be listed in the Registration Document (see Section 5.2). If the Enumservice cannot be assigned to one of the classes outlined below, the Registration Document MUST contain a section on the difficulties encountered while trying to classify the service to help the experts in their decision.

4.2.1. General Type/Subtype Considerations

To avoid confusion, the name of a URI Scheme MUST NOT be used as a Type string for an Enumservice that is not specifically about the respective protocol or URI Scheme. For example, the Type string "imap" would be inadequate for use in an Enumservice about "Internet mapping" services, because it corresponds to an existing URI Scheme / protocol for something different.

If Subtypes are defined, the minimum number SHOULD be two (including the empty Subtype, if defined). The choice of just one possible Subtype for a given Type does not add any information when selecting an ENUM record, and hence can be left out completely. However, potential future expansion of a Type towards several Subtypes may justify the use of Subtypes, even in the case that just one is currently defined, as noted in Section 9.

It is perfectly legal under a certain Type to mix the Enumservice without a Subtype (empty Subtype) with Enumservices containing a Subtype. In that case, however, the Enumservice with an empty Subtype SHOULD be specified to reflect the base service, while the other Enumservices SHOULD be specified to reflect variants.

4.2.2. Protocol-Based Enumservices Class

Such an Enumservice indicates that an interaction using the named protocol will result for use of this NAPTR. The expected behavior of a system using this Enumservice MUST be clear from the protocol.

A good indication that an Enumservice belongs to this Class is the fact that a client does not need to understand the actual application to make use of an instance of this Enumservice.

Examples of such Enumservices include "xmpp" [RFC4979] and "sip" [RFC3764].

4.2.2.1. Protocol-Based Enumservice "Type" Strings

A protocol-based Enumservice SHOULD use the lowercase name of the protocol as its Type string. Names as registered in the IANA Port Number Registry (<http://www.iana.org/assignments/port-numbers>, defined in Section 8 and 9 of [RFC2780]) are preferred.

4.2.2.2. Protocol-Based Enumservice "Subtype" Strings

Where there is a single URI Scheme associated with this protocol, a Subtype SHOULD NOT be specified for the Enumservice.

Where there are a number of different URI Schemes associated with this protocol, the Enumservice Specification MAY use the empty Subtype for all URI Schemes that it specifies as mandatory to implement. For each URI Scheme that is not mandatory to implement, a distinct Subtype string MUST be used.

If Subtypes are defined, it is RECOMMENDED to use the URI Scheme name as the Subtype string.

4.2.3. Application-Based Enumservice Classes

Application-based Enumservices are used when the kind of service intended is not fully defined by a protocol specification. There are three cases here:

- Common Application Enumservice:

  The application reflects a kind of interaction that can be realized by different protocols, but where the intent of the publisher is the same. From a user’s perspective, there is a common kind of interaction -- how that interaction is implemented is not important. The Enumservice Specification MUST describe the interaction and expected behavior in enough detail that an

implementation can decide if this activity is one in which it can engage. However, it is RECOMMENDED that the Enumservice be defined in a way that will allow others to use it at a later date. An Enumservice that defines a generalized application is preferred to one that has narrow use.

An example of this flavor of Enumservice is email. Whilst this might appear to be a "pure" protocol scheme, it is not. The URI Scheme is 'mailto', and it does not identify the protocol used to offer or retrieve emails by the sender or the recipient.

Another example is the Short Messaging Service (SMS), where the existence of such an Enumservice indicates that the publishing entity is capable of engaging in sending or receiving a message according to the SMS specifications. The underlying protocol used and the URI Scheme for the addressable end point can differ, but the "user visible" interaction of sending and receiving an SMS is similar.

- Subset Enumservice:

  The application interaction reflects a subset of the interactions possible by use of a protocol. Use of this Enumservice indicates that some options available by use of the protocol will not be accepted or are not possible in this case. Any such Enumservice Specification MUST define the options available by use of this NAPTR in enough detail that an implementation can decide whether or not it can use this Enumservice. Examples of this kind of Enumservice are "voice:tel" and "fax:tel". In both cases, the URI holds a telephone number. However, the essential feature of these Enumservices is that the telephone number is capable of receiving a voice call or of receiving a Facsimile transmission, respectively. These form subsets of the interactions capable of using the telephone number, and so have their own Enumservices. These allow an end point to decide if it has the appropriate capability to engage in the advertised user service (a voice call or sending a fax) rather than just being capable of making a connection to such a destination address. This is especially important where there is no underlying mechanism within the protocol to negotiate a different kind of user interaction.

- Ancillary Application Enumservice

  Another variant on this is the Ancillary Application. This is one in which further processing (potentially using a number of different protocols or methods) is the intended result of using this Enumservice. An example of this kind of application is the "pstn:tel" Enumservice. This indicates that the NAPTR holds

number portability data. It implies that the client should engage in number portability processing using the associated URI. Note that this Enumservice usually does not itself define the kind of interaction available using the associated URI. That application is negotiated with some other "out of band" means (either through prior negotiation, or explicitly through the number portability process, or through negotiation following the selection of the final destination address).

4.2.3.1. Application-Based Enumservice "Type" Strings

It is recommended that Application-class Enumservices use the lowercase well-known name of the abstract application as the Type string.

4.2.3.2. Application-Based Enumservice "Subtype" Strings

It is RECOMMENDED that the URI Scheme(s) used by the application be used as the Subtype string(s). Subtype strings MAY be shared between URI Schemes, if all the URI Schemes within the same Subtype are mandatory to implement.

If it is foreseen that there is only one URI Scheme ever to be used with the application, the empty Subtype string MAY be used.

4.2.4. Data Type-Based Enumservice Class

"Data Type" Enumservices typically refer to a specific data type or format, which may be addressed using one or more URI Schemes and protocols. Examples of such Enumservices include "vpim" [RFC4238] and "vcard" [RFC4969].

4.2.4.1. Data Type-Based Enumservice "Type" Strings

It is recommended to use the lowercase well known name of the data type or format name as the Type string.

4.2.4.2. Data Type-Based Enumservice "Subtype" Strings

It is RECOMMENDED to use the URI Schemes used to access the service as Subtype strings. Subtype strings MAY be shared between URI Schemes, if all the URI Schemes within the same Subtype are mandatory to implement.

If there is only one URI Scheme foreseen to access the data type or format, the empty Subtype string MAY be used.

4.2.5. Other Enumservice

In case an Enumservice proposal cannot be assigned to any of the classes mentioned above, the <class> element (Enumservice Class) in the IANA Registration Template (see Section 5.2) MUST be populated with "Other". In that case, the Enumservice Specification MUST contain a section elaborating on why the Enumservice does not fit into the classification structure.

5. Required Sections and Information

There are several sections that MUST appear in an Enumservice Specification. These sections are as follows, and they SHOULD be in the given order.

The following terms SHOULD begin with a capital letter, whenever they refer to the IANA Registration:

- Class
- Type
- Subtype
- URI Scheme

5.1. Introduction (REQUIRED)

An introductory section MUST be included. This section will explain, in plain English, the purpose and intended use of the proposed Enumservice registration.

The Introduction SHOULD start with a short sentence about ENUM, introduce the protocol used in the Enumservice, and discuss the Enumservice as it refers from the E.164 number to the protocol or service.

5.2. IANA Registration (REQUIRED)

This section MUST be included in an Enumservice Specification. Where a given Enumservice Type has multiple Subtypes, there MUST be a separate "IANA Registration" section for each Subtype. The following sections list the elements that are to be used in the XML-chunk-based Registration Template of an "IANA Registration" section.

5.2.1. Enumservice Class (<class>)

This element contains the Class of the Enumservice as defined in Section 4.2. Its value MUST be one of (without quotes):

- "Protocol-Based": The Enumservice belongs to the Protocol-based class as described in Section 4.2.2.

- "Application-Based, Common": The Enumservice is a "common" case of the Application-based class as described in Section 4.2.3.
- "Application-Based, Subset": The Enumservice belongs to the "subset" case of the Application-based class as described in Section 4.2.3.
- "Application-Based, Ancillary": The Enumservice is an "ancillary" case of the Application-based class, as described in Section 4.2.3.
- "Data Type-Based": The Enumservice belongs to the Data Type-Based class as described in Section 4.2.4.
- "Other": The majority of the functionality of the Enumservice does not fall into one of the classes defined.

Class Example

```
<class>Protocol-Based</class>
```

5.2.2. Enumservice Type (<type>)

The Type of the Enumservice. All Types SHOULD be listed in lower-case. The choice of Type depends on the Enumservice Class. Please find further instructions in Section 4.

Type Example

```
<type>foo</type>
```

5.2.3. Enumservice Subtype (<subtype>)

The Subtype of the Enumservice. All Subtypes SHOULD be listed in lower-case. The choice of Subtype depends on the Enumservice Class. Should the Enumservice not utilize a Subtype, then the <subtype> element MUST be omitted in the IANA Registration Template. If a given Enumservice Type has multiple Subtypes, then there MUST be a separate IANA Registration Template for each Subtype. Please find further instructions in Section 4.

Subtype Example

```
<subtype>bar</subtype>
```

5.2.4. URI Scheme(s) (<urischeme>)

The URI Schemes [RFC3986] that are used with the Enumservice. The selection of URI Schemes often depends on the Enumservice Class, Type, and/or Subtype. A colon MUST NOT be placed after the URI Scheme name. If there is more than one URI Scheme, then one <urischeme> element per URI scheme MUST be used in the IANA Registration Template. Please find further instructions in Section 4.

URI Scheme Example

```
<urischeme>bar</urischeme>
<urischeme>sbar</urischeme>
```

Note: A client cannot choose a specific ENUM record in a record set based on the URI Scheme - the selection is only based on Type and Subtype, in accordance with [RFC3402].

5.2.5. Functional Specification (<functionalspec>)

The Functional Specification describes how the Enumservice is used in connection with the URI to which it resolves.

Functional Specification Example

```
<functionalspec>
  <paragraph>
    This Enumservice indicates that the resource
    identified can be addressed by the associated
    URI in order to foo the bar.
  </paragraph>
  <paragraph>
    [...]
  </paragraph>
</functionalspec>
```

Where the terms used are non-obvious, they should be defined in the Enumservice Specification, or a reference to an external document containing their definition should be provided.

5.2.6. Security Considerations (<security>)

A reference to the "Security Considerations" section of a given Enumservice Specification.

Security Considerations Example

```
<security>
  See <xref type="rfc" data="rfc4979"/>, Section 6.
</security>
```

5.2.7. Intended Usage (<usage>)

One of the following values (without quotes):

- "COMMON": Indicates that the Enumservice is intended for widespread use on the public Internet, and that its scope is not limited to a certain environment.

- "LIMITED USE": Indicates that the Enumservice is intended for use on a limited scope, for example in private ENUM-like application scenarios. The use case provided in the Enumservice Specification should describe such a scenario.

- "DEPRECATED": Indicates that the Enumservice has been declared deprecated (Section 11.7) and is not to be used in new deployments. Applications SHOULD however expect to encounter legacy instances of this Enumservice.

Intended Usage Example

```
<usage>COMMON</usage>
```

5.2.8. Enumservice Specification (<registrationdocs>)

Reference(s) to the Document(s) containing the Enumservice Specification.

Enumservice Specification Examples

```
<registrationdocs>
  <xref type="rfc" data="rfc4979"/>
</registrationdocs>
```

or

```
<registrationdocs>
  <xref type="rfc" data="rfc2026"/> (obsoleted by RFC 2551)
  <xref type="rfc" data="rfc2551"/>
</registrationdocs>
```

or

```
<registrationdocs>
  [International Telecommunications Union,
  "Enumservice Specification for Foobar",
  ITU-F Recommendation B.193, Release 73, Mar 2009.]
</registrationdocs>
```

5.2.9. Requesters (<requesters>)

The persons requesting the registration of the Enumservice. Usually these are the authors of the Enumservice Specification.

Requesters Example

```
<requesters>
  <xref type="person" data="John_Doe"/>
</requesters>

[...]

<people>
  <person id="John_Doe">
    <name>John Doe</name>
    <org>ACME Research and Development Inc.</org>
    <uri>mailto:jd@acme.example.com</uri>
    <updated>2008-11-20</updated>
  </person>
</people>
```

If there is more than one requester, there MUST be one <xref> element per requester in the <requesters> element, and one <person> chunk per requester in the <people> element.

5.2.10. Further Information (<additionalinfo>)

Any other information the authors deem interesting, including artwork.

Further Information Example

```
<additionalinfo>
  <paragraph>more info goes here</paragraph>
</additionalinfo>
```

Note: If there is no such additional information, then the <additionalinfo> element is omitted.

5.3. Examples (REQUIRED)

This section MUST show at least one example of the Enumservice being registered, for illustrative purposes. The example(s) shall in no way limit the various forms that a given Enumservice may take, and this should be noted at the beginning of this section of the document. The example(s) MUST show the specific formatting of the intended NAPTRs (according to [RFC3403] and [RFC6116]), including one or more NAPTR example(s), AND a brief textual description, consisting of one or more sentences written in plain English, explaining the various parts or attributes of the record(s).

The example(s) SHOULD contain a brief description how a client supporting this Enumservice could behave, if that description was not already given in, e.g., the Introduction or the Functional Specification.

The example(s) SHOULD follow any relevant IETF guidelines on the use of domain names, phone numbers, and other resource identifier examples, such as [RFC2606].

For example:

```
$ORIGIN 9.7.8.0.6.9.2.3.6.1.4.4.e164.arpa.
@ IN NAPTR 100 10 "u" "E2U+foo:bar" "!^.*$!bar://example.com/!" .
```

5.4. Implementation Recommendations / Notes (OPTIONAL)

Recommendations that pertain to implementation and/or operations SHOULD be included. Such a section is helpful to someone reading an Enumservice Specification and trying to understand how best to use it to support their network or service.

5.5. DNS Considerations (REQUIRED)

In case the inclusion of protocols and URI Schemes into ENUM specifically introduces new DNS issues, those MUST be described within this section.

Such DNS issues include, but are not limited to:

- Assumptions about ownership or administrative control of the namespace.
- Requirement or need to use DNS wildcards.
- Incompatibility with DNS wildcards.

- Presence or absence of respective NAPTR Resource Records at particular levels in the DNS hierarchy (e.g., only for "full" E.164 numbers or wildcards only).

- Use of any Resource Records (especially non-NAPTR) within or beyond the e164.arpa namespace other than those needed to resolve the domain names that appear in the "replacement" URI.

- Potential for significant additional load on the nameserver chain due to use of the service, and the mitigation of such additional load.

- Mitigation of potential for DNS loops, specifically in cases where the result URI of an Enumservice might be used to trigger additional (subsequent) ENUM queries. This applies in particular to Enumservices using the 'tel' URI Scheme [RFC3966] or any other (future) URI Scheme using (E.164) numbers. "The ENUM Dip Indicator Parameter for the tel URI" [RFC4759] provides an example of a loop mitigation mechanism.

Rationale: some Enumservices try to exploit side effects of the DNS that need to be explicitly discussed.

5.6. Security Considerations (REQUIRED)

A section explaining any potential security threats that are especially applicable to the given registration MUST be included. This MUST also include any information about access to Personally Identifiable Information (PII).

An Enumservice Specification SHOULD NOT include general and obvious security recommendations, such as securing servers with strong password authentication.

For additional background, please note that [RFC3552] provides guidance to write a good Security Considerations section. In addition, [RFC6116] already outlines security considerations affecting ENUM as a whole. Enumservice Specifications do not need to and SHOULD NOT repeat considerations already listed in that document. However, Enumservice Specifications SHOULD include a reference to that section.

Also, ENUM refers to resources using existing URI Schemes and protocols. Enumservice Specifications do not need to and SHOULD NOT repeat security considerations affecting those protocols and URI Schemes themselves.

However, in some cases, the inclusion of those protocols and URI Schemes into ENUM specifically could introduce new security issues. In these cases, those issues or risks MUST be covered in the "Security Considerations" section of the Enumservice Specification. Authors should pay particular attention to any indirect risks that are associated with a proposed Enumservice, including cases where the proposed Enumservice could lead to the discovery or disclosure of Personally Identifiable Information (PII).

5.7. IANA Considerations (REQUIRED)

Describe the task IANA needs to fulfill to process the Enumservice Registration Document.

For example:
This document requests the IANA registration of the Enumservice with Type "foo" and Subtype "bar" according to the definitions in this document, [RFC6117], and [RFC6116].

For example:
This document requests an update of the IANA registration of the Enumservice Type "foo" with Subtype "bar", according to the definitions in this document, [RFC6117], and [RFC6116]. Therefore, in the existing IANA registration for this Enumservice, the <registrationdocs> element (Enumservice Specification) is enhanced by adding a supplementary reference that points to this document.

For example:
This document requests an update of the IANA registration of the Enumservice Type "foo" with all its Subtypes, in order to declare it deprecated. Therefore, in the existing IANA registration for this Enumservice, the <usage> element (Intended Usage) is changed to "DEPRECATED", and the <registrationdocs> element (Enumservice Specification) is enhanced by adding a supplementary reference that points to this document.

5.8. Other Sections (OPTIONAL)

Other sections beyond those required above MAY be included in an Enumservice Specification. These sections may relate to the specifics of the intended use of the Enumservice registration, as well as to any associated technical, operational, administrative, or other concerns.

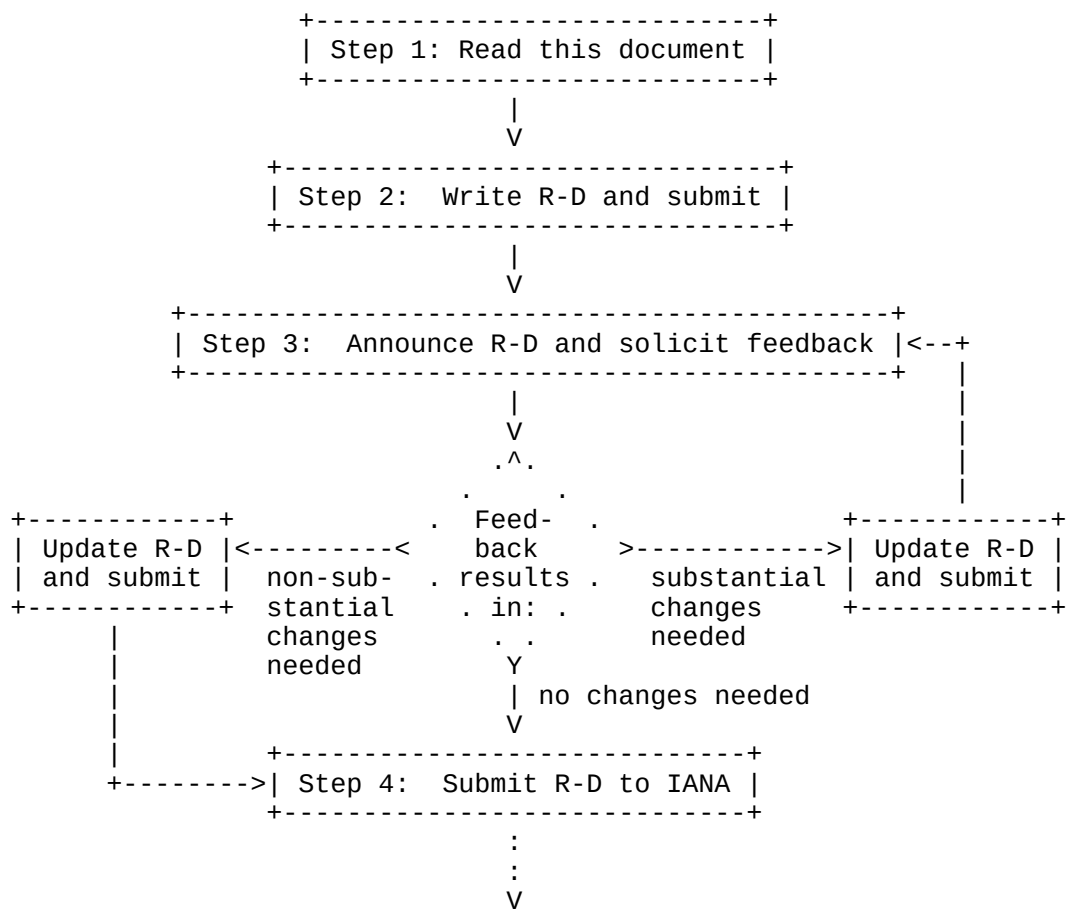
A use case SHOULD be included by the authors of the proposal, so that experts can better understand the problem the proposal seeks to solve (intended use of the Enumservice). The inclusion of such a use case

will both accelerate the Expert Review process, as well as make any eventual registration easier to understand and implement by other parties.

6. The Process of Registering New Enumservices

This section is an illustration of the process by which a new Enumservice Registration Document is submitted for review and comment, how such proposed Enumservices are reviewed, and how they are published. This section is a non-normative description of the process. The normative process is described in [RFC5226].

Figure 1 shows what authors of a Registration Document describing an Enumservice must carry out before said Registration Document can be formally submitted to IANA for Expert Review. Figure 2 shows the process from Expert Review onwards.



R-D: Registration Document

Figure 1

6.1. Step 1: Read This Document in Detail

This document, particularly in Sections 3, 4, and 5, describes all of the recommended and required sections, as well as requirements and suggestions for content of an Enumservice Specification.

6.2. Step 2: Write and Submit Registration Document

An Internet-Draft (or another specification as appropriate) must be written and made publicly available (submitted). The Registration Document shall follow the guidelines according to Sections 4 and 5 of

this document. The Review Guidelines for experts are defined in Section 7.2.

6.3. Step 3: Request Comments From the IETF Community

The authors shall send an email to <enum@ietf.org>, in which comments on the Registration Document are requested. A proper public reference (a URL is recommended) to the Registration Document must be included in this email.

Note: The ENUM WG mailing list <enum@ietf.org> will be kept open after conclusion of the ENUM WG.

The authors should allow a reasonable period of time to elapse, such as two to four weeks, in order to collect any feedback. The authors then consider whether or not to take any of those comments into account, by making changes to the Registration Document and submitting a revision, or otherwise proceeding. The following outcomes are open to the authors. The choice of path is left to the authors' judgement.

Note: Whatever the outcome is, the experts performing the Expert Review later in the process are not bound to any decision during this phase.

6.3.1. Outcome 1: No Changes Needed

No changes to the Registration Document are made, and the authors proceed to Step 4 below.

This outcome is recommended when the feedback received does not lead to a new revision of the Registration Document.

6.3.2. Outcome 2: Changes, But No Further Comments Requested

The authors update the Registration Document and is/are confident that all issues are resolved and do not require further discussion. The authors proceed to Step 4 below.

This outcome is recommended when minor objections have been raised, or minor changes have been suggested.

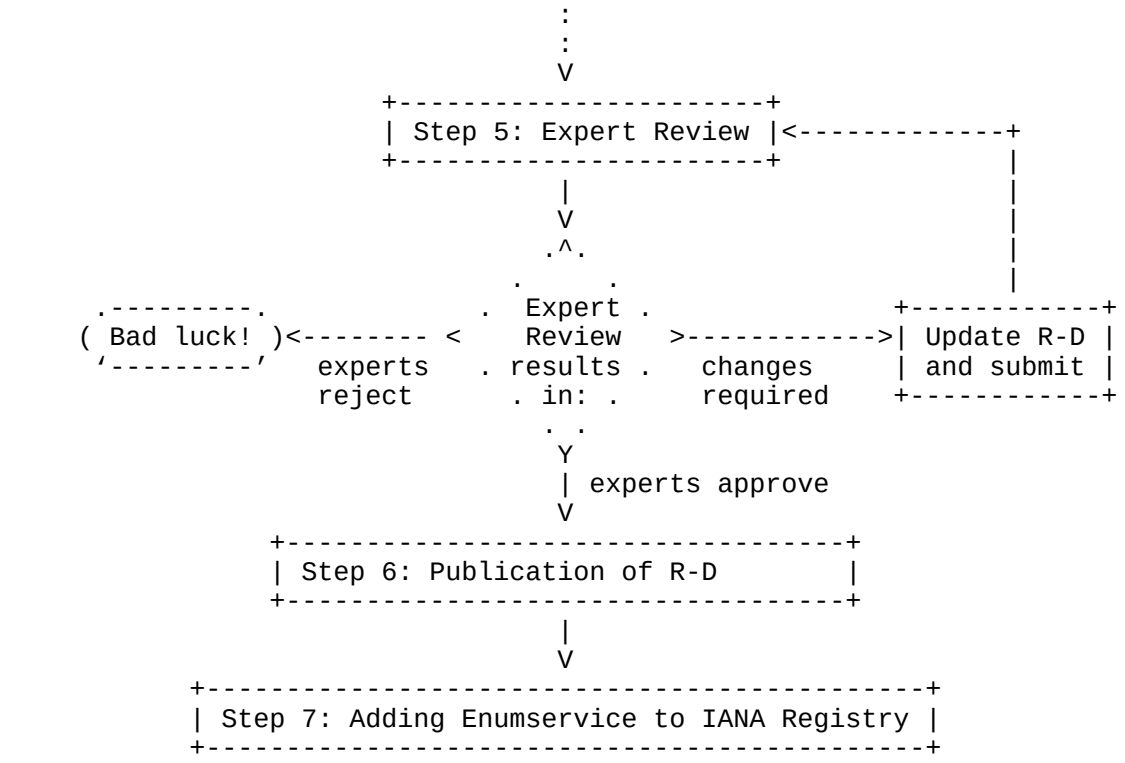
6.3.3. Outcome 3: Changes and Further Comments Requested

The authors update and submit the Registration Document, and proceed to Step 3 above, which involves sending another email to <enum@ietf.org> to request additional comments for the updated version.

This outcome is recommended when substantial objections have been raised, or substantial changes have been suggested.

6.4. Step 4: Submit Registration Document to IANA

The authors submit the Registration Document to IANA (using the <http://www.iana.org/> website) for Expert Review.



R-D: Registration Document

Figure 2

6.5. Step 5: Expert Review

IANA will take care of the "Expert Review" according to [RFC5226]. The Expert Review guidelines are outlined in Section 7.2 of this document. The authors must be prepared for further interaction with IANA and the experts.

6.5.1. Outcome 1: Experts Approve the Registration Document

No (more) changes to the Registration Document are made. IANA will inform the authors, who then will proceed to Step 6 below.

6.5.2. Outcome 2: Changes Required

The experts might require changes before they can approve the Registration Document. The authors update and submit the Registration Document. The authors inform the experts about the available update, who then continue the Expert Review Process.

6.5.3. Outcome 3: Experts Reject the Registration Document

The expert might reject the Registration, which means the Expert Review process is discontinued.

6.6. Step 6: Publication of the Registration Document

The authors are responsible for ensuring that the Registration Document is published according to "Specification Required" as defined in [RFC5226].

As set out in Section 3.4 it is strongly RECOMMENDED that Enumservice Specifications be published RFCs. As to every RFC, the normal IETF publication process applies (see [Instructions2authors]); i.e., the Registration Document is submitted in the form of an Internet Draft (e.g. via an IETF Working Group or a sponsoring Area Director). [Instructions2authors] also contains an option to publish an RFC as 'Independent Submission', which is further described in "Independent Submissions to the RFC Editor" [RFC4846].

6.7. Step 7: Adding Enumservice to the IANA Registry

In cases where the Registration Document is to be published as an RFC, the RFC publication process ensures that IANA will add the Enumservice to the registry.

In cases where the Registration Document is to be published in a specification other than RFC, the authors must inform IANA, as soon as the Enumservice Specification has been published according to "Specification Required" as defined in [RFC5226]. The <registrationdocs> element in the IANA Registration Template must contain an unambiguous reference to the Enumservice Specification (see also Section 5.2). In addition, the authors must provide IANA with a stable URL to the Enumservice Specification, in order that IANA may obtain the information included in the Enumservice Specification. IANA will then add the Enumservice to the registry.

7. Expert Review

7.1. Expert Selection Process

According to Section 3.2 of [RFC5226], experts are appointed by the IESG. The IESG is responsible for ensuring that there is always a sufficient pool of experts available.

7.2. Review Guidelines

Generally, the "Expert Review" process of an Enumservice follows the guidelines documented in Section 3.3 of "Guidelines for Writing an IANA Considerations Section in RFCs" [RFC5226]. Note that RFC 5226 says 'The review may be wide or narrow, depending on the situation and the judgment of the designated expert'. Therefore, the following list should be considered a guideline, rather than a binding list.

In case of conflicts between [RFC5226] and the guidelines in this section, [RFC5226] remains authoritative.

The expert evaluates the criteria as set out in [RFC5226], and should additionally consider the following:

- Verify conformance with the ENUM specification [RFC6116].
- Verify that the requirements set out in this document (Sections 3 and 5) are met. This includes checking for completeness and whether all the aspects described in Sections 3 and 5 are sufficiently addressed.
- If a use case is provided, the experts should verify whether the proposed Enumservice does actually match the use case. The experts should also determine whether the use case could be covered by an existing Enumservice.
- Verify that the Enumservice proposed cannot be confused with identical (or similar) other Enumservices already registered.
- If the Enumservice is classified according to Section 4.2, the experts must verify that the principles of the Class in question are followed.
- In case the Enumservice is not classified, the experts must verify whether a convincing reason for the deviation is provided in the Registration Document.

- Investigate whether the proposed Enumservice has any negative side effects on existing clients and infrastructure, particularly the DNS.

- If the output of processing an Enumservice might be used for input to more ENUM processing (especially services returning 'tel' URIs), the experts should verify that the authors have adequately addressed the issue of potential query loops.

7.3. Appeals

Appeals of Expert Review decisions follow the process described in Section 7 of [RFC5226] and Section 6.5 of [RFC2026].

8. Revision of Existing Enumservice Specifications

Many Enumservice registrations, published via IETF RFCs, already exist at the time of the development of this document. These existing Enumservice Specifications MAY be revised to comply with the specifications contained herein. All revisions of Enumservice Specifications MUST be compliant with the specifications contained herein.

Note: Enumservice Specifications updated only by [RFC6118] are not compliant with the specifications contained herein!

9. Extension of Existing Enumservice Specifications

There are cases where it is more sensible to extend an existing Enumservice registration rather than propose a new one. Such cases include adding a new Subtype to an existing Type. Depending on the nature of the extension, the original Enumservice Specification needs to be extended (Updates) or replaced (Obsoletes) [RFC2223]. Specifically, an update is appropriate when a new Subtype is being added without changes to the existing repertoire. A replacement is needed if there is a change to the default, or changes to the assumptions of URI support in clients.

Any Enumservice Specifications for existing Enumservices that are extended MUST comply with the specifications contained herein. As a consequence, revisions of existing Enumservice Specifications may be required according to Section 8.

10. Security Considerations

10.1. Considerations Regarding This Document

Since this document does not introduce any new technology, protocol, or Enumservice Specification, there are no specific security issues to be considered for this document. However, as this is a guide to authors of new Enumservice Specifications, the next section should be considered closely by authors and experts.

10.2. Enumservice Security Considerations Guideline

Guidelines concerning the Security Considerations section of an Enumservice Specification can be found in Section 5.6.

11. IANA Considerations

11.1. Registry Update

IANA updated the registry "Enumservice Registrations" as defined in (this) Section 11, which replaces the old mechanism as defined in [RFC3761].

It is noted that the process described herein applies only to ordinary Enumservice registrations (i.e., the registration process of "X-" Enumservices is beyond the scope of this document, and as per [RFC6116] "P-" Enumservices will not be registered at all).

11.2. Registration Template (XML chunk)

```
<record>
  <class> <!-- Enumservice Class --> </class>
  <type> <!-- Type --> </type>
  <subtype> <!-- Subtype --> </subtype>
  <urischeme> <!-- URI Schema Name --> </urischeme>
  <urischeme> <!-- another URI Schema Name --> </urischeme>
  <functionalspec>
    <paragraph>
      <!-- Text that explains the functionality of
           the Enumservice to be registered -->
    </paragraph>
  </functionalspec>
  <security>
      <!-- Security Considerations of the
           Enumservice to be registered -->
  </security>
  <usage> <!-- COMMON, LIMITED USE, or OBSOLETE --> </usage>
  <registrationdocs>
```

```
            <!-- Change accordingly -->
            <xref type="rfc" data="rfc2551"/>
          </registrationdocs>
          <requesters>
            <!-- Change accordingly -->
            <xref type="person" data="John_Doe"/>
            <xref type="person" data="Jane_Dale"/>
          </requesters>
          <additionalinfo>
            <paragraph>
              <!-- Text with additional information about
                   the Enumservice to be registered -->
            </paragraph>
            <artwork>
              <!-- There can be artwork sections, too -->
              :-)
            </artwork>
          </additionalinfo>
        </record>

       <people>
         <person id="John_Doe">
           <name> <!-- Firstname Lastname --> </name>
           <org> <!-- Organisation Name --> </org>
           <uri> <!-- mailto: or http: URI --> </uri>
           <updated> <!-- date format YYYY-MM-DD --> </updated>
         </person>
         <!-- repeat person section for each person -->
       </people>
```

Authors of an Enumservice Specification are encouraged to use these XML chunks as a template to create the IANA Registration Template. Examples for the use of this template are contained in Appendix A.

11.3. Location

Approved Enumservice registrations are published in the IANA registry named "Enumservice Registrations", which is available at the following URI: <http://www.iana.org/assignments/enum-services>.

This registry publishes representations derived from the IANA Registration Template as described in Section 11.2 and specified in Section 5.2.

Where the Enumservice Specification is not an RFC, IANA must hold an escrow copy of that Enumservice Specification. Said escrow copy will act as the master reference for that Enumservice registration.

11.4. Structure

IANA maintains the Enumservice Registry sorted in alphabetical order. The first sort field is Type, the second is Subtype.

[RFC6118] updates the existing Enumservices by transforming them into the new XML-chunk-based IANA Registration Template (see also Section 8).

11.5. Expert Review Procedure

Whenever a Registration Document is submitted via the IANA website, IANA will take care of the "Expert Review" process according to "Guidelines for Writing an IANA Considerations Section in RFCs" [RFC5226].

To prevent clashes, IANA will check whether a request with identical "type:subtype" (or "type" without Subtype) was submitted for Expert Review earlier and will inform the experts accordingly. The experts are authorized to resolve clashes as they see fit. The requesters may need to update their registration request before getting expert approval.

Once the experts have conditionally approved the Enumservice, IANA will inform the authors. This information should also include a reminder that (i) the authors are now responsible for publication of the Registration Document (see also Section 6.6) and (ii) the Enumservice will be added to the IANA registry only after its Enumservice Specification is published according to the "Specification Required" policy as defined in [RFC5226] (see also Section 6.7).

Note: After sending the approval note to the authors, IANA has no further responsibilities besides keeping internal records of approved Registration Documents. IANA will be involved again at registration of the Enumservice (see Section 11.6).

11.6. Registration Procedure

There is a slight difference in process depending on whether or not the Enumservice Specification will be published as an RFC. The reason for this difference lies in the current RFC publication process that includes IANA interaction shortly before publication of an RFC.

11.6.1. Published as an RFC

As per the RFC publication process, IANA will receive the Enumservice Specification to carry out IANA actions shortly before publication of the RFC. The IANA action will be to register the Enumservice, i.e., add the Enumservice to the IANA "Enumservice Registrations" registry (see also Section 11.3).

IANA must only add Enumservices to the Registry, if the experts have (conditionally) approved the corresponding Enumservice Specification. IANA should attempt to resolve possible conflicts arising from this together with the experts. In case there are substantial changes between the (conditionally) approved and the to be published version, IANA may reject the request after consulting the experts.

IANA must ensure that any further substantial changes the Enumservice Specification might undergo before final RFC publication are approved by the experts.

Note: Clearly editorial changes (such as typos) or minor changes in purely editorial sections (such as Authors' Addresses, Acknowledgments, References, and alike) are not considered substantial.

11.6.2. Published as a Non-RFC

Once the authors have informed IANA about the publication, IANA must ensure that the requirements for "Specification Required" as defined in [RFC5226] are met, the reference to the specification is unambiguous, and the content of the Enumservice Specification is identical to the Registration Document as approved by the experts. IANA will then register the Enumservice, i.e., add the Enumservice to the IANA "Enumservice Registrations" registry, and make an escrow copy (see also Section 11.3).

IANA must only add Enumservices to the Registry, if the experts have approved the corresponding Enumservice Specification. IANA should attempt to resolve possible conflicts arising from this together with the experts. In case there are substantial changes between the approved and the published version, IANA may reject the request after consulting the experts.

Note: Clearly editorial changes (such as typos) or minor changes in purely editorial sections (such as Authors' Addresses, Acknowledgments, References, and alike) are not considered substantial.

11.7. Change Control

Change control of any Enumservice registrations is done by "Specification Required", which implies the use of a Designated Expert, according to [RFC5226]. Updates of Enumservice Specifications MUST comply with the requirements described in this document. Updates are handled the same way as initial Enumservice registrations.

Authorized Change Controllers are the experts and the IESG.

Enumservice registrations must not be deleted. An Enumservice that is believed to be no longer appropriate for use can be declared deprecated by publication of a new Enumservice Specification, changing the Enumservice <usage> element (Intended Usage) to "DEPRECATED"; such Enumservices will be clearly marked in the lists published by IANA. As obsoletions are updates, they are also handled the same way as initial Enumservice registrations. Alternatively, Enumservices may be declared deprecated by an IESG action.

11.8. Restrictions

As stated in Section 3.2, a "-" (dash) MUST NOT be used as the first nor as the second nor as the last character of a Type or a Subtype. Furthermore, Type or Subtype of any Enumservice MUST NOT be set to, nor start with, "E2U". Any Enumservice registration requests not following these restrictions must be rejected by IANA, and the Expert Review process should not be initiated.

Section 5.2 contains examples for Enumservice registrations. Therefore, IANA must not register an Enumservice with Type or Subtype set to "foo", "bar", or "sbar", unless the experts explicitly confirm an exception.

12. Acknowledgments

The authors would like to thank the following people who have provided feedback or significant contributions to the development of this document: Jari Arkko, Stewart Bryant, Gonzalo Camarillo, Lawrence Conroy, Michelle Cotton, Miguel Garcia, David Harrington, Alfred Hoenes, Ari Keranen, Peter Koch, Edward Lewis, Alexey Melnikov, Jon Peterson, Pekka Savola, and Peter Saint-Andre.

Lawrence Conroy has provided extensive text for the Enumservice Classification section.

Section 3 of [RFC3761], which was edited by Patrik Faltstrom and Michael Mealling, has been incorporated into this document. Please see the Acknowledgments section in RFC 3761 for additional acknowledgments.

13. References

13.1. Normative References

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

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

[RFC3402] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Two: The Algorithm", RFC 3402, October 2002.

[RFC3403] Mealling, M., "Dynamic Delegation Discovery System (DDDS) Part Three: The Domain Name System (DNS) Database", RFC 3403, October 2002.

[RFC3761] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC6116] Bradner, S., Conroy, L., and K. Fujiwara, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 6116, March 2011.

13.2. Informative References

[ITU.E164.2005]
International Telecommunications Union, "The International Public Telecommunication Numbering Plan", ITU-T Recommendation E.164, Feb 2005.

[Instructions2authors]
Reynolds, J. and R. Braden, "Instructions to Request for Comments (RFC) Authors", RFC Editor http://www.rfc-editor.org/rfc-editor/instructions2authors.txt, August 2004.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2223] Postel, J. and J. Reynolds, "Instructions to RFC Authors", RFC 2223, October 1997.

[RFC2606] Eastlake, D. and A. Panitz, "Reserved Top Level DNS Names", BCP 32, RFC 2606, June 1999.

[RFC2780] Bradner, S. and V. Paxson, "IANA Allocation Guidelines For Values In the Internet Protocol and Related Headers", BCP 37, RFC 2780, March 2000.

[RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, July 2003.

[RFC3764] Peterson, J., "enumservice registration for Session Initiation Protocol (SIP) Addresses-of-Record", RFC 3764, April 2004.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

[RFC4238] Vaudreuil, G., "Voice Message Routing Service", RFC 4238, October 2005.

[RFC4759] Stastny, R., Shockey, R., and L. Conroy, "The ENUM Dip Indicator Parameter for the "tel" URI", RFC 4759, December 2006.

[RFC4846] Klensin, J. and D. Thaler, "Independent Submissions to the RFC Editor", RFC 4846, July 2007.

[RFC4969] Mayrhofer, A., "IANA Registration for vCard Enumservice", RFC 4969, August 2007.

[RFC4979] Mayrhofer, A., "IANA Registration for Enumservice 'XMPP'", RFC 4979, August 2007.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[RFC6118] Hoeneisen, B. and A. Mayrhofer, "Update of Legacy IANA Registrations of Enumservices", RFC 6118, March 2011.

Appendix A. IANA Registration Template Examples

This section contains non-normative examples of the XML-chunk-based IANA Registration Template:

This is the first example:

```
<record>
  <class>Protocol-Based</class>
  <type>email</type>
  <subtype>mailto</subtype>
  <urischeme>mailto</urischeme>
  <functionalspec>
    <paragraph>
      This Enumservice indicates that the resource
      can be addressed by the associated URI in
      order to send an email.
    </paragraph>
  </functionalspec>
  <security>
    See <xref type="rfc" data="rfc4355"/>, Section 6.
  </security>
  <usage>COMMON</usage>
  <registrationdocs>
    <xref type="rfc" data="rfc4355"/>
  </registrationdocs>
  <requesters>
    <xref type="person" data="Lawrence_Conroy"/>
  </requesters>
</record>

<people>
  <person id="Lawrence_Conroy">
    <name>Lawrence Conroy</name>
    <org>Siemens Roke Manor Research</org>
    <uri>mailto:lwc@roke.co.uk</uri>
    <updated>2008-11-20</updated>
  </person>
</people>
```

This is the second example.

```
<record>
  <class>Protocol-Based</class>
  <type>xmpp</type>
  <urischeme>xmpp</urischeme>
  <functionalspec>
    <paragraph>
      This Enumservice indicates that the
      resource identified is an XMPP entity.
    </paragraph>
  </functionalspec>
  <security>
    See <xref type="rfc" data="rfc4979"/>, Section 6.
  </security>
  <usage>COMMON</usage>
  <registrationdocs>
    <xref type="rfc" data="rfc4979"/>
  </registrationdocs>
  <requesters>
    <xref type="person" data="Alexander_Mayrhofer"/>
  </requesters>
</record>

<people>
  <person id="Alexander_Mayrhofer">
    <name>Alexander Mayrhofer</name>
    <org>enum.at GmbH</org>
    <uri>mailto:alexander.mayrhofer@enum.at</uri>
    <updated>2008-10-10</updated>
  </person>
</people>
```

This is the third example:

```
<record>
  <class>Application-Based</class>
  <type>voicemsg</type>
  <subtype>sip</subtype>
  <urischeme>sip</urischeme>
  <functionalspec>
    <paragraph>
      This Enumservice indicates that the resource
      identified can be addressed by the associated
      URI scheme in order to initiate a voice
      communication session to a voice messaging system.
    </paragraph>
  </functionalspec>
  <security>
    See <xref type="rfc" data="rfc4279"/>, Section 3.
  </security>
  <usage>COMMON</usage>
  <registrationdocs>
    <xref type="rfc" data="rfc4279"/>
  </registrationdocs>
  <requesters>
    <xref type="person" data="Jason_Livingood"/>
    <xref type="person" data="Donald_Troshynski"/>
  </requesters>
  <additionalinfo>
    <paragraph>
      Implementers should review a non-exclusive list of
      examples in <xref type="rfc" data="rfc4279"/>,
      Section 7.
    </paragraph>
  </additionalinfo>
</record>

<people>
  <person id="Jason_Livingood">
    <name>Jason Livingood</name>
    <org>Comcast Cable Communications</org>
    <uri>mailto:jason_livingood@cable.comcast.com</uri>
    <updated>2008-11-20</updated>
  </person>
```

```
            <person id="Donald_Troshynski">
              <name>Donald Troshynski</name>
              <org>Acme Packet</org>
              <uri>mailto:dtroshynski@acmepacket.com</uri>
              <updated>2008-11-20</updated>
            </person>
          </people>
```

In the third IANA Registration Template example above, the "voicemsg" Enumservice is used. This Enumservice actually has several Subtypes, and one of those is shown in the example. For each Subtype, an individual Registration Template must be submitted to IANA, so that an Enumservice with several Subtypes will have several corresponding IANA Registration Templates. This is to avoid any ambiguity of the relation between <subtype> and <urischeme> elements.

Appendix B. Changes from RFC 3761

This section lists the changes applied to the Enumservice registration process and the IANA registry definition, compared to RFC 3761.

- While RFC 3761 required "Standards track or Experimental" RFCs for an Enumservice to be registered, this document mandates "Specification Required", which implies the use of a Designated Expert.

- This document defines the classification of Enumservices. The IANA Registration Template has been complemented to contain a <class> element (Enumservice Class).

- A new element <registrationdocs> (Enumservice Specification) has been added to the IANA Registration Template.

- The former field "Any other information that the author deems interesting" of the IANA Registration Template turned into the <additionalinfo> element (Further Information).

- The Enumservice "Name" field has been removed from the IANA Registration Template.

- The Registration Template is now a chunk of XML data, reflecting IANA’s recent work to convert registries to XML.

Authors' Addresses

Bernie Hoeneisen
Ucom Standards Track Solutions GmbH
CH-8000 Zuerich
Switzerland

Phone: +41 44 500 52 44
EMail: bernie@ietf.hoeneisen.ch (bernhard.hoeneisen AT ucom.ch)
URI: http://www.ucom.ch/

Alexander Mayrhofer
enum.at GmbH
Karlsplatz 1/9
Wien A-1010
Austria

Phone: +43 1 5056416 34
EMail: alexander.mayrhofer@enum.at
URI: http://www.enum.at/

Jason Livingood
Comcast Cable Communications
One Comcast Center
1701 John F. Kennedy Boulevard
Philadelphia, PA 19103
USA

Phone: +1-215-286-7813
EMail: jason_livingood@cable.comcast.com
URI: http://www.comcast.com/