The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)
<draft-ietf-enum-3761bis-09.txt>

Abstract

   This document discusses the use of the Domain Name System (DNS) for
   storage of data associated with E.164 numbers, and for resolving
   those numbers into URIs that can be used (for example) in telephony
   call setup. This document also describes how the DNS can be used to
   identify the services associated with an E.164 number. This document
   obsoletes RFC3761.

Table of Contents

1. Introduction

   This document discusses the use of the Domain Name System (DNS,
   [RFC1034],[RFC1035]) for storage of data associated with E.164
   [E.164] numbers, and for resolving those numbers into URIs that can
   be used (for example) in telephony call setup. This document also
   describes how the DNS can be used to identify the services associated
   with an E.164 number. This document includes a Dynamic Delegation
   Discovery System (DDDS) Application specification, as detailed in the
   document series described in [RFC3401]. This document obsoletes
   [RFC3761].

   Using the process defined in this document, International Public
   Telecommunication Numbers in the international format defined in
   International Telecommunications Union (ITU) Recommendation E.164
   [E.164] (called here "E.164 numbers") can be transformed into DNS
   names. Using existing DNS services (such as delegation through NS
   records and queries for NAPTR resource records), one can look up the
   services associated with that E.164 number. This takes advantage of
   standard DNS architectural features of decentralized control and
   management of the different levels in the lookup process.

   The domain "e164.arpa" has been assigned to provide an infrastructure
   in the DNS for storage of data associated with E.164 numbers. To
   facilitate distributed operations, this domain is divided into
   subdomains. Holders of E.164 numbers who want the numbers to be
   listed in the DNS should contact the appropriate zone administrator
   as listed in the policy attached to the zone. One should start
   looking for this information by examining the SOA resource record
   associated with the zone, just like in normal DNS operations.

   Of course, as with other domains, policies for such listings will be
   controlled on a subdomain basis and may differ in different parts of
   the world.

1.1. 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 BCP 14, RFC 2119
   [RFC2119].

   DNS Resource Record types mentioned in this document are defined,
   respectively, in [RFC1035] (NS, SOA, A, MX), [RFC3403] (NAPTR), and
   [RFC2782] (SRV).

   All other capitalized terms are taken from the vocabulary found in
   the DDDS algorithm specification found in [RFC3402].

2. Use of These Mechanisms for Private Dialling Plans

Similar mechanisms might be used for other kinds of digit strings
(such as numbers in private dialling plans). If these mechanisms are
used for dialling plans (or for other unrelated digit strings), the
domain apex used for such translation MUST NOT be e164.arpa, to avoid
conflict with this specification.

Also, the Application Unique String (see Section 3.1) used with
dialling plans SHOULD be the full number as specified, without the
leading '+' character. The '+' character is used to further
distinguish E.164 numbers in international format from dialled digit
strings or other digit sequences.

   For example, to address the E.164 number +44-3069-990038 a user
   might dial "03069990038" or "00443069990038" or "011443069990038".
   These dialled digit strings differ from one another, but none of
   them start with the '+' character.

Finally, if these techniques are used for dialling plans or other
digit strings, implementers and operators of systems using these
techniques for such purpose MUST NOT describe these schemes as
"ENUM". The initial "E" in ENUM stands for E.164, and the term
"ENUM" is used exclusively to describe application of these
techniques to E.164 numbers according to this specification.

3. The ENUM Application Specifications

This template defines the ENUM DDDS Application according to the
rules and requirements found in [RFC3402]. The DDDS database used by
this Application is found in [RFC3403], which is the document that
defines the NAPTR DNS Resource Record type.

ENUM is designed as a way to translate from E.164 numbers to URIs
using NAPTR records stored in DNS. The First Well Known Rule for any
ENUM query creates a key (a fully qualified domain name, or FQDN,
within the e164.arpa domain apex) from an E.164 number. This FQDN is
queried for NAPTR records and returned records are processed and
interpreted according to this specification.

3.1. Application Unique String

The Application Unique String (AUS) is a fully qualified E.164 number
minus any non-digit characters except for the '+' character which
appears at the beginning of the number. The '+' is kept to provide a
well understood anchor for the AUS in order to distinguish it from
other telephone numbers that are not part of the E.164 namespace.

For example, the E.164 number could start out as "+44-116-496-0348".
To ensure that no syntactic sugar is allowed into the AUS, all non-
digits except for '+' are removed, yielding "+441164960348".

3.2. First Well Known Rule

The First Well Known Rule converts an Application Unique String (AUS)
into an initial key. That key is used as an index into the
Application's Rules Database. For ENUM, the rules database is the
DNS, so the key is a fully qualified domain name (FQDN).

In order to convert the AUS to a unique key in this database the
string is converted into a domain name according to this algorithm:

1. Remove all characters with the exception of the digits. For
    example, given the E.164 number "+44-20-7946-0148" (which would
    then have been converted into an AUS of "+442079460148"), this
    step would simply remove the leading '+', producing
    "442079460148".
2. Reverse the order of the digits. Example: "841064970244"
3. Put dots ('.') between each digit. Example:
    "8.4.1.0.6.4.9.7.0.2.4.4"
4. Append the string ".e164.arpa." to the end and interpret as a
    domain name. Example: 8.4.1.0.6.4.9.7.0.2.4.4.e164.arpa.

The E.164 namespace and this Application's database are organized in
such a way that it is possible to go directly from the name to the
smallest granularity of the namespace directly from the name itself,
so no further processing is required to generate the initial key.

This domain name is used to request NAPTR records. Each of these
records may contain the end result or, if its flags field is empty,
produces a new key in the form of a domain name that is used to
request further NAPTR records from the DNS.

3.3. Expected Output

The output of the last DDDS loop is a Uniform Resource Identifier in
its absolute form according to the <absolute-URI> production in the
Collected ABNF found in [RFC3986].

3.4. Valid Databases

At present only one DDDS Database is specified for this Application.
"Dynamic Delegation Discovery System (DDDS) Part Three: The DNS
Database" [RFC3403] specifies a DDDS Database that uses the NAPTR DNS
resource record to contain the rewrite rules. The Keys for this
database are encoded as domain names.

The character set used for the substitution expression is UTF-8
[RFC3629]. The allowed input characters are all those characters
that are allowed anywhere in an E.164 number. The characters allowed
to be in a Key are those that are currently defined for DNS domain
names.

3.4.1. Optional Name Server Additional Section Processing

Some nameserver implementations attempt to be intelligent about items
that are inserted into the additional information section of a given
DNS response. For example, BIND will attempt to determine if it is
authoritative for a domain whenever it encodes one into a packet. If
it is, then it will insert any A records it finds for that domain
into the additional information section of the answer until the
packet reaches the maximum length allowed. It is therefore
potentially useful for a client to check for this additional
information.

It is also easy to contemplate an ENUM enhanced nameserver that
understands the actual contents of the NAPTR records it is serving
and inserts more appropriate information into the additional
information section of the response. Thus, DNS servers MAY interpret
flag values and use that information to include appropriate resource
records in the Additional Information section of the DNS packet.
Clients are encouraged to check for additional information but are
not required to do so. See Section 4.2 of [RFC3403] ("Additional
Information Processing") for more information on NAPTR records and
the Additional Information section of a DNS response packet.

3.4.2. Flags

This Database contains a field that contains flags that signal when
the DDDS algorithm has finished. At this time only one flag, "U", is
defined. This means that this Rule is the last one and that the
output of the Rule is a URI [RFC3986]. See Section 4.3 of [RFC3404].

If a client encounters a resource record with an unknown flag, it
MUST ignore it and move to the next Rule. This test takes precedence
over any ordering since flags can control the interpretation placed
on fields.

A novel flag might change the interpretation of the Regexp and/or
Replacement fields such that it is impossible to determine if a
resource record matched a given target.

If this flag is not present then this rule is non-terminal. If a
Rule is non-terminal then the result produced by its rewrite rule
MUST be a FQDN. Clients MUST use this result as the new Key in the

DDDS loop (i.e., the client will query for NAPTR resource records at
this FQDN).

3.4.3. Service Parameters

Service Parameters for this Application take the following Augmented
Backus-Naur Form (ABNF, specified in [RFC5234]) and are found in the
Services field of the NAPTR record that holds a terminal rule. Where
the NAPTR holds a non-terminal Rule, the Services field SHOULD be
empty, and clients SHOULD ignore its content.

```
    service-field = "E2U" 1*(servicespec)
    servicespec   = "+" enumservice
    enumservice   = type 0*(subtypespec)
    subtypespec   = ":" subtype
    type          = 1*32(ALPHA / DIGIT / "-")
    subtype       = 1*32(ALPHA / DIGIT / "-")
```

In other words, a non-optional "E2U" (used to denote ENUM only
Rewrite Rules in order to mitigate record collisions) followed by one
or more Enumservices which indicate the class of functionality a
given end point offers. Each Enumservice is indicated by an initial
'+' character.

3.4.3.1. ENUM Services

Enumservices may be specified and registered via the process defined
in "IANA Registration of Enumservices: Guide, Template and IANA
Considerations" [SV_GUIDE]. This registration process is not open to
any Enumservice that has '-' as the second character in its type
string.

In particular, this registration process is not open to Enumservice
types starting with the facet "X-". This "X-" facet is reserved for
experimental or trial use, and any such Enumservices cannot be
registered using the normal process.

Finally, any Enumservice type that starts with the facet "P-" is
intended for use exclusively on private networks. As such, NAPTRs
containing Enumservice types starting "P-" should not be seen on the
global Internet. Even if an ENUM client recognizes and can engage in
the Enumservice, it may be incapable of resolving the URI generated
by the containing NAPTR. These Enumservices WILL NOT be registered.

Such Enumservices MUST NOT be provisioned in any system that provides
answers to DNS queries for NAPTR resource record sets (RRSets) from
entities outside the private network context in which these

Enumservices are intended for use. Unless an ENUM client is sure
that it is connected to the private network for which these NAPTRs
are provisioned and intended, it MUST discard any NAPTR with an
Enumservice type that starts with the "P-" facet.

3.4.3.2. Compound NAPTRs and Implicit ORDER/PREFERENCE Values

It is possible to have more than one Enumservice associated with a
single NAPTR. These Enumservices share the same Regexp field and so
generate the same URI. Such a "compound" NAPTR could well be used to
indicate a mobile phone that supports both "voice:tel" and "sms:tel"
Enumservices.
The Services field in that case would be "E2U+voice:tel+sms:tel".

A compound NAPTR can be treated as a set of NAPTRs that each hold a
single Enumservice. These reconstructed NAPTRs share the same ORDER
and PREFERENCE/PRIORITY field values but should be treated as if each
had a logically different priority. A left-to-right priority is
assumed.

3.5. The ENUM Algorithm Always Returns a Single Rule

The ENUM algorithm always returns a single rule. Individual
applications may have application-specific knowledge or facilities
that allow them to present multiple results or speed selection, but
these should never change the operation of the algorithm.

3.6. Case Sensitivity in ENUM

Case sensitivity was not mentioned at all in [RFC3761] (or
[RFC2916]), but has been seen as an issue during interoperability
test events since then. There are a lot of case-sensitive clients in
current deployment.

The only place where NAPTR field content is case sensitive is in any
static text in the Repl sub-field of the Regexp field (see Section
3.2 of [RFC3402] for Regexp field definitions). In that sub-field,
case must be preserved when generating the record output. Elsewhere,
case sensitivity is not used.

Where ENUM clients can be exposed to NAPTR records that may hold
field content of different capitalisation, clients MUST use case-
insensitive processing. ENUM clients that operate using the Internet
to send their queries, typically called "Public ENUM" scenarios, fall
into this category.

Some ENUM clients operate within closed networks; for example, within
isolated data networks operated by Communication Service Providers.

These are typically called "Infrastructure ENUM" scenarios. All
zones provisioned within such closed networks usually have a known
capitalisation for ENUM record string content, as provisioning
systems for such networks are often carefully controlled. In such an
environment, clients are never exposed to records with capitalisation
that is "unexpected" and so can be (and have been) designed with case
sensitive processing. Only if a client is known to operate in an
environment in which capitalisation of all ENUM records it will
encounter is known and controlled MAY that client use case sensitive
processing.

3.7. Collision Avoidance

An ENUM-compliant application MUST only pass numbers to the ENUM
client query process that it believes are E.164 numbers (e.g. it MUST
NOT pass dialled digit strings to the ENUM query process).

Since number plans may change over time, it can be impossible for a
client to know if the number it intends to query is assigned and
active within the current number plan. Thus it is important that
such clients can distinguish data associated with the E.164 number
plan from that associated with other digit strings (i.e. numbers NOT
according to the E.164 number plan).

It is the responsibility of operators provisioning data into domains
to ensure that data associated with a query on an E.164 number cannot
be mistaken for data associated with other uses of NAPTRs.

Three techniques are used to achieve this:

o   the domain apex used for purposes other than data associated with
    the E.164 number plan MUST NOT be e164.arpa.
o   for use other than with E.164 numbers, the Application Unique
    String MUST NOT begin with the '+' character, whilst for ENUM use,
    the AUS MUST begin with this character.
o   NAPTRs that are intended for other DDDS applications MUST NOT
    include the E2U token in their service field, whilst NAPTRs
    intended for ENUM use MUST include this token.

4. ENUM Service Example

```
  $ORIGIN 3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa.
   NAPTR 100 50 "u" "E2U+sip"
       "!^(\\+441632960083)$!sip:\\1@example.com!"    .
   NAPTR 100 51 "u" "E2U+h323"
       "!^\\+441632960083$!h323:operator@example.com!"    .
   NAPTR 100 52 "u" "E2U+email:mailto"
```

          "!^.*$!mailto:info@example.com!"   .

   This describes that the domain 3.8.0.0.6.9.2.3.6.1.4.4.e164.arpa. is
   preferably contacted by SIP, secondly via H.323 for voice, and
   thirdly by SMTP for messaging. Note that the Enumservice tokens
   "sip", "h323", and "email" are Enumservice Types registered with
   IANA, and they have no implicit connection with the protocols or URI
   schemes with the same names.

   In all cases, the next step in the resolution process is to use the
   resolution mechanism for each of the protocols (specified by the URI
   schemes sip, h323 and mailto) to know what node to contact.

   In each of the first two records, the ERE sub-field matches only
   queries that have been made for the telephone number +441632960083.
   In the last record, the ERE matches any Application Unique String
   value. The first record also demonstrates how the matched pattern
   can be used in the generated URI.

   Note that where NAPTR resource records are shown in DNS master file
   syntax (as in this example above), each backslash must itself be
   escaped using a second backslash. The DNS on-the-wire packet will
   have only a single backslash in each case.

5. Clarification of DDDS Use in ENUM

   ENUM is a DDDS Application. This means that it relies on the DDDS
   for its operation. DDDS is designed to be flexible, but that opens
   the possibility of differences of interpretation. This section is
   intended to cover ENUM-specific interpretation of text within the
   DDDS specifications. The goal is to ensure interoperability between
   ENUM clients and provisioning systems used to populate domains with
   E2U NAPTRs.

   As part of on-going development work on the ENUM specifications, an
   analysis has been carried out into the way in which ENUM client and
   provisioning system implementations behave, and the interoperability
   issues that have arisen. This (informative) analysis is provided in
   [RFC5483]. The following recommendations reflect that analysis, and
   further narrative explaining the issues can be found there.

5.1. Collected Implications for ENUM Provisioning

   ENUM NAPTRs SHOULD NOT include characters outside the printable US-
   ASCII equivalent range (U+0020 to U+007E) unless it is clear that all
   ENUM clients they are designed to support will be able to process
   such characters correctly. If ENUM zone provisioning systems require
   non-ASCII characters, these systems MUST encode the non-ASCII data to

emit only US-ASCII characters by applying the appropriate mechanism
(such as those in [RFC3492], [RFC3987]). Non-printable characters
SHOULD NOT be used, as ENUM clients may need to present NAPTR content
in a human-readable form.

The case-sensitivity flag ('i') is inappropriate for ENUM, and SHOULD
NOT be provisioned into the Regexp field of E2U NAPTRs.

The Registrant and the ENUM zone provisioning system he or she uses
SHOULD NOT rely on ENUM clients solely taking account of the value of
the ORDER and the PREFERENCE/PRIORITY fields in ENUM NAPTRs. Thus, a
Registrant SHOULD place into his or her zone only contacts that he or
she is willing to support; even those with the worst ORDER and
PREFERENCE/PRIORITY values MAY be selected by an end user.

All E2U NAPTRs SHOULD hold a default value in their ORDER field. A
value of "100" is recommended, as it seems to be used in most
provisioned domains.

   Some ENUM clients have been known to pre-discard NAPTRs within an
   RRSet simply because these records do not have the lowest ORDER
   value found in that RRSet. Other ENUM client implementations
   appear to have confused ORDER and PREFERENCE/PRIORITY fields,
   using the latter as the major sort term rather than the former as
   specified. Conversely, ENUM zones have been provisioned within
   which the ORDER value varies but the PREFERENCE/PRIORITY field
   value is static. This may have been intentional, but given the
   different client behaviour in the face of varying ORDER field
   values, may not produce the desired response.

Multiple NAPTRs with identical ORDER and identical PREFERENCE/
PRIORITY field values SHOULD NOT be provisioned into an RRSet unless
the intent is that these NAPTRs are truly identical and there is no
preference between them. Implementers SHOULD NOT assume that the DNS
will deliver NAPTRs within an RRSet in a particular sequence.

An ENUM zone provisioning system SHOULD assume that, if it generates
compound NAPTRs, the Enumservices will normally be processed in left-
to-right order within such NAPTRs.

ENUM zone provisioning systems SHOULD assume that, once a non-
terminal NAPTR has been selected for processing, the ORDER field
value in a domain referred to by that non-terminal NAPTR will be
considered only within the context of that referenced domain (i.e.,
the ORDER value will be used only to sort within the current RRSet
and will not be used in the processing of NAPTRs in any other RRSet).

ENUM zone provisioning systems SHOULD use '!' (U+0021) as their

Regexp delimiter character.

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). Note that when a NAPTR resource record is entered in DNS master file syntax, the backslash itself must be escaped using a second backslash.

If present in the ERE sub-field of an ENUM NAPTR, the literal character '+' MUST be escaped as "\+" (i.e. U+005C U+002B). Note that, as always, when a NAPTR resource record is entered in DNS master file syntax, the backslash itself must be escaped using a second backslash.

Whilst this client behaviour is non-compliant, ENUM provisioning systems and their users should be aware that some ENUM clients have been detected with poor (or no) support for non-trivial ERE sub-field expressions.

ENUM provisioning systems SHOULD be cautious in the use of multiple back-reference patterns in the Repl sub-field of NAPTRs they provision. Some clients have limited buffer space for character expansion when generating URIs. These provisioning systems SHOULD check the back-reference replacement patterns they use, ensuring that regular expression processing will not produce excessive-length URIs.

ENUM zones MUST NOT be provisioned with NAPTRs according to the obsolete syntax of [RFC2916], and MUST be provisioned with NAPTRs in which the Services field is according to Section 3.4.3 of this document.

   [RFC2915] and [RFC2916] have been obsoleted by [RFC3401]-[RFC3404]
   and by this document, respectively.

Enumservices in which the Enumservice type starts with the facet "P-" MUST NOT be provisioned in any system that provides answers to DNS queries for NAPTR resource record sets from entities outside the private network context in which these Enumservices are intended for use.

As current support is limited, non-terminal NAPTRs SHOULD NOT be provisioned in ENUM zones unless it is clear that all ENUM clients that this environment supports can process these.

When populating a set of domains with NAPTRs, ENUM zone provisioning systems SHOULD NOT configure non-terminal NAPTRs so that more than 5 such NAPTRs will be processed in an ENUM query.

In a non-terminal NAPTR that may be encountered in an ENUM query
(i.e., one with an empty Flags field), the Services field SHOULD be
empty.

A non-terminal NAPTR MUST include its target domain in the (non-
empty) Replacement field as this field will be interpreted as holding
the FQDN that forms the next key output from this non-terminal rule.
The Regexp field MUST be empty in a non-terminal NAPTR intended to be
encountered during an ENUM query.

5.2. Collected Implications for ENUM Clients

If a NAPTR is discarded, this SHOULD NOT cause the whole ENUM query
to terminate and processing SHOULD continue with the next NAPTR in
the returned RRSet.

ENUM clients SHOULD NOT discard NAPTRs in which they detect
characters outside the US-ASCII printable range (0x20 to 0x7E
hexadecimal).

ENUM clients MAY discard NAPTRs that have octets in the Flags,
Services, or Regexp fields that have byte values outside the US-ASCII
equivalent range (i.e., byte values above 0x7F). Clients MUST be
ready to encounter NAPTRs with such values without failure.

ENUM clients MUST sort the records of a retrieved NAPTR RRSet into
sequence using the ORDER and PREFERENCE fields of those records. The
ORDER is to be treated as the major sort term, with lowest numerical
values being earlier in the sequence. The PREFERENCE/PRIORITY field
is to be treated as the minor sort term, with lowest numerical values
being earlier in the sequence.

ENUM clients SHOULD NOT discard a NAPTR record until it is considered
or a record previous to it in the evaluation sequence has been
accepted.

   Notably, if a record has a "worse" ORDER value than others in this
   RRSet, that record MUST NOT be discarded before consideration
   unless a record has been accepted as the result of this ENUM
   query.

Where the ENUM client presents a list of possible URLs to the end
user for his or her choice, it MAY present all NAPTRs -- not just the
ones with the lowest currently unprocessed ORDER field value. The
client SHOULD observe the ORDER and PREFERENCE/PRIORITY values
specified by the Registrant.

ENUM clients SHOULD accept all NAPTRs with identical ORDER and

identical PREFERENCE/PRIORITY field values, and process them in the
sequence 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 might have done this already).

ENUM clients SHOULD consider the ORDER field value only when sorting
NAPTRs within a single RRSet. The ORDER field value SHOULD NOT be
taken into account when processing NAPTRs across a sequence of DNS
queries created by traversal of non-terminal NAPTR references.

ENUM 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.

ENUM clients MUST be ready to process NAPTRs that use a different
character from '!' as their Regexp Delimiter without failure.

ENUM clients SHOULD NOT assume that the delimiter is the last
character of the Regexp field.

   Unless they are sure that in their environment this is the case,
   in general an ENUM client may still encounter NAPTRs that have
   been provisioned with a following 'i' (case-insensitive) flag,
   even though that flag has no effect at all in an ENUM scenario.

ENUM clients SHOULD discard NAPTRs that have more or less than 3
unescaped instances of the delimiter character within the Regexp
field.

   In the spirit of being liberal with what it will accept, if the
   ENUM client is sure how the Regexp field should be interpreted, it
   MAY choose to process the NAPTR even in the face of an incorrect
   number of unescaped delimiter characters. If it is not clear how
   the Regexp field should be interpreted, the client MUST discard
   the NAPTR.

ENUM clients MUST be ready to process NAPTRs that have non-trivial
patterns in their ERE sub-field values without failure.

ENUM clients MUST be ready to process NAPTRs with many copies of
back-reference patterns within the Repl sub-field without failure.

ENUM clients MUST be ready to process NAPTRs with a DDDS Application
identifier other than 'E2U' without failure.

When an ENUM client encounters a compound NAPTR (i.e., one containing
more than one Enumservice) and cannot process or cannot recognize one

of the Enumservices within it, that ENUM client SHOULD ignore this
Enumservice and continue with the next Enumservice within this
NAPTR's Services field, discarding the NAPTR only if it cannot handle
any of the Enumservices contained. These conditions SHOULD NOT be
considered errors.

ENUM clients MUST support ENUM NAPTRs according to syntax defined in
Section 3.4.3. ENUM clients SHOULD also support ENUM NAPTRs
according to the obsolete syntax of [RFC2916]; there are still zones
that hold "old" syntax NAPTRs. The informational [RFC3824]
recommended such support.

Unless an ENUM client is sure that it is connected to the private
network for which these NAPTRs are provisioned and intended, it MUST
discard any NAPTR with an Enumservice type that starts with the "P-"
facet.

5.2.1. Non-terminal NAPTR Processing

ENUM clients MUST be ready to process NAPTRs with an empty Flags
field ("non-terminal" NAPTRs) without failure. More generally, non-
terminal NAPTR processing SHOULD be implemented, but ENUM clients MAY
discard non-terminal NAPTRs they encounter.

ENUM clients SHOULD ignore any content of the Services field when
encountering a non-terminal NAPTR with an empty Flags field.

ENUM clients receiving a non-terminal NAPTR with an empty Flags field
MUST treat the Replacement field as holding the FQDN to be used in
the next round of the ENUM query. An ENUM client MUST discard such a
non-terminal NAPTR if the Replacement field is empty or does not
contain a valid FQDN. By definition, it follows that the Regexp
field will be empty in such a non-terminal NAPTR. If present in a
non-terminal NAPTR, a non-empty Regexp field MUST be ignored by ENUM
clients.

If a problem is detected when processing an ENUM query across
multiple domains (by following non-terminal NAPTR references), the
ENUM query SHOULD NOT be abandoned, but instead processing SHOULD
continue at the next NAPTR after the non-terminal NAPTR that referred
to the domain in which the problem would have occurred.

If all NAPTRs in a domain traversed as a result of a reference in a
non-terminal NAPTR have been discarded, the ENUM client SHOULD
continue its processing with the next NAPTR in the "referring" RRSet
(i.e., the one including the non-terminal NAPTR that caused the
traversal).

ENUM clients MUST be prepared to encounter a referential loop in
which a sequence of Non-Terminal NAPTRs are retrieved within an ENUM
query that refer back to an earlier FQDN. ENUM clients MUST be able
to detect and recover from such a loop, without failure.

ENUM clients MAY consider a chain of more than 5 "non-terminal"
NAPTRs traversed in a single ENUM query as an indication that a
referential loop has been entered.

When a domain is about to be entered as the result of a reference in
a non-terminal NAPTR, and the ENUM client has detected a potential
referential loop, the client SHOULD discard the non-terminal NAPTR
from its processing and continue with the next NAPTR in its list. It
SHOULD NOT make the DNS query indicated by that non-terminal NAPTR.

6. IANA Considerations

RFC 2916 and then RFC 3761 (which this document replaces) requested
IANA to delegate the E164.ARPA domain following instructions that
were provided by the IAB (as described in [RFC3245]). The domain was
delegated according to those instructions (which are published at
<http://www.ripe.net/enum/instructions.html>).

Names within this zone are to be delegated to parties consistent with
ITU Recommendation E.164. The names allocated should be hierarchic
in accordance with ITU Recommendation E.164, and the codes should be
assigned in accordance with that Recommendation.

The IAB is to coordinate with ITU Telecommunications Standardization
Bureau (TSB) if the technical contact for the domain e164.arpa is to
change, as ITU TSB has an operational working relationship with this
technical contact which would need to be reestablished.

See [SV_GUIDE] for Enumservice-related IANA Considerations.

7. Security Considerations

7.1. DNS Security

As ENUM uses DNS, which in its current form is an insecure protocol,
there is no mechanism for ensuring that the data one gets back is
authentic. As ENUM is deployed on the global Internet, it is
expected to be a popular target for various kind of attacks, and
attacking the underlying DNS infrastructure is one way of attacking
the ENUM service itself.

There are multiple types of attacks that can happen against DNS that
ENUM implementations should consider. See Threat Analysis of the

Domain Name System [RFC3833] for a review of the various threats to
the DNS.

Because of these threats, a deployed ENUM service SHOULD include
mechanisms to mitigate these threats. Most of the threats can be
solved by verifying the authenticity of the data via mechanisms such
as DNS Security (DNSSEC) [RFC4033].

Others, such as Denial-Of-Service attacks, cannot be solved by data
authentication. It is important to remember that these threats
include not only the NAPTR lookups themselves, but also the various
records needed for the services to be useful (for example NS, MX, SRV
and A records).

Even if DNSSEC is deployed, it cannot protect against every kind of
attack on DNS. ENUM is often used for number or address translation;
retrieving an address through an ENUM lookup with DNSSEC support does
not, however, ensure that the service is imune to attack. It is
unwise for a service blindly to trust that the address it has
retrieved is valid and that the entity to which it connects using
that address is the service peer it intended to contact. A service
SHOULD always authenticate the entity to which it connects during the
service setup phase, and not rely on address or identity data
retrieved outside that service.

Finally, as an ENUM service will be implementing some type of
security mechanism, software which implements ENUM MUST be prepared
to receive DNSSEC and other standardized DNS security responses,
including large responses and other EDNS0 signalling (see [RFC2671]),
unknown resource records (see [RFC3597]), and so on.

7.2. Caching Security

The DNS architecture makes extensive use of cacheing of records at
intermediary nodes to improve performance. The propagation time (for
changes to resource records to be reflected in query responses to end
nodes) approaches the "time to live" value for those records. There
may be a number of different resource records involved in the
resolution of a communication target. Changes to these records may
not be synchronised (particularly if these resource records indicate
different times to live). Thus a change in any one of these records
may cause inappropriate decisions on communications targets to be
made. Given that DNS Update (specified in [RFC2136]) can introduce
quite rapid changes in content in different zones, these transient
states may become important.

Consider a typical set of queries that follow an ENUM query that
returns a SIP URI (for details, see [RFC3263]):

o Evaluation of the SIP URI triggers a query on the SIP domainpart
   for D2U/D2T NAPTRs.
o This in turn triggers a query on that records's target domain for
   SRV records.
o The SRV records will return the SIP server hostname, which will
   trigger a further query on that hostname for an A record to get
   the server's associated IP address.
o Finally, the local SIP User Agent Client will then attempt to
   initiate a communications session to that IP address.

The E2U NAPTR may have changed its URI, indicating a new SIP
identity. The D2U NAPTR for the SIP URI domainpart may have changed
its target. The SRV record pointed to by that D2U NAPTR may have
changed its target hostname. The hostname's A record may have
changed its IP address. Finally, if the server exists in an
environment where IP-addresses are dynamically assigned (for example,
when using DHCP [RFC2131]), an unexpected end point may have been
allocated to the IP address returned from the SIP resolution chain.

In environments where changes to any of the chain of resource records
or dynamic assignments to IP addresses occur, those systems
provisioning this data SHOULD take care to minimise changes and to
consider the respective times to live of resource records and/or DHCP
lease times. Users of this data SHOULD take care to detect and
recover from unintended communications session attempts; in a
transient environment, these may occur.

7.3. Call Routing Security

There are a number of countries (and other numbering environments) in
which there are multiple providers of call routing and number/name-
translation services. In these areas, any system that permits users,
or putative agents for users, to change routing or supplier
information may provide incentives for changes that are actually
unauthorized (and, in some cases, for denial of legitimate change
requests). Such environments should be designed with adequate
mechanisms for identification and authentication of those requesting
changes and for authorization of those changes.

7.4. URI Resolution Security

A large amount of Security Issues have to do with the resolution
process itself, and use of the URIs produced by the DDDS mechanism.
Those have to be specified in the registration of the Enumservice
used, as specified in "IANA Registration of Enumservices: Guide,
Template and IANA Considerations" [SV_GUIDE].

8. Acknowledgements

This document is an update of RFC 3761, which was edited by Patrik
Faltstrom and Michael Mealling. Please see the Acknowledgements
section in that RFC for additional acknowledgements. The authors
would also like to thank Alfred Hoenes and Bernie Hoeneisen for their
detailed reviews.

9. Changes from RFC 3761

A section has been added to explain the way in which DDDS is used
with this specification. These recommendations have been collected
from experience of ENUM deployment. Differences of interpretation of
the DDDS specifications led to interoperability issues; this document
updates RFC 3761 to add many clarifications, intended to ameliorate
interoperability.

Clarifications include a default value for the ORDER field and for
the Regexp delimiter character, required use of Replacement field in
non-terminal NAPTRs, and that string matching is case insensitive
(Section 3.6).

Other substantive changes include removing the discussion of
registration mechanisms, (now specified in "IANA Registration of
Enumservices: Guide, Template and IANA Considerations" [SV_GUIDE]),
correcting an existing error by adding "-" as a valid character in
the type and subtype fields specified in Services Parameters
(Section 3.4.3) and adding the "P-" private service type
(Section 3.4.3.1).

9.1. Draft Change Log

change log - RFC Editor - please remove this section for publication.

 version 01 -> 02
   clean up English - many places
   removed Registration mechanism for Enumservices section
   removed IANA considerations - point to draft-ietf-enum-enumservices-
   guide,
   replace DNS Security Threats in section 6.1 with pointer to RFC 3833
   fold in text from the ENUM Experiences ID - many places

 version 02 -> 03
   fixed minor typos
   revised section 2.4.4.1, added P-
   expanded IANA Considerations - Section 6

 version 03 -> 04
   Many changes to bring into sync with RFC 5483

 version 04 -> 05
   change "ameliorate" to "mitigate" in 7.1
   fix reference to IANA Registration of Enumservices
   clarify ENUM definition of DDDS First Well Known Rule

 version 05 -> 06
   fix language in section 2.2
   add Alfred Hoenes to the acknowledgements section

 version 06 -> 07
   major re-structuring, collecting together provisioning and client
    recommendations into a their own sub-sections within a new section
    and referring to analysis in RFC 5483 rather than copying text
    directly
   corrected typos in the "Collected Implications" items
   slight rewording of section 1 introductory text and matching abstract
   rewording in section 1.2 to clarify goal
   create a separate sub-section for collision avoidance at the end of
    section 2 (to replace the second paragraph at the start of this
    specification section).
   add Bernie Hoeneisen to the acknowledgements section

  version 07 -> 08
   Renumbered sections to pull out the old section 1.2 as its own
    section.
   Added at end of section 4.1 -- "The informational RFC3824 recommended
    such support."
   Changed phrasing of sentence in introduction paragraph 3 to be
    consistent with abstract and paragraph 1.
   Corrected spelling in Acknowledgements.

  version 08 -> 09
   Response to IESG Evaluation:
    Correction of error/typo in 3.2 step 3
    Extensive clarification of 3.6 & addition of MUST requirement
    Clarification of 5.1 & change of SHOULD to MUST
    Clarification of 7.1 & SHOULD for service authentication made
     explicit
    Extensive clarification of 7.2 & SHOULD actions made explicit
    Minor clarification text in section 1, section 1.1, 3, 3.2 step 1,
     3.4, 3.4.2, 3.4.3, and 6.
    Addition of missing references
    Slight simplification of IANA section and added ref to RFC3245

[10](#). References

[10.1](#). Normative References

[E.164]   ITU-T, "The International Public Telecommunication Number
             Plan", Recommendation E.164, February 2005.

[RFC1034]   Mockapetris, P., "Domain names - concepts and
             facilities", STD 13, RFC 1034, November 1987.

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

[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.

[RFC3404]   Mealling, M., "Dynamic Delegation Discovery System (DDDS)
             Part Four: The Uniform Resource Identifiers (URI)",
             RFC 3404, October 2002.

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

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

[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.

[RFC3987]   Duerst, M. and M. Suignard, "Internationalized Resource
             Identifiers (IRIs)", RFC 3987, January 2005.

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

[10.2](#). Informative References

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

   [RFC2131]   Droms, R., "Dynamic Host Configuration Protocol",
                RFC 2131, March 1997.

   [RFC2136]   Vixie, P., Thomson, S., Rekhter, Y., and J. Bound,
                "Dynamic Updates in the Domain Name System (DNS UPDATE)",
                RFC 2136, April 1997.

   [RFC2671]   Vixie, P., "Extension Mechanisms for DNS (EDNS0)",
                RFC 2671, August 1999.

   [RFC2782]   Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for
                specifying the location of services (DNS SRV)", RFC 2782,
                February 2000.

   [RFC2915]   Mealling, M. and R. Daniel, "The Naming Authority Pointer
                (NAPTR) DNS Resource Record", RFC 2915, September 2000.

   [RFC2916]   Faltstrom, P., "E.164 number and DNS", RFC 2916,
                September 2000.

   [RFC3245]   Klensin, J. and IAB, "The History and Context of
                Telephone Number Mapping (ENUM) Operational Decisions:
                Informational Documents Contributed to ITU-T Study Group
                2 (SG2)", RFC 3245, March 2002.

   [RFC3263]   Rosenberg, J. and H. Schulzrinne, "Session Initiation
                Protocol (SIP): Locating SIP Servers", RFC 3263,
                June 2002.

   [RFC3401]   Mealling, M., "Dynamic Delegation Discovery System (DDDS)
                Part One: The Comprehensive DDDS", RFC 3401,
                October 2002.

   [RFC3597]   Gustafsson, A., "Handling of Unknown DNS Resource Record
                (RR) Types", RFC 3597, September 2003.

   [RFC3824]   Peterson, J., Liu, H., Yu, J., and B. Campbell, "Using
                E.164 numbers with the Session Initiation Protocol
                (SIP)", RFC 3824, June 2004.

   [RFC3833]   Atkins, D. and R. Austein, "Threat Analysis of the Domain
                Name System (DNS)", RFC 3833, August 2004.

   [RFC4033]   Arends, R., Austein, R., Larson, M., Massey, D., and S.
                Rose, "DNS Security Introduction and Requirements",
                RFC 4033, March 2005.

   [RFC5483]   Conroy, L. and K. Fujiwara, "ENUM Implementation Issues
                and Experiences", RFC 5483, March 2009.

   [SV_GUIDE]   Hoeneisen, B., Mayrhofer, A., and J. Livingood, "IANA
                Registration of Enumservices: Guide, Template and IANA
                Considerations", draft-ietf-enum-enumservices-guide (work
                in progress), April 2010.

Authors' Addresses

   Scott Bradner
   Harvard University
   29 Oxford St.
   Cambridge MA 02138
   USA

   Phone: +1-617-495-3864
   EMail: sob@harvard.edu

   Lawrence Conroy
   Roke Manor Research
   Roke Manor
   Old Salisbury Lane
   Romsey
   United Kingdom

   Phone: +44-1794-833666
   EMail: lconroy@insensate.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 101-0165
   JAPAN

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