URNBIS
Internet-Draft
Obsoletes: 2141, 3406 (if approved)
Intended status: Standards Track
Expires: September 10, 2015

P. Saint-Andre
&yet
J. Klensin

March 9, 2015

Uniform Resource Names (URNs)
draft-ietf-urnbis-rfc2141bis-urn-10

Abstract

A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) that is assigned under the "urn" scheme and a particular URN namespace, typically with the intent that the URN will be a persistent, location-independent resource identifier or abstract designator. With regard to URN syntax, this document defines the canonical syntax for URNs (in a way that is consistent with URI syntax), specifies methods for determining URN equivalence, and discusses URI conformance. With regard to URN namespaces, this document specifies a method for defining a URN namespace and associating it with a namespace identifier, and describes procedures for registering namespace identifiers with the Internet Assigned Numbers Authority (IANA). This document obsoletes both RFC 2141 and RFC 3406.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on September 10, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction 3
2. Terminology 4
3. URN Syntax 5
  3.1. Namespace Identifier Syntax 5
  3.2. Namespace Specific String Syntax 5
  3.3. p-component, q-component, and f-component 6
4. Equivalence of URNs 8
  4.1. Procedure 8
  4.2. Examples 9
5. URI Conformance 10
6. URN Namespaces 11
  6.1. Formal Namespaces 13
  6.2. Informal Namespaces 15
7. Defining a URN Namespace 15
  7.1. Purpose 16
  7.2. Syntax 16
  7.3. Assignment 17
  7.4. Security and Privacy 18
  7.5. Resolution 18
8. Registering a URN Namespace 19
  8.1. Formal Namespaces 19
  8.2. Informal Namespaces 19
9. Guidelines for Designated Experts 20
10. IANA Considerations 20

10.1. URI Scheme 20
10.2. Registration of URN Namespaces 21
11. Security and Privacy Considerations 21
12. References 22
12.1. Normative References 22
12.2. Informative References 22
Appendix A. Registration Template 24
A.1. Namespace ID 24
A.2. Version 24
A.3. Date 24
A.4. Registrant 24
A.5. Purpose 24
A.6. Syntax 24
A.7. Assignment 25
A.8. Resolution 25
A.9. Documentation 25
A.10. Revision Information 25
Appendix B. Changes from RFC 2141 25
Appendix C. Changes from RFC 3406 25
Appendix D. Contributors 26
Appendix E. Acknowledgements 26
Appendix F. Change log for versions of draft-ietf-urnbis-rfc2141bis-urn 26
F.1. Changes from -08 to -09 26
F.2. Changes from -09 to -10 27
Authors' Addresses 27

1. Introduction

[[CREF1: Note on Draft -10: the editors had a choice between posting this version in a stable, but early, form and waiting until the last day for comment from the mailing list that might affect it. We chose the latter but, since there have been few comments, the draft is a little bit more rough in places, and contains more questions, than we would have liked. Notes marked "WG:" are in particular need of attention by the working group. References to "Henry" in those notes are to various comments from Henry S. Thompson, so of which did not appear on the mailing list. Similarly, reference to "Ted" are to Ted Hardie's on-list messages of 3 March and discussions of them. We hope those issues can be resolved on the mailing list or during IETF 92 and a new, cleaner, version posted at the end of the month.]]

A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) [RFC3986] that is assigned under the "urn" scheme and a particular namespace, typically with the intent that the URN will be a persistent, location-independent resource identifier or abstract designator.

The assignment of URNs is done by an organization (or, in some cases, according to an algorithm or other automated process) that has been formally delegated a namespace within the "urn" scheme (e.g., a URN in the 'example' namespace [RFC6963] might be of the form "urn:example:foo").

This document rests on two key assumptions:

1. Assignment of a URN is a managed process.

2. The space of URN namespaces is itself managed.

While other schemes may allow identifiers to be freely chosen and assigned, this is not the case for URNs. The syntactical correctness of a string appearing after "urn:" is not sufficient to make it a URN; both the namespace identifier and namespace specific string must be registered or generated according to the rules given for it to be a valid URN.

So that information about both URN syntax and URN namespaces is available in one place, this document does the following:

1. Defines the canonical syntax for URNs in general (in a way that is consistent with URI syntax), specifies methods for determining URN equivalence, and discusses URI conformance.

2. Specifies a method for defining a URN namespace and associating it with a namespace identifier, and describes procedures for registering namespace identifiers with the Internet Assigned Numbers Authority (IANA).

For URN syntax and URN namespaces, this document modernizes and replaces the definitions from [RFC2141] and [RFC3406]. These modifications build on the requirements provided in [RFC1737] and many years of experience with URNs, in both cases attempting to make the smallest reasonable set of changes from the previous definitions.

This document obsoletes both [RFC2141] and [RFC3406].

2. Terminology

Several important terms used in this document are defined in the URI specification [RFC3986].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. URN Syntax

The syntax of URNs as provided in [RFC2141] was defined before the updated specification of URIs in [RFC3986]. To ensure consistency with the URI syntax as well as semantic flexibility in the use of URNs within particular applications (see [I-D.ietf-urnbis-semantics-clarif] for further discussion), this specification extends the syntax of URNs to explicitly allow several characters (and thus URI components) that were not allowed by [RFC2141], and also makes several smaller syntax adjustments.

As a result, the syntax for a URN is defined as follows using the Augmented Backus-Naur Form (ABNF) as specified in [RFC5234]. Rules not defined below (i.e., alphanum, pchar, path-absolute, query, and fragment) are defined in [RFC3986].

```
namestring    = assigned-name
                [ p-component ]
                [ q-component ]
                [ f-component ]
assigned-name = "urn" ":" NID ":" NSS
              ; the URI scheme ("urn") is case insensitive
NID           = (alphanum) 0*30(ldh) (alphanum)
ldh           = alphanum / "-"
NSS           = 1*(pchar)
p-component   = "/" path-absolute
q-component   = "?" query
f-component   = "#" fragment
```

Note that "?" can be used without %-encoding inside q-components and f-components.

The following sections provide additional information about these rules.

3.1. Namespace Identifier Syntax

The syntax here is slightly more restrictive than what was defined in [RFC2141], since it forbids the character "-" at the end of a NID.

NIDs are case insensitive (e.g., "ISBN" and "isbn" are equivalent).

3.2. Namespace Specific String Syntax

[[CREF2: -09 text follows; see below:]] Depending on the rules governing a namespace, names that are valid in a namespace might contain characters that are not allowed in URNs according to the "pchar" rule (e.g., characters outside the ASCII range or characters

that are reserved in URIs, such as "/", "?", and "#"). Such a string MUST be translated into a conformant NSS before using it as a protocol element or otherwise passing it on to other applications. Translation is done by percent-encoding each disallowed character using the method defined in Section 2.1 of [RFC3986]. Note that the "%" character is allowed only for the purpose of percent-encoding.
[[CREF3: WG: note Ted's suggested replacement text (email of 2015-03-06 and and below:]]
[[CREF4: Edited version of Ted's proposal for replacement text -- the WG needs to decide whether to use it and, if so to check it carefully:]]
Depending on the rules governing a namespace, names that are valid in a namespace might contain characters that are not found in the "pchar" production above (e.g., characters outside the ASCII range or characters that are reserved in URIs, such as "/", "?", and "#"). While these may be valid names, they are not valid URNs until those strings are translated into a conformant NSS. Translation is done by percent-encoding each disallowed character using the method defined in Section 2.1 of the generic URI specification [RFC3986] Note that the "%" character is allowed only for the purpose of percent-encoding.

In order to make URNs as stable and persistent as possible when protocols evolve and the environment around them changes, namespaces SHOULD NOT allow characters outside the basic Latin repertoire [RFC20] unless the nature of the particular namespace makes such characters necessary.

If a namespace designates one or more characters conforming to the "pchar" rule as having special meaning for that namespace (e.g., "@") and the namespace also uses that character in a literal sense, when used in a literal sense the character MUST be percent-encoded (e.g., "%40"). For related considerations with regard to NID registration, see below.

3.3. p-component, q-component, and f-component

The p-component, q-component, and f-component are optional components that follow the assigned-name. In terms of URI syntax these components are essentially equivalent to the URI "path-absolute", "query", and "fragment" constructions, respectively. However, the URN p-component, q-component, and f-component need not be semantically equivalent to the URI path component, query component, and fragment component; therefore they are called by different names in this specification.

Unless specifically defined for a particular namespace after publication of this document, use of these components is disallowed,

thereby maintaining strict backward compatibility with namespaces defined in accordance with [RFC2141] and registered in accordance with [RFC3406].

This specification does not define the semantics of the p-component, q-component, and f-component for URNs in general. Instead, additional specifications might establish these matters for URN-related services (such as URN resolution) or for individual URN namespaces (e.g., to handle extended information about the resource identified by a URN). For example, it is possible that the q-component might be used in requests to URN resolution services, or that the f-component might be used to distinguish the integral parts of resources named by URNs in particular namespaces (say, the chapters of a book). However, defining such usage is the responsibility of specifications for URN resolution services, namespace registration requests and specifications for individual namespaces, and other appropriate documentation (such as policy documents governing the management of a given URN namespace).

As general guidance that might not apply to all cases, it would be inappropriate for namespaces that do not intend to support resolution services to allow q-components. Namespaces that deal with digital manifestations might be able to support f-components. [[CREF5: WG: What, if anything, should be said about p-components here?]]

3.3.1. p-component

The only formal restriction placed upon a p-component by this specification is that the syntax SHALL adhere to the "path-absolute" rule from [RFC3986]. The inner syntax of a p-component is to be defined by the specification for a particular namespace or URN-related service. (For example, a namespace specification might define a character such as "~" or "@" as a delimiter inside p-components assigned within that namespace.)

As described under Section 4, the p-component SHALL be taken into account when determining URN equivalence.

3.3.2. q-component

The only formal restriction placed upon a q-component by this specification is that the syntax SHALL adhere to the "query" rule from [RFC3986] (prepended by the "?" character). The inner syntax of a q-component is to be defined by the specification for a particular namespace. (For example, a namespace specification might define a character such as ";" or "=" as a delimiter inside q-components assigned within that namespace.)

As described under Section 4, the q-component SHALL NOT be taken into account when determining URN equivalence.

3.3.3. f-component

The only formal restriction placed upon an f-component by this specification is that the syntax SHALL adhere to the "fragment" rule from [RFC3986] (prepended by the "#" character). The inner syntax of an f-component is to be defined by the specification for a particular namespace. (For example, a namespace specification might define a character such as "&" or "+" as a delimiter inside f-components assigned within that namespace.)

As described under Section 4, the f-component SHALL NOT be taken into account when determining URN equivalence.

4. Equivalence of URNs

4.1. Procedure

For various purposes such as caching, often it is desirable to determine if two URNs are "the same". This is done by testing for equivalence (see Section 6.1 of [RFC3986]).

The generic URI specification [RFC3986] is very flexible about equality comparisons, putting the focus on allowing false negatives and avoiding false positives. If comparisons are made in a scheme-independent way, i.e., as URI comparisons only, URNs that this specification considers equal would be rejected. The discussion below applies when the URIs involved are known to be URNs.

Two URNs are equivalent if they are octet-by-octet equal after applying case normalization (as specified in Section 6.2.2.1 of [RFC3986]) to the following constructs:
[[CREF6: WG: See a variety of notes about "case normalization" and related issues, e.g., the tracker entry https://github.com/MelindaShore/urnbis/issues/3]]

1. the URI scheme "urn"

2. the NID

3. any percent-encoded characters in the NSS (that is, all character triplets in the NSS that match the <pct-encoding> production found in Section 2.1 of the base URI specification [RFC3986]).

Percent-encoded characters MUST NOT be decoded, i.e., percent-encoding normalization (as specified in Section 6.2.2.2 of [RFC3986]) MUST NOT be applied.

[[CREF7: WG: Ted's note of 2015-03-06 recommends adding p-component here somewhere. If we do so, we need to be sure that the new text doesn't accidentally make them case-insensitive again.]]

If a q-component or f-component (or both) are included in a URN, they MUST be ignored for purposes of determining equivalence.

URN namespace definitions may include additional rules for equivalence, such as case-insensitivity of the NSS (or parts thereof). Such rules MUST always have the effect of eliminating some of the false negatives obtained by the procedure above and MUST NOT result in treating two URNs as not equivalent if the procedure here says they are equivalent. For related considerations with regard to NID registration, see below.

4.2. Examples

The following six URN comparisons (which use the "example" NID defined in [RFC6963]) highlight the equivalence rules:

1. URN:example:a123,456

2. urn:example:a123,456

3. urn:EXAMPLE:a123,456

4. urn:example:a123%2C456

5. URN:EXAMPLE:a123%2c456

6. urn:example:A123,456

7. urn:example:a123,456/789

8. urn:example:a123,456/abc

9. urn:example:a123,456?789

10. urn:example:a123,456?abc

11. urn:example:a123,456#789

12. urn:example:a123,456#abc

URNs 1, 2, 3, 9, 10, 11, and 12 are equivalent. URNs 4 and 5 are equivalent only to each other. URNs 6, 7, and 8 are not equivalent to any of the others.

[[CREF8: Placeholder: We should still add some examples with p-, q-, and/or f-components to the above list.]]

5. URI Conformance

Because a URN is, syntactically, a URI under the "urn" scheme, in theory a URN can be placed in any protocol slot that allows for a URI (e.g., an XML namespace name [XML-NAMES]). However, this does not imply that, semantically, it always makes sense in practice to place a URN in a given URI protocol slot; in particular, because a URN might not specify the location of a resource or even point indirectly to one, it might not be appropriate to place a URN in a URI protocol slot that points to a resource (examples include the 'href' and 'src' attributes and the <base/> element in HTML, as well as the 'xml:base' attribute in XML [XML-BASE]). Ultimately, specifications of where it is appropriate to use URNs, or URNs created within particular URN namespaces, are the responsibility of descriptions of individual URI schemes and their expected uses; this specification cannot possibly anticipate all of the relevant cases.
[[CREF9: Ted suggests "...individual URI contexts;"]]

Despite the fact that URNs are not hierarchical and are not appropriate for use as a base URI (see Section 5.1 of [RFC3986]), the relative resolution algorithm specified in Section 5.2 of [RFC3986] still applies to the "urn" URI scheme; implementers need to be aware, however, that running the algorithm against URNs will lead to results that might be unexpected or not useful.

A resolver that conforms to the URI specification [RFC3986] will extract a scheme of "urn" rather than a scheme value of "urn:<nid>".
[[CREF10: WG: Henry has pointed out that this spec really cannot constrain generic URI resolvers (or processors more generally. Is the following rewrite of 'A URN MUST be considered an opaque URI by URI resolvers and passed (with the "urn" scheme) to a URN resolver for resolution.' sufficient.]]
In part because of the separation of semantics from syntax [I-D.ietf-urnbis-semantics-clarif], generic URI processors must pay special attention to the parsing and analysis rules of RFC 3986 and, in particular, must treat the URI as opaque unless the scheme and its requirements are recognized. The URN resolver can either be an external resolver that the URI resolver knows of, or it can be functionality built into the URI resolver. Note that this requirement MAY impose constraints on the contexts in which URNs are appropriately used; see the previous section.

To minimize user confusion, a URI browser SHOULD display the complete URN (including the "urn" scheme and any components) to ensure that there is no confusion between URN namespace identifiers and URI scheme identifiers. For example, a URI beginning with "urn:xmpp:" [RFC4854] is very different from a URI beginning with "xmpp:" [RFC5122]. Similarly, a potential DOI scheme [DOI-URI] is different from, and possibly completely unrelated to, a possible DOI URN namespace.

When URNs are transported and exchanged, they MUST be represented in this format. Further, all URN-aware applications MUST offer the option of displaying URNs in this canonical form to allow for direct transcription (for example by cut and paste techniques). Such applications might support display of URNs in a more human-friendly form and might use a character set that includes characters that are not permitted in URN syntax as defined in this specification (e.g., when displaying URNs to humans, such applications might replace percent-encoded strings with characters from an extended character repertoire such as that of [UNICODE]).

As mentioned, the assignment of URNs is a managed process, as is the assignment of namespaces themselves. Although design of the URNs to be assigned within a given namespace is ceded by this specification to the namespace owner, doing so in a managed way avoids the problems inherent in unmanaged generation of URIs as described in the recommendations regarding URI design and ownership [RFC7320].

6. URN Namespaces

A URN namespace is a collection of identifiers that obey three constraints. Such a namespace is (1) unique, (2) assigned in a consistent way, and (3) assigned according to a common definition.

1. The "uniqueness" constraint means that an identifier within the namespace is never assigned to more than one resource and never reassigned to a different resource, even if the identifier itself is deprecated or becomes obsolete.

2. The "consistent assignment" constraint means that an identifier within the namespace is assigned by an organization or created in accordance with a process or algorithm that is always followed.

3. The "common definition" constraint means that there are clear definitions for the syntax of identifiers within the namespace and for the process of assigning or creating them.

A URN namespace is identified by a particular NID in order to ensure the global uniqueness of URNs and, optionally, to provide a cue regarding the structure of URNs assigned within a namespace.

With regard to global uniqueness, using different NIDs for different collections of identifiers ensures that no two URNs will be the same for different resources, since each collection is required to uniquely assign each identifier. However, a single resource can have more than one URN assigned to it for different purposes (e.g., some numbers might be valid identifiers in two different identifier systems, where the namespace identifier differentiates between the resulting URNs). Subject to other constraints, such as those imposed by the URI syntax [RFC3986], the rules of the URN scheme are intended to allow preserving the normal and natural form of identifiers specified elsewhere and treated as URN namespaces.

[[CREF11: WG: Ted suggested adding a concrete example, "as this seems likely to be open to confusion. The RFC 5031 emergency services may provide one (e.g. urn:service:counseling.mental-health and urn:service:counseling.suicide may be the same resource), but any concrete example would likely be an improvement." Is that example ok? Would anyone like to suggest other text or an additional example?]]

With regard to the structure of URNs assigned within a namespace, the development of an identifier structure (and thereby a collection of identifiers) depends on the requirements of the community defining the identifiers, how the identifiers will be assigned and used, etc. These issues are beyond the scope of URN syntax and the general rules for URN namespaces, because they are specific to the community defining a namespace (e.g., the bibliographic and publishing communities in the case of the 'ISBN' and 'ISSN' namespaces, or the developers of extensions to the Extensible Messaging and Presence Protocol in the case of the 'XMPP' namespace).

URN namespaces inherit certain rights and responsibilities by the nature of URNs, e.g.:

1. They uphold the general principles of a well-managed URN namespace by providing persistent identification of resources and unique assignment of identifier strings.

2. They can be registered in global registration services.

There are two types of URN namespace: formal and informal. These are distinguished by the expected level of service, the information needed to define the namespace, and the procedures for registration.

Because the majority of the namespaces registered so far have been formal, this document concentrates on formal namespaces.

Note: [RFC3406] defined a third type of "experimental namespaces", denoted by prefixing the namespace identifier with the string "X-". Consistent with [RFC6648], this specification removes the experimental category. Because experimental namespaces were never registered, removing the experimental category has no impact on the existing registries or future registration procedures. Because they are not registered, strings that refer to existing experimental namespaces are not valid URNs. Truly experimental usages can, of course, employ the 'example' namespace [RFC6963].

6.1. Formal Namespaces

A formal namespace provides benefit to some subset of users on the Internet. In particular, it would not make sense for a formal namespace to be used only by a community or network that is not connected to the Internet. For example, it would be inappropriate for a NID to effectively force someone to use a proprietary network or service not open to the general Internet user. The intent is that, while the community of those who might actively use the names assigned within that NID might be small, the potential use of identifiers within that NID is open to any user on the Internet. Formal NIDs might be appropriate even when some aspects are not fully open. For example, a namespace might make use of a fee-based, privately managed, or proprietary registry for assignment of URNs in the namespace. However, it might still benefit some Internet users if the associated services have openly-published access protocols.

An organization that will assign URNs within a formal namespace ought to meet the following criteria:

1. Organizational stability and the ability to maintain the URN namespace for a long time; absent such evidence, it ought to be clear how the namespace can remain viable if the organization can no longer maintain the namespace.

2. Competency in name assignment. This will improve the likelihood of persistence (e.g. to minimize the likelihood of conflicts).

3. Commitment to not reassigning existing names and to allowing old names to continue to be valid, even if the owners or assignees of those names are no longer members or customers of that organization. With regard to URN resolution [RFC2276], this does not mean that there needs to be resolution of such names, only that the names will not resolve to false or stale information.

[[CREF12: WG: Henry says "Really? If a piece of retrievable metadata for some URN is the name and (postal) address of the publisher of whatever it identifies, and they've gone out of business, that metadata is necessarily stale, isn't it?" How do you think this should be handled?]]

A formal namespace establishes a particular NID, subject to the following constraints (above and beyond the syntax rules already specified):

1. It MUST NOT be an already-registered NID.

2. It MUST NOT start with "urn-" (which is reserved for informal namespaces).

3. It MUST be more than two characters long.

4. It MUST NOT start with "aa-", where "aa" is any combination of two ASCII letters and the hyphen is followed by something other than another hyphen.

5. It MUST NOT start with the string "xn--" or any other string consisting of two letters followed by two hyphens. Those strings are reserved for potential representation of DNS A-labels and similar strings in the future [RFC5890].
   [[CREF13: Note in Draft: IDNA reserves the entire collection of strings starting with two characters followed by two hyphens.]]

6. [[CREF14: WG: Ted also points out that proposed NIDs that are, or resemble, ISO 3166-1 alpha-3 codes may need special consideration in expert review becuase some countries claim ownership of them (and, at least implicitly, their use by anyone other than that country could be deceptive. On a related and more general note, Henry inquired whether we need dispute resolution and/or trademark procedures. The WG needs to decide whether to address those issues and, if so, how.]]

All two-letter strings, and all two-letter strings followed by "-" and any sequence of valid NID characters, are reserved for potential use as NIDs based on ISO alpha-2 country codes [ISO3166] for eventual national registrations of URN namespaces. The definition and scoping of rules for allocation of responsibility for such country-code-based namespaces is beyond the scope of this document.

6.2. Informal Namespaces

Informal namespaces are full-fledged URN namespaces, with all the associated rights and responsibilities. Informal namespaces differ from formal namespaces in the process for assigning a NID: for an informal namespace, the registrant does not designate the NID; instead, IANA assigns a NID consisting of the string 'urn-' followed by one or more digits (e.g., "urn-7") where the digits consist of the next available number in the sequence of positive integers assigned to informal namespaces. Thus the syntax of an informal namespace is:

```
InformalNamespaceName = "urn-" Number
Number                = DigitNonZero 0*Digit
DigitNonZero          = "1"/ "2" / "3" / "4"/ "5"
                      / "6" / "7" / "8" / "9"
Digit                 = "0" / DigitNonZero
```

The only restrictions on <Number> are that it (1) consist strictly of ASCII digits, that it (2) not have leading zeros, and that it (3) not cause the NID to exceed the length limitations defined for the URN syntax.

7. Defining a URN Namespace

The material in this section is intended as guidance to both designers of URN namespaces and (expert) reviewers of proposed URN namespace registrations. More information for the latter appears in Section 9.
[[CREF15: WG: Henry would like to see a SHOULD requirement on the definitions here. The WG needs to decide whether that is necessary and/or desirable.]]

The definition of a formal namespace ought to pay particular attention to:

1. The purpose of the namespace.

2. The syntax of URNs assigned within the namespace, including whether p-, q-, and/or f-components are allowed.

3. The process for assigning URNs within the namespace.

4. The security implications of assigning URNs within the namespace and using the assigned URNs.

5. Optionally, the process for resolving URNs issued within the namepace.

The following sections explain these matters in greater detail. For convenience, a template for defining and registering a URN namespace is provided under Appendix A. This information can be especially helpful to entities that wish to request assignment of a URN in a namespace and to entities that wish to provide URN resolution for a namespace.

7.1. Purpose

The "Purpose" section of the template describes matters such as:

1. The kinds of resources identified by URNs assigned within the namespace.

2. Why it is preferable to use URNs rather than some other technology (e.g., separate URI schemes or URIs in existing schemes) and why no existing URN namespace is a good fit.

3. The kinds of software applications that can use or resolve the assigned URNs (e.g., by differentiating among disparate namespaces, identifying resources in a persistent fashion, or meaningfully resolving and accessing services associated with the namespace).

4. The scope of the namespace (public vs. private, global vs. local to a particular organization, nation, or industry). For example, a namespace claiming to deal in "national identification numbers" ought to have a global scope and address all identity number structures, whereas a URN scheme for a particular national identification number system would need to handle only the structure for that nation's identity numbers.

5. How the intended community (and the Internet community at large) will benefit from using or resolving the assigned URNs.

7.2. Syntax

The "Syntax" section of the template contains:

1. A description of the structure of URNs within the namespace, in conformance with the fundamental URN syntax. The structure might be described in terms of a formal definition (e.g., using Augmented BNF for Syntax Specifications (ABNF) as specified in [RFC5234]), an algorithm for generating conformant URNs, or a regular expression for parsing the identifier into components; alternatively, the structure might be opaque.

2. Any special character encoding rules for assigned URNs (e.g., which character ought to always be used for single-quotes).

3. If p-components, q-components, and/or f-components are allowed for the namespace, a discussion of how they are used.

4. Rules for determining equivalence between two identifiers in the namespace. Such rules ought to always have the effect of eliminating false negatives that might otherwise result from comparison. If it is appropriate and helpful, reference can be made to specific equivalence rules defined in the URI specification [RFC3986]. Examples of equivalence rules include equivalence between uppercase and lowercase characters in the Namespace Specific String, between hyphenated and non-hyphenated groupings in the identifier string, or between single-quotes and double-quotes. (Note that these are not normative statements for any kind of best practice related to handling of equivalences between characters in general; they are statements limited to one particular namespace only.)

5. Any special considerations necessary for conforming with the URN syntax. This is particularly applicable in the case of existing naming systems that are used in the context of URNs. For example, if a namespace is used in contexts other than URNs, it might make use of characters that are reserved in the URN syntax. This section ought to note any such characters, and outline necessary mappings to conform to URN syntax. Normally, this will be handled by percent-encoding the character as specified in the URI specification [RFC3986].

7.3. Assignment

The "Assignment" section of the template describes matters such as:

1. Mechanisms or authorities for assigning URNs to resources. It ought to make clear whether assignment is completely open (e.g., following a particular procedure such as first-come, first-served (FCFS)), completely closed (e.g., for a private organization), or limited in various ways (e.g., delegated to authorities recognized by a particular organization); if limited, it ought to explain how to become an assigner of identifiers or how to request assignment of identifiers from existing assignment authorities.

2. Methods for ensuring that URNs within the namespace are unique. For example, identifiers might be assigned sequentially or in accordance with some well-defined process by a single authority, assignment might be partitioned among delegated authorities that

are individually responsible for respecting uniqueness rules, or URNs might be created independently following an algorithm that itself guarantees uniqueness.

7.4. Security and Privacy

The "Security" section of the template describes any potential issues related to security and privacy with regard to assignment, use, and resolution of identifiers within the namespace. Examples of such issues include:

- o The consequences of producing false negatives and false positives during comparison for equivalence (see "Issues in Identifier Comparison for Security Purposes" [RFC6943])

- o Leakage of private information when identifiers are communicated on the public Internet

- o The potential for directory harvesting

- o Various issues discussed in the guidelines for security considerations in RFCs [RFC3552] and the privacy considerations for Internet protocols [RFC6973].

7.5. Resolution

The "Resolution" section specifies the rules for resolution of URNs assigned within the namespace. If such URNs are intended to be resolvable, the namespace needs to be registered in a Resolution Discovery System (RDS, see [RFC2276]) such as DDDS. Resolution then proceeds according to standard URI resolution processes, as well as the mechanisms of the RDS. This section ought to list the requirements for becoming a recognized resolver of URNs in the relevant namespace (and being so listed in the RDS registry). Answers might include, but are not limited to:

[[CREF16: WG: Possible alternate paragraph instead of above, see email from Klensin, 2015-03-04, subject: "Resolution services and Section 7.5 of draft-ietf-urnbis-rfc2141bis-urn-09": The "Resolution" section specifies the rules for resolution of URNs assigned within the namespace. This section SHOULD list the requirements for becoming a recognized resolver of URNs in the relevant namespace. Answers might include, but are not limited to:]]

1. The namespace is not listed with an RDS; therefore this section is not applicable.

2. Resolution mirroring is completely open, with a mechanism for updating an appropriate RDS.

3. Resolution is controlled by entities to which assignment has been delegated.

8. Registering a URN Namespace

[[CREF17: WG: As pointed out on the mailing list (message from Klensin, 4 March 10:16 -0500, subject: "Expert reveiw and draft-ietf-urnbis-rfc2141bis-urn-09"), we've had several discussions that suggest a somewhat different Expert Review model than is covered by RFC 5226. Those discussions haven't gotten quite far enough to let us generate specific text, so the WG has work to do in this area. That email message makes a suggestion.]]

8.1. Formal Namespaces

The registration policy for formal namespaces is Expert Review as defined in the "IANA Considerations" document [RFC5226] and in Section 9 below. The key steps for registration of a formal namespace are:

1. Fill out the namespace registration template (see Appendix A). This can be done as part of an Internet-Draft or a specification in another series, although that is not necessary.

2. Send the completed template to the urn-nid@ietf.org discussion list for review.

3. If necessary to address comments received, repeat steps 1 and 2.

4. If the designated experts approve the request, the IANA will register the requested NID.

A formal namespace registration can be revised by updating the registration template, following the same steps outlined above for new registrations. A revised registration MUST describe differences from prior versions and SHOULD make special note of any relevant changes in the underlying technologies or namespace management processes.

8.2. Informal Namespaces

The registration policy for informal namespaces is First Come First Served [RFC5226]. The key steps for registration of an informal namespace are:

1. Write a completed namespace definition template (see Appendix A).

2. Send it to the urn-nid@ietf.org discussion list for feedback.

3. Once the review period has expired, send the final template to IANA (via the iana@iana.org email address).

An informal namespace registration can be revised by updating the registration template, following the same steps outlined above for new registrations.

9. Guidelines for Designated Experts

Experience to date with NID registration requests has shown that registrants sometimes do not initially understand some of the subtleties of URN namespaces, and that defining the namespace in the form of a specification enables the registrants to clearly formulate their "contract" with the intended user community. Therefore, although the registration policy for formal namespaces is Expert Review and a stable specification is not strictly required, the designated experts for NID registration requests ought to encourage applicants to provide a stable specification documenting the namespace definition.

Naming can be difficult and contentious; the designated experts and applicants are strongly encouraged to work together in a spirit of good faith and mutual understanding to achieve rough consensus on progressing registrations through the process. They are also encouraged to bring additional expertise into the discussion if that would be helpful in adding perspective or otherwise resolving issues.

10. IANA Considerations

10.1. URI Scheme

This section updates the registration of the 'urn' URI scheme in the Permanent URI Registry [URI-Registry] .

[Note to RFC Editor: please replace "XXXX" with the number assigned to this document upon publication.]

URI Scheme Name: urn

Status: permanent

URI Scheme Syntax: See Section 3 of [this document].

URI Scheme Semantics: The 'urn' scheme identifies Uniform Resource Names, which are persistent, location-independent resource identifiers.

Encoding Considerations: See Section 3.2 of [this document].

Applications/Protocols That Use This URI Scheme Name: Uniform Resource Names are used in a wide variety of applications, including bibliographic reference systems and as names for Extensible Markup Language (XML) namespaces.

Interoperability Considerations: See Section 5 of [this document]. [[CREF18: WG: More text needed, possibly along the lines of Juha's posting of Tuesday, January 13, 2015 14:45 +0200, Subject: Re: [urn] IANA considerations and interoperability]]

Security Considerations: See Section 7.4 and Section 11 of [this document].

Contact: URNBIS WG [mailto:urn@ietf.org]

Author/Change Controller: This scheme is registered under the IETF tree. As such, the IETF maintains change control.

References None.

10.2. Registration of URN Namespaces

This document outlines the processes for registering URN namespaces, and has implications for the IANA in terms of registries to be maintained. In all cases, the IANA ought to assign the appropriate NID (formal or informal) once the procedures outlined in this document have been completed.

11. Security and Privacy Considerations

The definition of a URN namespace needs to account for potential security and privacy issues related to assignment, use, and resolution of identifiers within the namespace (e.g., some namespace resolvers might assign special meaning to certain characters in the Namespace Specific String); see Section 7.4 for further discussion.

In most cases, URN namespaces provide a way to declare public information. Nominally, these declarations will have a relatively low security profile, however there is always the danger of "spoofing" and providing misinformation. Information in these declarations ought to be taken as advisory.

12. References

12.1. Normative References

[RFC20] Cerf, V., "ASCII format for network interchange", RFC 20, October 1969.

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

[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.

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

[URI-Registry] IANA, "Permanent URI Schemes", <http://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml#uri-schemes-1>.

[ISO3166] ISO, "Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", ISO 3166-1:2013, 2013.

12.2. Informative References

[I-D.ietf-urnbis-semantics-clarif] Klensin, J., "URN Semantics Clarification", draft-ietf-urnbis-semantics-clarif-01 (work in progress), February 2015.

[DOI-URI] Paskin, N., Neylon, E., Hammond, T., and S. Sun, "The "doi" URI Scheme for the Digital Object Identifier (DOI)", June 2003, <http://tools.ietf.org/id/draft-paskin-doi-uri-04.txt>.

[RFC1737] Sollins, K. and L. Masinter, "Functional Requirements for Uniform Resource Names", RFC 1737, December 1994.

[RFC2141] Moats, R., "URN Syntax", RFC 2141, May 1997.

[RFC2276] Sollins, K., "Architectural Principles of Uniform Resource Name Resolution", RFC 2276, January 1998.

[RFC3406] Daigle, L., van Gulik, D., Iannella, R., and P. Faltstrom, "Uniform Resource Names (URN) Namespace Definition Mechanisms", BCP 66, RFC 3406, October 2002.

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

[RFC4854] Saint-Andre, P., "A Uniform Resource Name (URN) Namespace for Extensions to the Extensible Messaging and Presence Protocol (XMPP)", RFC 4854, April 2007.

[RFC5122] Saint-Andre, P., "Internationalized Resource Identifiers (IRIs) and Uniform Resource Identifiers (URIs) for the Extensible Messaging and Presence Protocol (XMPP)", RFC 5122, February 2008.

[RFC5890] Klensin, J., "Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework", RFC 5890, August 2010.

[RFC6648] Saint-Andre, P., Crocker, D., and M. Nottingham, "Deprecating the "X-" Prefix and Similar Constructs in Application Protocols", BCP 178, RFC 6648, June 2012.

[RFC6943] Thaler, D., "Issues in Identifier Comparison for Security Purposes", RFC 6943, May 2013.

[RFC6963] Saint-Andre, P., "A Uniform Resource Name (URN) Namespace for Examples", BCP 183, RFC 6963, May 2013.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

[RFC7320] Nottingham, M., "URI Design and Ownership", BCP 190, RFC 7320, July 2014.

[UNICODE] The Unicode Consortium, "The Unicode Standard, Version 6.3", 2013, <http://www.unicode.org/versions/Unicode6.3.0/>.

[XML-BASE]
Marsh, J. and R. Tobin, "XML Base (Second Edition)", World Wide Web Consortium Recommendation REC-xmlbase-20090128, January 2009, <http://www.w3.org/TR/2009/REC-xmlbase-20090128>.

[XML-NAMES]
Thompson, H., Hollander, D., Layman, A., Bray, T., and R. Tobin, "Namespaces in XML 1.0 (Third Edition)", World Wide Web Consortium Recommendation REC-xml-names-20091208, December 2009, <http://www.w3.org/TR/2009/REC-xml-names-20091208>.

Appendix A. Registration Template

A.1. Namespace ID

Requested of IANA (formal) or assigned by IANA (informal).

A.2. Version

The version of the registration, starting with 1 and incrementing by 1 with each new version.

A.3. Date

The date when the registration is requested of IANA, using the format YYYY-MM-DD.

A.4. Registrant

The person or organization that has registered the NID, including the following information:

- The name and address of the registering organization.

- The name and contact information (email, phone number, and/or postal address) of the designated contact person.

A.5. Purpose

Described under Section 7.1 of this document.

A.6. Syntax

Described under Section 7.2 of this document. Unless the registration explicitly says otherwise, use of p-, q-, and/or f-components is not allowed for this namespace.

A.7. Assignment

Described under Section 7.3 of this document.

A.8. Resolution

Described under Section 7.5 of this document.

A.9. Documentation

A pointer to an RFC, a specification published by another standards development organization, or another stable document that provides further information about the namespace.

A.10. Revision Information

(Applicable only when earlier registrations have been revised.)

Description of changes from prior version(s).

Appendix B. Changes from RFC 2141

This document makes the following substantive changes from [RFC2141]:

- o Allows p-components, q-components, and f-components.
- o Disallows "-" at the end of a NID.
- o Allows the "~" and "&" characters in an NSS.
- o Formally registers 'urn' as a URI scheme.

Appendix C. Changes from RFC 3406

This document makes the following substantive changes from [RFC3406]:

1. Relaxes the registration policy for formal namespaces from "IETF Review" to "Expert Review" as discussed in Section 8.1 and Section 9.
2. Removes the category of experimental namespaces, consistent with [RFC6648].
3. Simplifies the registration template.

In addition, some of the text has been updated to be consistent with the definition of Uniform Resource Identifiers (URIs) [RFC3986] and the processes for registering information with the IANA [RFC5226], as

well as more modern guidance with regard to security issues [RFC3552] and identifier comparison [RFC6943].

Appendix D. Contributors

RFC 2141, which provided the basis for the syntax portion of this document, was authored by Ryan Moats.

RFC 3406, which provided the basis for the namespace portion of this document, was authored by Leslie Daigle, Dirk-Willem van Gulik, Renato Iannella, and Patrik Faltstrom.

Their work is gratefully acknowledged.

Appendix E. Acknowledgements

Many thanks to Marc Blanchet, Leslie Daigle, Martin Duerst, Juha Hakala, Ted Hardie, Alfred Hoenes, Paul Jones, Barry Leiba, Sean Leonard, Larry Masinter, Keith Moore, Mark Nottingham, Julian Reschke, Lars Svensson, Dale Worley, and other participants in the URNBIS WG for their input. Alfred Hoenes in particular edited an earlier version of this document and served as co-chair of the URNBIS WG.

Juha Hakala deserves special recognition for his dedication to successfully completing this work, as do Andrew Newton and Melinda Shore in their roles as working group co-chairs and Barry Leiba in his role as area director.

Appendix F. Change log for versions of draft-ietf-urnbis-rfc2141bis-urn

[[RFC Editor: please remove this appendix before publication.]]

F.1. Changes from -08 to -09

- o Altered the text in Section 5 to reflect list discussions about the earlier phrasing. Also added DOI example and citation to that section.

- o Clarified the naming rules for formal namespaces and their relationship to ISO 3166, IDNA, etc., reserved strings.

- o Added an explicit statement about use of URNs in various protocols and contexts to Section 5.

- o Clarified that experimental namespace NIDs, which were explicitly not registered, are not valid URNs (in Section 6.

- o Transformed the partial production in Section 6.2 into valid ABNF.

- o Added more text about p-/q-/f-components and recommendations about use.

- o Added clarifying note about "?" within q-components and f-components.

- o Added explicit requirement that revisions of existing registrations document the changes and added a slot for that description to the template.

- o Many small editorial changes and adjustments including adding additional references and cross-references for clarification.

- o Inserted a placeholder for additional examples.

F.2. Changes from -09 to -10

- o Several clarifying editorial changes, most suggested by Ted Hardie and Henry S. Thompson (some of them off-list).

- o Added a large number of placeholders that identify issues that require WG consideration and resolution (or WG delegation to the editors).

Authors' Addresses

Peter Saint-Andre
&yet

Email: peter@andyet.com
URI: https://andyet.com/

John C Klensin
1770 Massachusetts Ave, Ste 322
Cambridge, MA 02140
USA

Phone: +1 617 245 1457
Email: john-ietf@jck.com