IETF URNbis WG
Internet-Draft
Obsoletes: 2141 (if approved)
Intended status: Standards Track
Expires: May 28, 2011

A. Hoenes, Ed.
TR-Sys

November 24, 2010

Uniform Resource Name (URN) Syntax
draft-ietf-urnbis-rfc2141bis-urn-00

Abstract

Uniform Resource Names (URNs) are intended to serve as persistent, location-independent, resource identifiers. This document serves as the foundation of the 'urn' URI Scheme according to RFC 3986 and sets forward the canonical syntax for URNs, which subdivides URNs into "namespaces". A discussion of both existing legacy and new namespaces and requirements for URN presentation and transmission are presented. Finally, there is a discussion of URN equivalence and how to determine it. This document supersedes RFC 2141.

The requirements and procedures for URN Namespace registration documents are currently set forth in RFC 3406, which is also being updated by a companion, revised specification dubbed RFC 3406bis.

Discussion

This draft version has been obtained by importing the text from RFC 2141 into modern tools and making a first rounds of updating steps. It is a chartered initial work item of the URNbis WG in the IETF; the aim is to bring URN RFCs in alignment with STD 66, STD 68, BCP 26, and the requirements from emerging distributed national and international URN resolution systems, and advance them on the IETF Standards Track.

Comments are welcome on the urn@ietf.org mailing list (or sent to the document editor). The home page of the URNbis WG is located at <http://tools.IETF.ORG/wg/urnbis/>.

Status of This Memo

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 May 28, 2011.

Copyright Notice

Table of Contents

1. Introduction

'urn' is a particular URI Scheme (according to STD 66, RFC 3986 [RFC3986] and BCP 35, RFC 4395 [RFC4395]) that is dedicated to forming a hierarchical framework for persistent identifiers.

Uniform Resource Names (URNs) are intended to serve as persistent, location-independent, resource identifiers and are designed to make it easy to map other namespaces (that share the properties of URNs) into URI-space. Therefore, the URN syntax provides a means to encode character data in a form that can be sent in existing protocols, transcribed on most keyboards, etc.

The first level of hierarchy is given by the classification of URIs into "URI Schemes", and for URNs, the second level is organized into "URN Namespaces". Henceforth both terms are used in this capitalization to distinguish them from the more general common meaning of "scheme" and "namespace".

1.1. Historical Perspective and Motivation

For the intended audience of this RFC, which is expected to include groups interested in persistent identifiers in general and not in continuous contact with the IETF and the RFC series, this section gives a brief outline of the evolution of the matter over time. Appendix A gives hints on how to obtain RFCs and related information.

Attempts to define generally applicable identifiers for network resources go back to the mid-1970 years. Among the applicable RFCs is RFC 615 [RFC0615], which subsequently has been obsoleted by RFC 645 [RFC0645].

The seminal document in the RFC series regarding URIs (Uniform Resource Identifiers) for use with the World Wide Web (WWW) has been RFC 1630 [RFC1630], published in 1994. In the same year, the general concept or Uniform Resource Names has been laid down in RFC 1737 [RFC1737] and that of Uniform Resource Locators in RFC 1736 [RFC1736].

The original formal specification of URN Syntax, RFC 2141 [RFC2141] has been adopted in 1997. That document was based on the original specification of URLs (Uniform Resource Locators) in RFC 1738 [RFC1738] and RFC 1808 [RFC1808], which later on, in 1998, has been generalized and consolidated in the Generic URI specification, RFC 2396 [RFC2396]. Most parts of these URI/URL documents have been superseded in 2005 by STD 66, RFC 3986 [RFC3986]. Notably, RFC 2141 makes (essentially normative) reference to a draft version of RFC 2396.

Over time, the terms "URI", "URL", and "URN" have been refined and slightly shifted according to emerging insight and use. This has been clarified in a joint effort of the IETF and the World Wide Web Council, published 2002 for the IETF in RFC 3305 [RFC3305].

The wealth of URI Schemes and URN Namespaces needs to be organized in a persistent way, in order to guide application developers and users to the standardized top level branches and the related specifications. These registries are maintained by the Internet Assigned Numbers Authority (IANA) [IANA] at [IANA-URI] and [IANA-URN], respectively. Registration procedures for URI Schemes originally had been laid down in RFC 2717 [RFC2717] and guidelines for the related specification documents were given in RFC 2718 [RFC2718]. These documents have been obsoleted and consolidated into BCP 35, RFC 4395 [RFC4395], which is based on, and aligned with, RFC 3986.

> Note that RFC 2141 predates RFC 2717 and, although the ’urn’ URI scheme is listed in [IANA-URI] with a pointer to RFC 2141, this registration has never been performed formally.

Similarly, the URN Namespace definition and registration mechanisms originally have been specified in RFC 2611 [RFC2611], which has been obsoleted by BCP 66, RFC 3406 [RFC3406]. Guidelines for documents prescribing IANA procedures have been revised as well over the years, and at the time of this writing, BCP 26, RFC 5226 [RFC5226] is the normative document. Neither RFC 4395 nor RFC 3406 conform to RFC 5226.

Early documents specifying URI and URN syntax, including RFC 2141, made use of an ad-hoc variant of the original Backus-Naur Form (BNF) that never has been formally specified.

Over the years, the IETF has shifted to the use of a predominant formal language used to define the syntax of textual protocol elements, dubbed "Augmented Backus-Naur Form" (ABNF). The specification of ABNF also has evolved, and now STD 68, RFC 5234 [RFC5234] is the normative document for it (that also will be used in this RFC).

1.2. Background on Properties of URNs

RFC 1738 [RFC1738] defined the purpose of URNs as follows:

- The purpose or function of a URN is to provide a globally unique, persistent identifier used for recognition, for access to characteristics of the resource or for access to the resource itself.

Section 2 of RFC 1738 [RFC1738] listed the functional requirements for URNs (quote slightly edited to reflect the time passed since that RFC had been written and the actual definition of the URN scheme that has happened):

- Global scope: A URN is a name with global scope which does not imply a location. It has the same meaning everywhere.
- Global uniqueness: The same URN will never be assigned to two different resources.
- Persistence: It is intended that the lifetime of a URN be permanent. That is, the URN will be globally unique forever, and may well be used as a reference to a resource well beyond the lifetime of the resource it identifies or of any naming authority involved in the assignment of its name.
- Scalability: URNs can be assigned to any resource that might conceivably be available on the network, for hundreds of years.
- Legacy support: The URN scheme permits the support of existing legacy naming systems, insofar as they satisfy the other requirements described here. [...]
- Extensibility: The URN scheme permits future extensions.
- Independence: It is solely the responsibility of a name issuing authority to determine the conditions under which it will issue a name.
- Resolution: URNs will not impede resolution. [...]

The URN syntax described below also accommodates the fundamental "Requirements for URN Encoding" in Section 3 of RFC 1738 [RFC1738], as far as experience gained has not lead to lessen unrealistical detail requirements:

- Single encoding: The encoding for presentation for people in clear text, electronic mail and the like is the same as the encoding in other transmissions.
- Simple comparison: A comparison algorithm for URNs is simple, local, and deterministic. [...]
- Human transcribability: For URNs to be easily transcribable by humans without error, they need to be short, use a minimum of special characters, and be case insensitive. [...]

- o Transport friendliness: A URN can be transported unmodified in the common Internet protocols, such as TCP, SMTP, FTP, Telnet, etc., as well as printed paper.

- o Machine consumption: A URN can be parsed by a computer.

- o Text recognition: The encoding of a URN needs to enhance the ability to find and parse URNs in free text.

1.3. Objective of this Memo

RFC 2141 does not seamlessly match current Internet Standards. The primary objective of this document is the alignment with the URI Standard [RFC3986] and guidelines [RFC4395], the ABNF Standard [RFC5234] and the current IANA Guidelines [RFC5226] in general.

Further, experience from emerging international efforts to establish a general, distributed, stable URN resolution service are expected to be taken into account during the draft stage of this document.

For advancing the URN specification on the Internet Standards-Track, it needs to be based on documents of comparable maturity. Therefore, to further advancements of the formal maturity level of this RFC, it deliberately makes normative references only to documents at Full Standard or Best Current Practice level.

Thus, this replacement document for RFC 2141 should make it possible to advance the URN framework on the Internet Standard maturity ladder. All other related documents depend on it; therefore this is the first step to undertake.

Out of scope for this document is a revision of the URN Namespace Definition Mechanisms document, BCP 66 [RFC3406]. This is going to be undertaken in a companion document, RFC 3406bis.

1.4. Requirement Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. URN Syntax

This document defines the URI Scheme ’urn’. Hence, URNs are specific URIs as specified in STD 66 [RFC3986]. The formal syntax definitions below are given in ABNF according to STD 68 [RFC5234] and make use of some "Core Rules" specified in Appendix B of that Standard and several generic rules defined in Appendix A of RFC 3986.

The syntax definitions below do, and syntax definitions in dependent documents MUST, conform to the URI syntax specified in RFC 3986, in the sense that additional syntax rules must only constrain the general rules from RFC 3986. In other words: a general URI parser based on RFC 3986 MUST be able to parse any legal URN, and specific semantics can be obtained from URN-specific parsing.

> NOTE: The remainder of this Section still requires substantial work! To give a starting point for WG discussion, within this entire Section, much of the elaborations and editorial comments from the Individual I-D predecessor of this draft are kept. This will be cleaned up after discussion.

URNs conform to the <path-rootless> variant of the general URI syntax specified in Section 3 of [RFC3986] :

```
URI = scheme ":" path-rootless [ "?" query ] [ "#" fragment ]

path-rootless = segment-nz *( "/" segment )

segment-nz    = 1*pchar
segment       = *pchar

pchar = unreserved / pct-encoded / sub-delims / ":" / "@"
```

In the case of URNs, we have:

```
scheme     = "urn"
```

and the following additional syntax rule is superimposed on <path-rootless> to establish a level of hierarchy called "Namespace":

```
urn-path   = NID ":" NSS
```

Here "urn" is the URI scheme name, <NID> is the Namespace Identifier, and <NSS> is the Namespace Specific String. The colons are REQUIRED separator characters.

Per RFC 3986, the URN Scheme name (here "urn") is case-insensitive.

The Namespace ID (also a case-insensitive string) determines the syntactic structure and the semantic interpretation of the Namespace Specific String. Generic details on NID syntax can be found below in Section 2.1, and the NSS syntax is elaborated upon in Section 2.2.

Each particular namespace is based on a specific document that must normatively describe (among other things) the details of the <NSS> values allowed in conjunction with the respective <NID>. The

specification requirements and registration procedures for URN Namespaces are the subject of a dedicated document, currently RFC 3406 [RFC3406] -- to be updated for conformance to BCP 26 and alignment with implementation experience, in RFC 3406bis.

Note:

RFC 2141 has deferred the decision on whether <query> and <fragment> components are applicable to URNs and reserved the use of bare (unencoded) question mark ("?") and hash ("#") characters in URNs for future usage in conformance with the generic URI syntax.

There is evidence of desire to be able to use these components (which are split off by the high-level parsing rules of RFC 3986), or at least the <fragment> component, in URNs belonging to selected namespaces. Thus, this draft version tentatively aims at allowing these components in the general syntax.

The considerations below reflect the current thinking based on implementation experience and preliminary discussion.

The syntax of <query> and <fragment> are defined in RFC 3986. Question mark and hash sign remain reserved as separator characters for these URI components and cannot appear unencoded in an NSS. This way, backwards compatibility with existing URN namespaces is guaranteed and compatibility with general URI parsers is improved.

The <query> part MUST NOT be present in any *assigned* URN. This specification reserves its use for future standardization related to URN resolution. This part can only be added to an assigned URN and appear in a URI reference [RFC3986] to a URN that is intended to be used with URN resolution services, and, in accordance with the general specification of this part in RFC 3986, its purpose is restricted to designate service aspects of the intended resolution response, e.g., to select the kind and amount of metadata sought about the given object that is identified by the basic, assigned URN.

The <fragment> part is not generally allowed in URNs. It is only applicable to URN Namespaces that specifically opt to support its usage. Thus, a URN Namespace registration document MAY specify the usage of <fragment> with URNs of that particular URN Namespace. Absent a registered namespace definition based on this document and RFC 3406bis that explicitly specifies its usage, URNs assigned within a particular URN Namespace MUST NOT contain a fragment identifier.

The use of fragment identifiers may be useful if the URN Namespace is based on an existing identifier scheme that designates objects of

reasonable complexity that there's a need to make reference of parts of such resources in typical network access environments.

A URN Namespace definition has two options to support fragment identifiers, and only one of these methods is possible within a given URN Namespace:

(a) Fragment identifiers (if any) are assigned individually to parts of a larger entity during the URN assignment process. If a URN Namespace opts for this model, its specification MUST describe the additional syntax restrictions to be adhered and the particulars of the (per-URN) assignment process.

(b) A specific set of fragment identifiers is generally applicable to all resources targeted by URNs of the specific URN Namespace. In this case, the specification document MUST specify a finite set of <fragment> values, or precise, generic rules for the formation of syntactically valid fragment identifiers for the particular URN Namespace. The specification SHOULD indicate the treatment of syntactically valid <fragment> values in case they are not semantically valid for a given base URN. Absent such specification, the default is to ignore such fragment identifiers.

URN resolver clients MUST pass a given <fragment> part of a URN unchanged to the resolver service. The default URN resolution behavior is to ignore any <fragment> part if either the applicable URN Namespace definition did not specify its use, or if no specific related information was available for the basic resource in case (b) above, or if that basic URN plus fragment identifier has not been assigned in case (a) above.

2.1. Namespace Identifier (NID) Syntax

The following is the syntax for the Namespace Identifier. To (A) be consistent with all potential resolution schemes and (B) not put any undue constraints on any potential resolution scheme, Namespace Identifiers are ASCII strings with the syntax:

```
NID = ( ALPHA / DIGIT ) 0*31 ( ALPHA / DIGIT / "-" )
```

Note for discussion:
The above definition is taken from RFC 2141. Should this be further restricted, e.g., to avoid possible confusion caused by multiple adjacent hyphens and NIDs looking like a numerical value or a numerical range? Does it really make sense to allow single-letter NIDs? Such restrictions would be fully backward compatible because no NIDs have been defined so far that would violate these

restrictions. Hyphens have been used only in the naming pattern for "Informal Namespace IDs" per RFC 3406.

Namespace Identifiers are case-insensitive, so that for instance "ISBN" and "isbn" refer to the same namespace.

To avoid confusion with the URI Scheme name "urn", the NID "urn" is permanently reserved by this RFC and MUST NOT be used or registered.

2.2. Namespace Specific String (NSS) Syntax

Note:
In order to make visible the migration path from RFC 2141 and the influence of the evolution of URI syntax from RFC 2396 to RFC 3986 on it, at this draft stage, the subsequent syntax description is highly annotated and expanded. After discussion, a substantial consolidation is expected.

As already required by RFC 1737, there is a single canonical representation of the NSS portion of an URN.

Note:
If the DISCUSSes above and below can be affirmed (allowing optional <query> and <fragment> components as well as "&" and "~" in the path), the syntax below could be simplified very much to:

```
NSS   = 1*pchar   ; or equivalent:    NSS   = segment-nz
```

The format of this single canonical form follows:

```
      NSS         = 1*URN-char

      URN-char    = trans / pct-encoded

      trans       = ALPHA / DIGIT / u-other
  ; NO?            / reserved
  ; Issue: This lead to ambiguity in RFC 2141 wrt "%".

      u-other     = ":" / "@"
                    ; those from RFC 3986 <gen-delims>
                    ; specifically allowed in <pchar>.
  ; From RFC 3986:
  ;    gen-delims  = ":" / "/" / "?" / "#" / "[" / "]" / "@"

                  / "!" / "$" /       "'" / "(" / ")"
                  / "*" / "+" / "," / ";" / "="
                    ; this is RFC 3986 <sub-delims> except "&".
  ; From RFC 3986:
  ;    sub-delims  = "!" / "$" / "&" / "'" / "(" / ")"
  ;                / "*" / "+" / "," / ";" / "="
  ; Issue: can/should "&" be allowed ?
  ; If we allow <query> and <fragment> according to the
  ; generic URI syntax, there seems to be no more need to exclude "&".

                  / "-" / "." / "_"   ; <unreserved> except "~"
  ; From RFC 3986:
  ;    unreserved  = ALPHA / DIGIT
  ;                / "-" / "." / "_" / "~"
  ; Issue: can/should "~" be allowed as well ?

  ; If we allow "&" and "~" , <trans> becomes <pchar> ,
  ; greatly simplifying the syntax rules and parsers!

  ; from RFC 2141:
  ;    reserved  = '%" / "/" / "?" / "#"         ; SIC!
  ;                ^ ^
```

Depending on the rules governing a namespace, valid identifiers in a namespace might contain characters that are not members of the URN character set above (<URN-char>). Such strings MUST be translated into canonical NSS format before using them as protocol elements or otherwise passing them on to other applications. Translation is done by encoding each character outside the URN character set as a sequence of octets using UTF-8 encoding STD 63 [RFC3629], and the "percent-encoding" of each of those octets as "%" followed by two <HEXDIG> characters. The two characters form the hexadecimal representation of that octet.

2.3. Special and Reserved Characters

The remaining printable characters left to be discussed above comprise the generic delimiters and the reserved characters, which are restricted for special use only. These characters are discussed below, giving the specifics of why each character is special or reserved.

2.3.1. Delimiter Characters

RFC 3986 [RFC3986] defines the general delimiter characters used in URIs:

```
gen-delims = ":" / "/" / "?" / "#" / "[" / "]" / "@"
```

From among the <gen-delims>, ":" and "@" are also included in <pchar> and hence allowed in the path components of URIs.

The at-character ("@") in generic URIs only has a specific meaning when contained in the <authority> part, which is absent in URNs. Hence, "@" is available in the <NSS> part of URNs.

With URNs, the colon (":") is used as a delimiter character not only between the scheme name ("urn") and the <NID>, but also between the latter and the <NSS>, and many existing URN namespaces additionally use ":" to further subdivide a single RFC 3986 path segment in the <NSS> in a hierarchical manner.

Note: Using ":" as a sub-delimiter in the path in favor of "/" is attractive because it avoids possible complications that could arise from accidental inappropriate use of relative URI references [RFC3986] for URNs.

The characters "/", "?", and "#" separate path components and the <query> and <fragment> parts in the generic URI syntax; they are restricted to this role in URNs as well, although the <path> in URNs only admits a single and hence "/" is not allowed. Therefore, these characters MUST NOT appear in the <NSS> part of a URN in unencoded form. Namespaces that need these characters MUST employ in their URNs the appropriate percent-encoding for each such character.

The square brackets ("[" and "]") also play a particular role when contained in the <authority> part, which is absent in URNs. However, for conformance with the generic URI syntax, they are not allowed literally in the <NSS> component of URNs. If a specific URN namespace reflects semantics that require these characters, they MUST be percent-encoded in the respective URNs.

2.3.2. The Percent Character

The percent character ("%") is reserved in the URN syntax for introducing the escape sequence for an octet that is either not a printable ASCII character or reserved for special purposes, as described in this section. The presence of a "%" character in a URN MUST always be followed by two <HEXDIG> characters, which three together semanticaly form an abstract <pct-encoded> octet. Literal use of the "%" character in an underlying namespace MUST therefore be encoded as "%25" in URNs for that namespace.

Namespaces MAY designate one or more characters from the URN character set as having special meaning for that namespace. If the namespace also uses that character in a literal sense as well, the character used in a literal sense MUST be encoded with "%" followed by the hexadecimal representation of that octet. Further, a character MUST NOT be percent-encoded if the character is not a reserved character. Therefore, the process of registering a namespace identifier shall include publication of a definition of which characters have a special meaning to that namespace.

2.3.3. Other Excluded Characters

The following list is included only for the sake of completeness. It includes the characters discussed in Sections 2.3.1 and 2.3.2. Any octets/characters on this list are explicitly NOT part of the URN <NSS> character set, and if used in an URN, MUST be percent-encoded.

```
            excluded = CTL / SP        ; control characters and space
                     / DQUOTE          ; "
                     / "#"             ; from <gen-delims>
                     / "%"             ; see above
   ; DISCUSS!        / "&"             ; DISCUSS -- see above!
                     / "/"             ; from <gen-delims>
                     / "<" / ">"
                     / "?"             ; from <gen-delims>
                     / "["             ; from <gen-delims>
                     / "\"
                     / "]"             ; from <gen-delims>
                     / "^"
                     / "`"
                     / "{" / "|" / "}"
   ; DISCUSS!        / "~"             ; DISCUSS -- see above!
                     / %x7F            ; DEL (control character)
                     / %x80-FF         ; non-ASCII
```

The NUL octet (0 hex) is renowned for a long history of trouble in implementations. It MUST NOT be used, in either unencoded or percent-encoded form.

In textual context, a URN ends when an octet/character from the excluded character set (<excluded>) is encountered. The character from the excluded character set is NOT part of the URN.

[Does that still make sense? -- it collides with possible query / fragment parts!]

3. Support of Existing Legacy Naming Systems and New Naming Systems

Any namespace (existing or newly devised) that is proposed as a URN namespace and fulfills the criteria of URN namespaces MUST be expressed in this syntax. If names in these namespaces contain characters other than those defined for the URN character set, they MUST be translated into canonical form as discussed in Section 2.2.

4. URN Presentation and Transport

The URN syntax defines the canonical format for URNs and all URN transport and interchanges MUST take place 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 MAY support display of URNs in a more human-friendly form and may use a character set that includes characters that aren’t permitted in URN syntax as defined in this RFC (that is, they may replace %-notation by characters in some extended character set in display to humans).

5. Lexical Equivalence of URNs

For various purposes such as caching, it is often desirable to determine whether two URNs are the same without resolving them. The general-purpose means of doing so is by testing for "lexical equivalence" as defined below.

Two URNs are lexically equivalent if they are octet-by-octet equal after the following preprocessing:

1. normalize the case of the leading "urn" scheme name;
2. normalize the case of the NID;
3. normalize the case of any percent-encoding.

Note that percent-encoding MUST NOT be removed. It is an implementation detail not affecting interoperability whether a URN comparison function internally prefers normalization (in the above 3 steps) to lower or to upper case.

Some namespaces may define additional lexical equivalences, such as case-insensitivity of the NSS (or parts thereof). Additional lexical equivalences MUST be documented as part of namespace registration, MUST always only have the effect of eliminating some of the false negatives obtained by the procedure above, i.e. they MUST NOT say that two URNs are not equivalent if the procedure above says they are equivalent.

5.1. Examples of Lexical Equivalence

The following hypothetical URN comparisons highlight the lexical equivalence definitions:

1- URN:foo:a123,456
2- urn:foo:a123,456
3- urn:FOO:a123,456
4- urn:foo:A123,456
5- urn:foo:a123%2C456
6- URN:FOO:a123%2c456

URNs 1, 2, and 3 are all lexically equivalent. URN 4 is not lexically equivalent to any of the other URNs of the above set. URNs 5 and 6 are only lexically equivalent to each other.

6. Functional Equivalence of URNs

Functional equivalence is determined by practice within a given namespace and managed by resolvers for that namespace. Thus, it is beyond the scope of this document. Namespace registrations must include guidance on how to determine functional equivalence for that namespace, i.e., when two URNs are identical within a namespace.

On the other hand, it is permissible to have two different URNs -- even from different URN namespaces -- be assigned to a particular resource. This can only be detected by resolving the URNs and analysis of the resolution responses; hence, this is out of scope for this memo.

7. The ’urn’ URI Scheme

At the time of publication of RFC 2141, no formal registration procedure for URI Schemes had been established yet, and so IANA only informally has registered the ’urn’ URI Scheme with a reference to [RFC2141].

Section 7.1 below contains the URI scheme registration template for the ’urn’ scheme, in accordance with RFC 4395 [RFC4395].

Note: In order to be usable as a standalone text (after being extracted from this RFC), the template below does not contain formal anchors to the references listed in section 11, but instead gives the common document designations in prose. However, for compliance with editorial policy, it needs to be noted here:

This registration template refers to RFCs 2196, 2276, 2608, 3401 through 3404, 3406, 3629 (STD 63), and 3986 (STD 66) ([RFC2169] [RFC2276] [RFC2608] [RFC3401] [RFC3402] [RFC3403] [RFC3404] [RFC3406] [RFC3629] [RFC3986]).

7.1. Registration of URI Scheme 'urn'

[RFC Editor: Please replace "XXXX" in all instances of "RFC XXXX" below by the RFC number assigned to this document.]

URI scheme name: urn

Status: permanent

URI scheme syntax:

See Section 2 of RFC XXXX.

URI scheme semantics:

'urn' URIs, known as Universal Resource Names (URNs), serve as persistent, location-independent, resource identifiers for concrete and abstract objects that have network accessible instances and/or metadata.

URNs are structured hierarchically into URN Namespaces, the management of which is delegated to namespace-specific authorities. Each such URN namespace is founded in an independent specification and registered with IANA, following the guidelines and procedures of BCP 66 (at the time of this registration: RFC 3406).

Encoding considerations:

All URNs are ASCII strings conforming to the general URI syntax from STD 66. As described in Sections 2.2 and 2.3.2 of RFC XXXX, characters needed by the URN namespace specific semantics but not contained in the US-ASCII charset MUST be encoded in UTF-8 according to STD 63; any octets outside the allowed character set MUST then be percent-encoded.

Applications/protocols that use this URI scheme:

URNs that serve to identify abstract resources for protocol purposes are expected to be recognized directly by the implementations of these portocols.

In general, resolution systems for URNs are specified on a per-namespace basis. If appropriate for the namespace, these systems resolve URNs to (possibly multiple) URIs that allow the network access to the identified object or metadata on it.

"Architectural Principles of Uniform Resource Name Resolution" (RFC 2276) explains the basic concepts. Some resolution systems laid down in IETF specifications are:

* Trivial HTTP-based URN Resolution (RFC 2169)

* Dynamic Delegation Discovery System (DDDS, RFCs 3401-3404)

* Service Location Protocol (SLPv2, RFC 2608)

Interoperability Considerations:

Persistence and stability of URNs require appropriate resolution systems.

Security Considerations:

See Section 8 of RFC XXXX.

Contact:

The IETF URNbis working group.
This registration will be discussed on the following IETF lists:
urn and uri-review (AT ietf.org).

Author / Change controller:

The authors of RFC XXXX.
Change control is with the IESG.

References:

RFC XXXX.

Procedures for the specification and registration of URN namespaces are detailed in BCP 66 (at the time of this writing: RFC 3406; the URNbis WG is chartered to provide a RFC 3406bis).

8. Security Considerations

This document specifies the syntax and general requirements for URNs, which are the specific URIs that use the 'urn' URI scheme. As such, the general security considerations of STD 66 [RFC3986] apply. However, each URN namespace will have specific security considerations, according to the semantics and usage of the underlying namespace. While some namespaces may assign special meaning to certain of the characters of the Namespace Specific String, any security considerations resulting from such assignment are outside the scope of this document. It is REQUIRED by BCP 66 [RFC3406] that the process of registering a namespace identifier include any such considerations.

9. IANA Considerations

IANA is asked to update the existing informal registration of the 'urn' URI Scheme by the template in Section 7.1 above and list this RFC as the current normative reference in [IANA-URI].

IANA is asked to add a note to [IANA-URN] that 'urn' is a permanently reserved formal namespace identifier string that cannot be registered, in order to avoid confusion with the 'urn' URI scheme.

10. Acknowledgements

This document is heavily based on RFC 2141, the author of which has laid the foundation for this work; that RFC contained the following Acknowledgements:

> Thanks to various members of the URN working group for comments on earlier drafts of this document. This document is partially supported by the National Science Foundation, Cooperative Agreement NCR-9218179.

This document also heavily relies on and acknowledges the work done for STD 66 [RFC3986] and earlier RFCs that are being quoted informally, in particular RFC 1737 [RFC1737].

Your name could go here ...

11. References

11.1. Normative References

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

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

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

[RFC4395] Hansen, T., Hardie, T., and L. Masinter, "Guidelines and Registration Procedures for New URI Schemes", BCP 35, RFC 4395, February 2006.

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

11.2. Informative References

[IANA] IANA, "The Internet Assigned Numbers Authority", <http://www.iana.org/>.

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

[IANA-URN] IANA, "URN Namespace Registry", <http://www.iana.org/assignments/urn-namespaces/>.

[RFC0615] Crocker, D., "Proposed Network Standard Data Pathname syntax", RFC 615, March 1974.

[RFC0645] Crocker, D., "Network Standard Data Specification syntax", RFC 645, June 1974.

[RFC1630] Berners-Lee, T., "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", RFC 1630, June 1994.

[RFC1736] Kunze, J., "Functional Recommendations for Internet Resource Locators", RFC 1736, February 1995.

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

[RFC1738] Berners-Lee, T., Masinter, L., and M. McCahill, "Uniform Resource Locators (URL)", RFC 1738, December 1994.

[RFC1808] Fielding, R., "Relative Uniform Resource Locators", RFC 1808, June 1995.

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

[RFC2169] Daniel, R., "A Trivial Convention for using HTTP in URN Resolution", RFC 2169, June 1997.

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

[RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.

[RFC2608] Guttman, E., Perkins, C., Veizades, J., and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999.

[RFC2611] Daigle, L., van Gulik, D., Iannella, R., and P. Faltstrom, "URN Namespace Definition Mechanisms", BCP 33, RFC 2611, June 1999.

[RFC2717] Petke, R. and I. King, "Registration Procedures for URL Scheme Names", BCP 35, RFC 2717, November 1999.

[RFC2718] Masinter, L., Alvestrand, H., Zigmond, D., and R. Petke, "Guidelines for new URL Schemes", RFC 2718, November 1999.

[RFC3305] Mealling, M. and R. Denenberg, "Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations", RFC 3305, August 2002.

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

[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.

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

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

Appendix A. How to Locate IETF Documents (Informative)

Request For Comments (RFCs) are available from the RFC Editor site using the canonical URIs <http://www.rfc-editor.org/rfc/rfcNNNN.txt> or <ftp://ftp.rfc-editor.org/in-notes/rfcNNNN.txt> (where 'NNNN' is the serial number of the RFC), and from numerous mirror sites. Additional metadata for any RFC, including possible Errata, are available from <http://www.rfc-editor.org/info/rfcNNNN> (where 'NNNN' again is the serial number of the RFC). A HTML-ized version and a PDF facsimile of each RFC are available from the IETF Tools site at <http://tools.ietf.org/http/rfcNNNN> and <http://tools.ietf.org/pdf/rfcNNNN>, respectively.

Current Internet Draft documents are available via the search engines at <http://www.ietf.org/id-info/> and <http://www.rfc-editor.org/idsearch.html>; archival copies of older IETF documents can be found at <http://tools.ietf.org/id/>.

Appendix B. Handling of URNs by URL Resolvers/Browsers

The URN syntax has been defined so that URNs can be used in places where URLs are expected. A resolver that conforms to the current URI syntax specification [RFC3986] will extract a scheme value of "urn" rather than a scheme value of "urn:<nid>".

An URN MUST be considered an opaque URI by URL resolvers and passed (with the "urn:" tag) to a URN resolver for resolution. The URN resolver can either be an external resolver that the URL resolver knows of, or it can be functionality built into the URL resolver.

To avoid confusion of users, a URL browser SHOULD display the complete URN (including the "urn:" tag) to ensure that there is no confusion between URN Namespace identifiers and URI Scheme names.

Appendix C. Collected ABNF (Informative)

As a service to implementers specifically interested in URN syntax, after consolidation of Section 2, the complete ABNF for URNs will be collected here, including the referenced rules from [RFC5234] and

[RFC3986]. In case of (unexpected) inconsistencies, these documents remain normative for the respective productions.

T.B.D.

...

Appendix D. Changes since RFC 2141 (Informative)

D.1. Essential Changes from RFC 2141

[RFC Editor: please remove the Appendix D.1 headline and all subsequent subsections starting with Appendix D.2.]

T.B.D. (after consolidation of this memo)

D.2. Changes from RFC 2141 to Individual Draft -00

Abstract amended: URI scheme, replacement for 2141, point to 3406. Use contemporary boilerplate. Added transient "Discussion" section.

s1: added new 1st para (URI scheme) and 3rd para (hierarchy).
s1.1 (Historical Perspective) added for background & motivation.
s1.2 (Objective) added.
s1.3 (2119 keywords) added -- used now throughout normative text.

s2 (URN Syntax): Shifted from BNF to ABNF; explain relationship to 3986 and gaps, how the gaps could be bridged, distinguish between URI generics and URN specifics; got rid of references to immature documents (1630, 1737).
s2.1 (NID syntax): Use ABNF and RFC 5234 terminals (core rules); removed reference to an old draft of 2396; clarified prohibition to use "urn" as NID.
s2.2 (NSS syntax): Shifted from BNF to ABNF; made ABNF consistent with subsequent textual description; exposition much expanded, showing relationship with 3986 and resulting incompatibilities; proposed how to bridge gaps, to make parsing more uniform among URIs; updated i18n considerations and pointer to UTF-8 specification.
s.2.3, s2.3.*: reworked and much expanded, along the grouping of delimiter characters from 3986 in new s2.3.1 (including old s.2.3.2); made text fully consistent with ABNF in s2.2; consistent usage of term "percent-encoded"; old s.2.3.1 became s2.3.2; old s3.4 became s3.3.3, providing complete, annotated list of excluded characters, ordered by ascending code point; and restating design decisions needed to be made to close gaps to 3986.

s3 through s6: only minor editorial changes.

s7: formal registration of 'urn' URI scheme added, using 4395 template.

s8: Security Cons. slightly amended.

s9: new: IANA Cons. added wrt s7.1 and prohibition of NID "urn".

s10: Acknowledgments amended.

s11: References split into Normative and Informative; updated refs and added many; only FS and BCP allowed as Normative Refs to further promotion of document.

Added Appendices A through D.

D.3. Changes from Individual Draft -00 to -02

Updated "Discussion" on front page to point to dedicated urn list.

Numerous editorial improvements and additions for clarification, in particular in the Introduction. No technical changes.

More Informative References; missing details supplied in D.1.

D.4. Changes from Individual Draft -02 to WG Draft -00

Added new s2.1 with excerpts from RFC 1737 to Introduction to provide background on URN functional and syntax requirements.

Supplied text in s2 regarding the envisioned use of query and fragment parts, based on various discussion -- including a preliminary evaluation in PersID.

Changed "SHOULD never" to "MUST NOT" for NUL character in NSS.

Various editorial and grammar fixes; corrected STD / BCP numbers.

Author's Address

Alfred Hoenes (editor)
TR-Sys
Gerlinger Str. 12
Ditzingen D-71254
Germany

EMail: ah@TR-Sys.de