IETF URNbis WG
Internet-Draft
Obsoletes: 2141 (if approved)
Intended status: Standards Track
Expires: April 19, 2013

A. Hoenes, Ed.
TR-Sys

October 16, 2012

Uniform Resource Name (URN) Syntax
draft-ietf-urnbis-rfc2141bis-urn-03

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 set forth in a companion document, RFC 3406bis (BCP 66).

Discussion

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/>.
[[RFC-Editor: this clause to be deleted before RFC publication]]

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 April 19, 2013.

Copyright Notice

Table of Contents

1. Introduction

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.

To this end, URNs are designed as an intrinsic part of the more general framework of Uniform Resource Identifiers (URIs); '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. (Other, legacy interpretations of the term URN are not considered in this memo.)

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".

It is an explicit design goal that pre-existing systems of persistent identifiers are mapped into the URN framework. Ordinarily, each such traditional identifier system (namespace) -- standard or otherwise -- will occupy its own URN Namespace. However, shared URN Namespaces are possible (and in fact, already exist), but the identifier-driven mechanisms needed to distinguish the originating namespaces make registration and maintenance of such URN Namespaces more complicated.

URN (as a URI Scheme) as such does not have a specific scope. The applicability of the URN system, that is, the totality of the resources that URNs can be assigned to, is the union of all identifier systems that have an associated registered URN Namespace. Ideally every new namespace will thus extend the URN applicability.

1.1. Historical Perspective and Motivation

Since this RFC will be of particular interest for groups and individuals that are interested in persistent identifiers in general, but often not in steady contact with the IETF and the RFC series, this section gives a brief outline of the evolution of the matter over time.

Attempts to define generally applicable identifiers for network resources go back to the mid-1970s. 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) was 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 (URLs) in RFC 1736 [RFC1736].

The original formal specification of URN Syntax, RFC 2141 [RFC2141] was adopted in 1997. That document was based on the original specification of URLs in RFC 1738 [RFC1738] and RFC 1808 [RFC1808], which later on, in 1998, was generalized and consolidated in the Generic URI specification, RFC 2396 [RFC2396]. Most parts of these URI/URL documents were 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 traditionally was 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. Objective of this Memo

As pointed out above, RFC 2141 does not seamlessly match current Internet Standards. Therefore, the primary objective of this document is the alignment with the URI standard [RFC3986] and URI Scheme 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 have been 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. This is being undertaken in a companion document, RFC 3406bis [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg].

1.3. Background on Properties of URNs

This section aims at quoting requirements as identified in the past; it does not attempt to revise or redefine these requirements, but it gives some hints where more than a decade of experience with URNs has shed a different light on past views. The citations below are given here to make this document self-contained and avoid normative down-references to old work.

RFC 1737 [RFC1737] 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.

This means that URNs are intended to uniquely and persistently bind a name to a resource and (some of) its properties (metadata).

Section 2 of RFC 1737 [RFC1737] listed the functional requirements for URNs (quote slightly edited to reflect the time passed since that RFC was 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 1737 [RFC1737], as far as experience gained has not lead to relax 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. [...]

  Note:
  In particular practice gained with active URN Namespaces has shown that this former goal is rather unrealistic, since usually preference is given to 1:1 embedding into URNs of identifier strings drawn from existing namespaces, which might not have this property. However, we hold that, at least, the rough kind of resource identified by a URN should be easily recognizable for humans.

- 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.

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

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

1.4. Requirement Language

When spelled in all-capitals as in this paragraph, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [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 are only allowed to further 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, URN-specific semantics can be obtained from URN-specific parsing of its outcome.

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

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

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

segment-nz    = 1*pchar
segment       = *pchar
query         = *( pchar / "/" / "?" )
fragment      = *( pchar / "/" / "?" )

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

In the case of URNs, we have:

```
scheme     = "urn"
```

and for <path-rootless>, only a single segment is used, but 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.

> Note that it is common practise in several existing URN Namespaces (and fully supported by this syntax) to use additional colon(s) as separator character(s) in order to introduce further level(s) of hierarchy into the NSS syntax, where needed. (See also Section 2.5.1 below.)

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. 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 URN 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 syntax and semantics of these <NSS> values are often carried over from an existing persistent identifier system (namespace); for instance, in the 'ISBN' URN Namespace, each NSS must be a valid ISBN. Some URN Namespaces may have strict rules for well formed NSSs, while

some others may be far more relaxed. There may also be significant differences regarding the identifier assignment process. The overall specification requirements and registration procedures for URN Namespaces are the subject of a dedicated companion document, BCP 66, which has been updated for conformance to BCP 26 and alignment with implementation experience RFC 3406bis [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg].

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 therefore MUST NOT appear unencoded in a NSS. This rule guarantees backwards compatibility with existing URN Namespaces and improves the compatibility of URN syntax with general URI parsers.

For more specifics on the <query> part with URNs, see Section 2.3 below; elaborations on the <fragment> part usage with URNs follow in Section 2.4 below.

2.1. Namespace Identifier (NID) Syntax

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

```
NID = (ALPHA / DIGIT) 0*30(ALPHA / DIGIT / "-") (ALPHA / DIGIT)
```

Note:

The above definition is slightly more restrictive than it was in RFC 2141, to better reflect common practice for "handle"-like identifiers in other IETF protocols (a.k.a. "LDH" syntax) and requirements from RFC 3406bis. RFC 3406bis contains further syntax restrictions on NID strings.

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

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

The format of this single canonical form follows:

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

(<pchar> and <segment-nz> are defined in Section 3.3 of RFC 3986.)

Note:
The informational Appendix C expands on the evolution of the NSS syntax specification since RFC 2141.

Depending on the rules governing a namespace, valid identifiers in a namespace might contain characters that are not members of the URN character repertoire above (<pchar>). In order to achieve conformance with this NSS specification, such strings MUST be translated into canonical NSS format before embedding them into a URN, using them as protocol elements, or otherwise passing them on to other applications. Translation is done by encoding each character outside the URN character repertoire 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 latter two characters form the hexadecimal representation of that octet. (See Section 2.5.2 below for more details.)

2.3. Query Part in URI References to URNs

The <query> part MUST NOT be present in any *assigned* URN. A <query> 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 the spirit of the general specification of this part in RFC 3986, its purpose is restricted to indicate the requested URN resolution service and particular service aspects of the intended resolution response, e.g., the kind of metadata or content sought that are bound to a given object identified by the basic, assigned URN.

This specification only defines a generic framework format for this part and basic items to be used therein; it defers more detailed specifications to future standardization related to generic URN services and resolution and to URN Namespace defining documents for namespace-specific usages.

Beyond following the generic syntax rules from [RFC3986] quoted above, <query> parts of URN references MUST adhere to the following restricted syntax (compatible with industry standard URL-encoding practice for HTTP).

```
   urn-query = directive *( "&" directive)

   directive = keywd "=" value
```

```
keywd     = ALPHA *( ["-"] (ALPHA / DIGIT))
value     = *v-pchar

v-pchar   = unreserved / pct-encoded / v-subdels
v-subdels = "!" / "$" / "'" / "(" / ")"
          / "*" / "+" / "," / ";" / "="
          / ":" / "@" / "/" / "?"
; this is <sub-delims> except "&"
; plus the extra characters allowed in <pchar>
; and for <query>, as per RFC 3986
```

The <query> part of URN references, if present, consists of an unordered sequence of "directives" of the form <keywd>=<value>, separated by single instances of the ampersand character ("&"). As common for <query> parts, these directives are regarded as case-sensitive.

The <keywd> tokens are -- preferably short, mnemonic -- LDH-strings of either global or namespace-centric scope. The following subsections specify two basic keywords of global scope. Other <keywd> tokens can be specified in other documents (including URN Namespace specifications) and are to be registered by IANA. See Section 9.2.1 for registration details.

Each registered <keywd> MUST NOT appear more than once in any URN. A URN resolver that receives a URN reference violating this rule MUST ignore all query directives therein using the offending <keywd>(s) (this is necessary to maintain independence of the semantics from directive ordering).

The <value> tokens have semantics specific to the <keywd> they are used with. The above syntax rule is the most liberal possible specification guaranteeing unambiguity and still conforming with RFC 3986, but prudent specifications of <keywd> tokens will keep the <value> forms admitted with them as simple as possible.

URN resolvers are expected to ignore query keywords they do not support or understand and "gracefully" fall back to namespace-specific default behavior. Similarly, unless specified otherwise in the specification of a particular query keyword, URN resolvers are expected to ignore directives with an unknown/unsupported <value> for a supported <keywd> and provide default behavior for these cases as well as if an expected directive is not supplied. New/revised URN Namespace specifications need to clearly indicate which <keywd>s are being supported for the respective URN Namespace and the set of valid <value>s for these (by listing enumerated values and/or specifying additional syntax rules) -- see RFC 3406bis for more information.

2.3.1. Query Instruction for URN Service Selection

The query keyword "s" has global scope and semantics; it serves to select a specific URN resolution service. The associated <value> is the mnemonic name of the URN resolution operation intended by the URI reference to a URN. Permissible values are registered with IANA by the documents specifying these URN services -- see Section 9.2.2 for details. Pending future revised URN service specifications, the registry is initially populated with provisional entries derived from RFC 2483 [RFC2483].

This query keyword is expected to be supported by new URN resolution systems for any URN namespaces. A URN resolver that does not support this query keyword (e.g., because it is based on RFC 2141) or that does not understand the <value> handed to it MUST gracefully fall back to provide the default service for the respective URN Namespace, as specified in the related URN Namespace definition. New/revised URN Namespace specifications need to clearly indicate which services are being supported for the respective URN Namespace -- see RFC 3406bis for more information.

Example directive (URI to URL service, RFC 2483, Sec. 4.1): s=I2U

2.3.2. Query Instruction for Component Resource Indication

The query keyword "c" serves to identify a component of the resource named by the basic URN in a uniform, media type independent manner; it applies to structured resources only and otherwise has global scope, but namespace-specific applicability, values, and semantics.

URN Namespace designers/maintainers MAY adopt the use of this query instruction for their resolver systems and need to specify that fact in the URN Namespace registration and supply the applicable rules for the "c=" values to be supported by the resolvers for that URN Namespace. See RFC 3406bis for more information.

Hypothetical example: Assuming that the ISBN Namespace adopts support of the "c=" query instruction for the I2R URN service provided by URN:ISBN resolvers, further assuming that for a printed book the table of contents is anyway being made available online somewhere by its publisher _and_ the resolver system is aware of this, and provided that the resolvers support designation of the table of contents via the "c" <value> "ToC", a URI reference to the URN:ISBN of such book might indicate the intent to resolve the URN to an URL for that ToC by including the <query> part "s=I2L&c=ToC".

2.4. Fragment Part in URI References to URNs

The <fragment> part is not generally allowed in URNs. It is only applicable to URN Namespaces that specifically opt to support its usage in a manner that conforms with RFC 3986. 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 for that particular URN Namespace MUST NOT contain a fragment identifier.

The <fragment> part MUST NOT be present in any *assigned* URN; it MAY be present in a URI *reference* to a URN that is intended to be used with URN resolution services, and -- according to RFC 3986 -- it will not be sent to the resolution service but be interpreted by the resolution client in accordance with the specification of the Internet media type returned by the URN service.

> Note that this is a backwards-compatible and fail-safe extension from RFC 2141 since, based on RFC 3986 and established implementation practice, clients/browsers ignore inapplicable fragment identifiers and silently fall back in such case to rendering the entire resource returned.

The requirements for documenting the usage of fragment identifiers with a particular URN Namespace are elaborated upon in RFC 3406bis [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg], and that document also explains the different methods available to URN Namespace designers for how URN assignment and resolution can deal with structured resources and their components.

2.5. Special and Reserved Characters

The remaining printable characters not included in the <pchar> repertoire 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.

2.5.1. 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 the <pchar> rule 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 literally 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.

2.5.2. The Percent Character, Percent-Encoding

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 characters together semantically represent 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 repertoire as having special meaning for that namespace (e.g. as being used as a separator character between distinguishable parts of the NSS). If such namespace also allows for such character to occur in identifiers from that namespace in a literal sense (in a

part of the identifier that shall be embedded literally into the NSS), the character used in a literal sense MUST be percent-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 -- cf. RFC 3406bis [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg].

2.5.3. Other Excluded Characters

The following list is included only for the sake of completeness. It includes the characters discussed in Sections 2.5.1 and 2.5.2. Any octets/characters on this list are explicitly NOT part of the URN <NSS> character repertoire, and if used in an URN, MUST be percent-encoded.

```
excluded = CTL / SP          ; control characters and space
         / DQUOTE            ; "
         / "#"               ; from <gen-delims>
         / "%"               ; see above
         / "/"               ; from <gen-delims>
         / "<" / ">"
         / "?"               ; from <gen-delims>
         / "["               ; from <gen-delims>
         / "\"
         / "]"               ; from <gen-delims>
         / "^"
         / "`"
         / "{" / "|" / "}"
         / %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 URNs, in either unencoded or percent-encoded form.

In a textual context for a URN, the NSS part 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 NSS.

The more general issue of discerning URNs in non-structured text is not specific to URNs, but a general issue for recognizing URIs (by humans or automata), and hence out of scope of this document.

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

Any identifier to be used as a URN MUST be expressed in conformance with the URI and URN syntax specifications ([RFC3986], and this document). If names from (existing or newly devised) 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.

On the other hand, every namespace specific string in such URN Namespace MUST be based on an identifier that conforms to the requirements of the identifier system to which the URN Namespace is assigned; in the simplest form, if the syntactical rules admit, the NSS can be the original identifier. For instance, every legal NSS in the ISBN Namespace must be a valid ISBN.

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 -- even in a manner specific to particular URN Namespaces -- display of URNs in a more human-friendly form and MAY use in that context 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).

Note: Such transformation for the purpose of presentation, if done blindly without NID-specific knowledge of special character usage, might introduce ambiguity, because in the cases described above in the second paragraph of Section 2.5.2, the unescaped and percent-escaped form of the same character might carry different semantics in NSSs of some URN Namespaces.

5. Lexical Equivalence of URNs

For various purposes such as caching, it is often desirable to determine whether two URNs are "the same" (i.e., designate the same resource), without resolving them. The general-purpose means of doing so is by testing for "lexical equivalence" as defined below. This procedure only can detect mismatches: two lexically different URNs might still be assigned to the same reource -- be it by assignment practice within a single URN Namespace or by a single resource having assigned names from different URN Namespaces.

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.
4. Remove the <fragment> part of the URI, if present.
5. Depending on the objective, perform either step 5a or step 5b:
   If the objective is related to distinguishing named resources, perform step 5a; if the objective is related to caching specific URN resolution results, perform step 5b.
   5a. Remove the <query> part of the URI, if present.
   5b. Reorder the directives in the <query> part of the URI, if present, bringing them into a preferred <keywd> order.

Note that percent-encoding MUST NOT be removed. It is an implementation detail not affecting interoperability whether a function for lexical URN comparison internally prefers normalization (in the first 3 steps above) to lower or to upper case. Similarly, the "preferred order" in step 5b is an implementation choice without impact on interoperability.

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. Only URN software that is aware of such additional rules for a specific NID can detect these additional lexical equivalences

5.1. Examples of Lexical Equivalence

The following hypothetical URN comparisons highlight the lexical equivalence definitions (assuming that the hypothetical 'foo' namespace does not define additional lexical equivalences):

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
7- urn:foo:a123,456?x=y
8- urn:foo:a123,456#xyz

URNs 1, 2, 3 and 8 are all lexically equivalent, and URN 7 is also lexically equivalent to these if step 5a is applied, but this does not hold if step 5b above is applied instead. in the normalization.

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 within a given URN Namespace is determined by the management of URN assignment practices therein and established by the 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 URN Namespace, i.e., when two URNs are identical within a namespace.

On the other hand, it is permissible to have two entirely 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].

Therefore, 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, 3406bis, 3629 (STD 63), and 3986 (STD 66) ([RFC2169] [RFC2276] [RFC2608] [RFC3401] [RFC3402] [RFC3403] [RFC3404] [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg] [RFC3629] [RFC3986]).

7.1. Registration Template for 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 ("resource") 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, an update is in progress as RFC 3406bis [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg]).

Encoding considerations:

All URNs are ASCII strings conforming to the general URI syntax from STD 66. As described in Sections 2.2 and 2.5.2 of RFC XXXX, there may be characters allowed by the syntax and semantics of the identifier system underlying the URN Namespace but not contained in the US-ASCII charset. Such characters MUST first be represented in Unicode and encoded in UTF-8 according to STD 63. Any octets outside the allowed character set MUST then be percent-encoded.

Note that it is perfectly possible that the syntax and semantics of an underlying identifier system does not admit specific characters allowed by the syntax rules in RFC XXXX.

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; an update is in progress in the URNbis WG as RFC 3406bis [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg]).

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 particular characters generically allowed in the Namespace

Specific String, any security considerations resulting from such assignment are outside the scope of this document. It is REQUIRED by BCP 66 (currently [RFC3406], to be replaced by RFC 3406bis [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg]) that the process of registering a namespace identifier include any such considerations.

9. IANA Considerations

9.1. Registration of URI Scheme 'urn', URN Registry Update

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.

[[RFC-Editor: this para to be deleted before RFC publication.]] IANA is asked to again make available the URN Namespace Registry [IANA-URN] in a generic form (i.e., HTML) at the generic URI given in the Reference, and to make the XML and TXT versions available from that HTML version. (This state already had been achieved, but something seems to have been lost in 2011.)

9.2. URN Query Parameters Registry

IANA is asked to establish a new registry entitled "URN Resolution Query Parameters" with two sub-registries as described below, referencing Section 2.3 of this RFC as the authoritative source.

9.2.1. URN Query Keywords Sub-Registry

This registry holds the <keywd> tokens that can be used in the query part of URI references to URNs.

Entries capture the following items (that need to be provided by registration requests:

Keyword - the token to be used in the query part

Purpose - short phrase describing the purpose

Scope - either "global" or "specific"

Defn. Ref. - Reference to defining RFC

Supported by - list of {URN NID, reference} pairs

Keywords of "global" scope are (in principle) open for use with URNs and URN resolvers for any URN Namespace that choses to adopt it. The creation or substantive update of such entries requires a document containing the specification of the query directives using such keyword, subject to "IETF Review" (cf. BCP 26 [RFC5226]), and the change control of such entries remains with the IESG.

Keywords of "specific" scope are designed to fulfill the purposes of a specific URN Namespace or a specific group of URN Namespaces. The creation or substantive update of such entries requires a specification document subject to the procedures set out in RFC 3406bis for URN Namespace registration documents; the specification of the query directives using such keyword can be part of a URN Namespace registration documents. These entries remain under the change control of the stakeholders of the URN Namespace(s) given in the specification document.

Changes in the "Supported by" list of any registry entry is considered a non-substantive update. Additions will usually be performed by URN Namespace registration documents (cf. RFC 3406bis), but to reduce the overhead and encourage usage of this registry, the maintainers of legacy URN Namespaces (URN NIDs registered before the publication of this RFC), a URN NID and a non-RFC reference to a stable document can be added if the maintainers of the URN namespace demonstrate to IANA the usability of query directives with the respective keyword; for such requests, IANA may seek advice from the URN-NID experts as well.

Initial registrations:

Keywd	Purpose	Scope	Defn. Ref.
s	intended URN resolution service	global	RFC this, s. 2.3.1
c	component of structured resource	global	RFC this, s. 2.3.2

The "Supported by" lists for both entries initially are left empty.

9.2.2. URN Resolution Service Designators Sub-Registry

This registry lists the value tokens that can be used with the "s" keyword in the query part of URI references to URNs, in order to identify the desired URN resolution service.

Entries capture the following items (that need to be provided by registration requests:

Name - mnemomonic for the URN service

Purpose - short phrase describing the service

Status - "std", "exp","provisional", or "deprecated"

Reference - Reference to defining RFC

Registration policy is "RFC required" according to BCP 26 [RFC5226], where the RFC category required needs to match the desired "Status": Standards Track for "std", Experimental for "exp". Beyond the initial assignments performed below, "provisional" status can be assigned for pending registrations using the procedures of BCP 100 [RFC4020]. IESG Approval ([RFC5226]) is required to modify an entry to change its status to "deprecated".

In preparation for future work to update that document, the registry is initially populated with entries derived from Section 4 of RFC 2483 [RFC2483], using uniform spelling of plural forms and marking all entries as "provisional":

Name	Purpose	Status	Reference
I2L	URI to URL	provisional	RFC 2483, sec. 4.1
I2Ls	URI to URLs	provisional	RFC 2483, sec. 4.2
I2R	URI to resource	provisional	RFC 2483, sec. 4.3
I2Rs	URI to resources	provisional	RFC 2483, sec. 4.4
I2C	URI to URC	provisional	RFC 2483, sec. 4.5
I2Cs	URI to URCs	provisional	RFC 2483, sec. 4.6
I2N	URI to URN	provisional	RFC 2483, sec. 4.7
I2Ns	URI to URNs	provisional	RFC 2483, sec. 4.8
I=I	URI equal to URI?	provisional	RFC 2483, sec. 4.9

10. Acknowledgements

This document is heavily based on RFC 2141 by Ryan Moats, 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] authored by Karen

Sollins and Larry Masinter. The experiences gathered during the first (more than a) decade of URN usage were also helpful, so individuals and organizations which have implemented and used URNs are also acknowledged. In particular, the experience gained with parties wanting to make use of the URN framework and submit URN Namespace registration documents, and their desire to obtain the necessary collected background information has motivated and shaped the text put into Section 1 of this document.

Many individuals in the URNbis working group have participated in the detailed discussion of this memo. Particular thanks for detailed review comments and text suggestions go to Juha Hakala, Mykyta Yevstifeyev, Peter Saint-Andre, Subramanian Moonesamy, Bengt Neiss, and Lars Svensson.

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.

[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.

11.2. Informative References

[I-D.ietf-urnbis-rfc3406bis-urn-ns-reg] Hoenes, A., "Uniform Resource Name (URN) Namespace Definition Mechanisms", draft-ietf-urnbis-rfc3406bis-urn-ns-reg-03 (work in progress), October 2012.

[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.

[RFC2483] Mealling, M. and R. Daniel, "URI Resolution Services Necessary for URN Resolution", RFC 2483, January 1999.

[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.

[RFC4020] Kompella, K. and A. Zinin, "Early IANA Allocation of Standards Track Code Points", BCP 100, RFC 4020, February 2005.

Appendix A. 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.

However note that, according to RFC 3986, the <fragment> part of a URN will be stripped by a resolver client before passing the URN to the resolver, and subsequently be applied to the returned result -- in the manner specified for the returned media type.

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 B. Collected ABNF (Informative)

As a service to implementers specifically interested in URN syntax, the complete ABNF for URNs is collected here, including the referenced rules from [RFC5234] and [RFC3986]. In case of (unexpected) inconsistencies, these documents remain normative for the respective productions.

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

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

scheme        = ALPHA *( ALPHA / DIGIT / "+" / "-" / "." )
path-rootless = segment-nz *( "/" segment )
query         = *( pchar / "/" / "?" )
fragment      = *( pchar / "/" / "?" )

segment-nz    = 1*pchar
segment       = *pchar
pchar         = unreserved / pct-encoded / sub-delims / ":" / "@"

unreserved    = ALPHA / DIGIT / "-" / "." / "_" / "~"
pct-encoded   = "%" HEXDIG HEXDIG
sub-delims    = "!" / "$" / "&" / "'" / "(" / ")"
              / "*" / "+" / "," / ";" / "="
```

In the case of URNs, the above rules are subject to more specific restrictions, specified in Section 2 of this RFC:

```
scheme        = "urn"
                 ; specific, fixed (assigned) value

urn-path      = NID ":" NSS
                 ; to be superimposed on <path-rootless>,
                 ; which needs to be <segment-nz> only

NID           = ( ALPHA / DIGIT ) 1*31( ALPHA / DIGIT / "-" )
                 ; RFC 3406[bis] contains more specific rules

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

urn-query     = directive *( "&" directive)
                 ; to be superimposed on <query>

directive     = keywd "=" value
keywd         = ALPHA *( ["-"] (ALPHA / DIGIT))
value         = *v-pchar

v-pchar       = unreserved / pct-encoded / v-subdels
v-subdels     = "!" / "$" /       "'" / "(" / ")"
              / "*" / "+" / "," / ";" / "="
                 ; this is equivalent to <sub-delims> except "&"
              / ":" / "@" / "/" / "?"
                 ; plus the extra characters allowed in <pchar>
                 ; and for <query>, as per RFC 3986
```

The above rules make use of the following "Core Rules" from Appendix B.1 of [RFC5234] :

```
ALPHA         =  %x41-5A / %x61-7A   ; A-Z / a-z
DIGIT         =  %x30-39             ; 0-9
HEXDIG        =  DIGIT / "A" / "B" / "C" / "D" / "E" / "F"
```

Appendix C. Breakdown of NSS Syntax Evolution since RFC 2141 (Informative)

In order to make visible the detailed migration path from RFC 2141 and the influence of the evolution of URI syntax from RFC 2396 to RFC 3986 on it, this appendix provides a highly annotated and expanded version of the NSS syntax provided in Section 2.2:

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

In particular, the breakdown below serves to provide evidence of that this syntax correctly reflects the addition of "&" and "~" to the repertoire of characters allowed in the NSS portion of URNs previously allowed by RFC 2141; it expands on the syntax specified in RFC 2141 after translation to standard ABNF.

```
      NSS         = 1*URN-char

      URN-char    = trans / pct-encoded
 ; Note that <pct-encoded> from RFC 3986 here replaces the
 ; explicit, expanded form used in RFC 2141.

      trans       = ALPHA / DIGIT / u-other
 ; Note that RFC 2141's <other> has been disambiguated here
 ; into <u-other>.
 ; RFC 2141 also said:
 ;                / reserved
 ; This caused an ambiguity in RFC 2141 with respect to "%", which
 ; now is resolved here by omission of this dangling alternative.
 ;
 ; After adoption of the generic URI syntax from RFC 3986, there
 ; is no more need to deal here with the higher-level separator
 ; characters "/", "?", and "#" contained in <reserved>
 ; (beyond "%", which is fully taken care of by <pct-encoded>),
 ; which are part of RFC 3986's <gen-delims>, as shown below.

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

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

                  / "!" / "$" /       "'" / "(" / ")"
                  / "*" / "+" / "," / ";" / "="
                    ; this is RFC 3986 <sub-delims> except "&".
 ; From RFC 3986:
 ;    sub-delims  = "!" / "$" / "&" / "'" / "(" / ")"
 ;                / "*" / "+" / "," / ";" / "="
 ; The URNbis WG arrived at unanimous consensus that "&" can be
 ; allowed without harm to backward compatibility for existing
 ; URN Namespaces.

                  / "-" / "." / "_"   ; <unreserved> except "~"
 ; From RFC 3986:
```

```
    ;    unreserved  = ALPHA / DIGIT
    ;                / "-" / "." / "_" / "~"
    ; The URNbis WG arrived at unanimous consensus that "~" can be
    ; allowed without harm to backward compatibility for existing
    ; URN Namespaces.

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

    ; From RFC 3986:
    ;    segment-nz  = 1*pchar
    ;    pchar       = unreserved / pct-encoded / sub-delims / ":" / "@"
```

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.]]

Expanded Introduction to cover background material frequently requested by interested parties not well acquainted with RFCs and past/present work in the IETF, in particular prospective URN Namespace stakeholders and applicants for URN Namespace registrations. The material included also serves to avoid normative downrefs to legacy RFCs that are very unlikely to be progressed on the Standards Track in the foreseeable future.

Document references updated and split; Normative References now only to Full Internet Standards to allow for future progress of this memo on the IETF Standards Track.

Formal syntax now specified using ABNF (STD 68), using productions from Generic URI Syntax (STD 66) and STD 68.

NID Syntax slightly more restrictive than in RFC 2141 (compatible with existing and in-progress NID registrations).

NSS syntax now allows "&" and "~" to align URN syntax with generic <pchar> rule from STD 66; an ambiguity in the formal rules and incompatibilities between the formal rules and the prose description in RFC 2141 have been straightened out ("%" no more allowed outside percent-encoding triples, other <reserved> characters no more admitted by formal syntax rules).

Use of query and fragment part with URNs now specified, mostly by reference to STD 66. Syntactical pattern for query part defined; IANA registry for query keywords in URN references established.

This document also performs the outstanding formal registration of the 'urn' URI scheme.

Supplemental material in Appendices documents considerations and decisions made in the development 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.2.

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

Added new s1.2 to Introduction, with excerpts from RFC 1737 to provide background on URN functional and syntax requirements. Renumbered previous s1.2 and s1.3 to s1.3 and s1.4, respectively.

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.

D.5. Changes from WG Draft -00 to WG Draft -01

Reflect WG consensus on adding "&" and "~" to the set of characters allowed in the NSS part of URNs, thus aligning URN syntax with generic URI syntax from RFC 3986.

Moved breakdown of NSS syntax evolution from s2.2 to new Appendix C.

Avoid "[URN] character set" in favor of "character repertoire" to minimize potential clashes with IETF terminology on charsets.

s2.3.3: URN recognition in text documents is regarded out of scope.

The previous version was ambiguous on whether eventual query and/or fragment parts were regarded as part of the NSS; after closer inspection of the syntax, clarification has been added that the <urn-path> syntax is indeed superimposed on the <segment-nz> ABNF rule for URNs, and hence does not cover the trailing higher level parts (query, fragment) according to the URI syntax.

Filled in Appendix B contents.

Numerous editorial and grammar improvements.

D.6. Changes from WG Draft -01 to WG Draft -02

Added note at the beginning of Section 1.3 highlighting the purpose of this section. The URNbis charter excludes a revision of RFC 1738, and hence the changes suggested on the list to alter and update this section have been dismissed.

Added hint to URN Namespace designers in Section 2 that ":" is customarily used in URN Namespaces to provide further level(s) of hierarchical subdivision of NSSs.

Reworked text on fragment identification issues and resulting specification, mostly based on Juha Hakala's evaluation of the consensus evolving from the list discussion.

Modified ABNF rule for NIDs to better align it with rules for similar identifiers used in IETF protocols. The new rule now prohibits a trailing hyphen, but defers further restricting rules on NID syntax (based on the kind of NID) to RFC 3406bis.

More clearly documented and marked (still open / already closed) ISSUES. The related text will be removed in the next draft version, whence it should have been transferred into the IETF issue tracking system.

Text of Section 3 revised, based on Juha's suggestion.

In Section 5, added removal of <query> part (but not <fragment> part) to canonicalization steps for the purpose of determining lexical equivalence of URNs (Juha's comment). Also added examples showing this.

Elaborated a bit more on Encoding Consideration in the URI Scheme registration template (Juha's comments).

Numerous editorial corrections and improvements.

D.7. Changes from WG Draft -02 to WG Draft -03

Added text in s1.1 to reflect a comment from SM on other, legacy interpretations of "URN".

Added note in old s1.2 to reflect importance of the name binding established by a URN (derived from list discussion on other topic,

Keith Moore et al.).
However, (despite comments from SM and PSA) preserved excerpts there to keep document self-contained and avoid normative down-references (as discussed during WG chartering process and pointed out in the third para of old s1.3). Doing so should also help to avoid another future recurrence of the discussion on these topics that has consumed a lot of resources unnecessarily during the WG formation process.

Swapped s1.2 and s1.3 (note from SM); however, for logical reasons, motivation (part of s1.1) needs to stay in the text before the objectives derived thereof (now s1.2).

Material on query part enhanced (new subsection 2.3); structure of query part formally specified with a rather liberal syntax (could be more restrictive, if WG prefers); IANA registry of URN query keywords established, with two initial entries for the global scope "s" and "c" keywords now specified in s2.3.1 and s2.3.2.

To avoid further confusion (as seen on the list discussion), this I-D uses the term "fragment" only for the trailing component in the Generic URI Syntax and the semantics associated with it in RFC 3986; otherwise this I-D talks about "components" of structured resources.

Material on fragment part heavily revised and stripped down, put in new subsection 2.4. New text is intended to reflect least common denominator of list discussion; i.e., mostly just enable usage by specific URN Namespace and otherwise point to RFC 3986 and RFC 3406bis.
Namespace designers now have three options to design-in component resource designation (if warranted for the namespace), whichever is the best fit for their underlying identifier system: (1) media-specific designation using fragment part, (2) media-independent, abstract designation using query part (to be dealt with by resolution system, not resolution client), and (3) media-independent designation via assignment of distinct NSSs to component resources.
(That is being elaborated upon to a greater extent in the -03 version of the rfc3406bis I-D.)

Added text to percent-encoding considerations (Bengt Neiss' concerns).

Amended text on support of existing identifier systems (s3), based on various comments received.

Revised part of text in s5 and s6 on lexical/functional equivalence to reflect the new specification for query and fragment (new s2.3, s2.4) and to address several comments received; changed s5.1 accordingly.

In spite of the challenges raised by serious evidence of improper management practices for the ISBN system and hence the URN:ISBN Namespace (Lars Svensson), the I-D still contains one (hypothetical) example based on URN:ISBN; this is being thought acceptable because it is in the tradition of earlier documents and we can expect that every potential reader of the memo will have an understanding what ISBNs are for (or should be).

Modified title of s7.1 to avoid clash with new s9.1. Added IANA Considerations for "URN Query Parameters" registries (s9.2).

Acknowledgements expanded.

Amended Appendix A with text regarding <fragment> usage.

Filled in details in Appendix D.1; added this Appendix D.7.

Former Appendix E (guide to IETF document repositories) and pointer to it removed (comment from SM).

Multiple editorial enhancements and fixes.

Author's Address

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

EMail: ah@TR-Sys.de