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 BCP 66, for which RFC 3406bis is the companion revised specification document replacing RFC 3406.

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/>.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 13, 2012.
# Table of Contents

1. Introduction .................................................. 4
   1.1. Historical Perspective and Motivation ...................... 4
   1.2. Background on Properties of URNs .......................... 4
   1.3. Objective of this Memo .................................. 7
   1.4. Requirement Language ................................... 8
2. URN Syntax .................................................... 8
   2.1. Namespace Identifier (NID) Syntax ......................... 13
   2.2. Namespace Specific String (NSS) Syntax ................... 15
   2.3. Special and Reserved Characters ......................... 15
      2.3.1. Delimiter Characters ................................ 16
      2.3.2. The Percent Character, Percent-Encoding ............. 16
      2.3.3. Other Excluded Characters ......................... 17
3. Support of Existing Legacy Naming Systems and New Naming Systems ......... 18
4. URN Presentation and Transport ................................ 18
5. Lexical Equivalence of URNs ................................... 18
   5.1. Examples of Lexical Equivalence .......................... 19
6. Functional Equivalence of URNs ................................ 19
7. The 'urn' URI Scheme .......................................... 20
   7.1. Registration of URI Scheme 'urn' .......................... 20
8. Security Considerations ........................................ 20
9. IANA Considerations ........................................... 22
10. Acknowledgements .............................................. 23
11. References .................................................... 23
   11.1. Normative References .................................... 23
   11.2. Informative References .................................. 24
Appendix A. Handling of URNs by URL Resolvers/Browsers .................. 26
Appendix B. Collected ABNF (Informative) .......................... 26
Appendix C. Breakdown of NSS Syntax Evolution since RFC 2141 (Informative) .... 27
Appendix D. Changes since RFC 2141 (Informative) .................. 29
   D.1. Essential Changes from RFC 2141 .......................... 29
   D.2. Changes from RFC 2141 to Individual Draft -00 ........... 29
   D.3. Changes from Individual Draft -00 to -02 ................ 30
   D.4. Changes from Individual Draft -02 to WG Draft -00 ....... 30
   D.5. Changes from WG Draft -00 to WG Draft -01 ............... 30
   D.6. Changes from WG Draft -01 to WG Draft -02 ............... 31
Appendix E. How to Locate IETF Documents (Informative) ............... 32
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.

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 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 E gives hints on how to obtain RFCs and related information.

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 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 (Uniform Resource Locators) 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. 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 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 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. [...]

o Extensibility: The URN scheme permits future extensions.

o Independence: It is solely the responsibility of a name issuing authority to determine the conditions under which it will issue a name.

o 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 relax unrealistical detail requirements:

o 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.

o Simple comparison: A comparison algorithm for URNs is simple, local, and deterministic. [...]

o 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 usage of 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.

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 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].

1.4. 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 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 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.

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

\[
\text{URI} = \text{scheme "::" path-rootless [ "?" query ] [ "#" fragment ]}
\]

\[
\text{path-rootless} = \text{segment-nz }*/( "/" \text{segment})
\]
segment-nz = 1*pchar
group = *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 practice 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 2.3.1 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 ordinarily specified by 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 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].
Notes:

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

URNs have been in use for more than a decade. Some user communities want 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 the context of their focal URNs. Therefore, this document allows the designers of selected URN Namespaces to specify the use of the <fragment> component with URNs belonging these Namespaces, whereas the specification of usage of the <query> component is set aside to future standardization efforts for URN resolution. Thus, this draft allows both of these components in the general syntax.

ISSUE:

Regarding fragment identifiers, Section 3.5, para 1 of RFC 3986, indicates that "The fragment identifier ... allows indirect identification of a secondary resource by reference to a primary resource and additional identifying information. The identified secondary resource may be some portion or subset of the primary resource, some view on representations of the primary resource, or some other resource defined or described by those representations." RFC 3986 continues in specifying that the details of the interpretation of fragment identifiers are specific to the media types returned upon resolution of an URI. The entirety of the purposes mentioned in the above quote obviously only can be achieved fully if the "consumer" of the URI becomes aware of the fragment identifier as part of the requested URI, since, e.g., secondary resources might consist in representations might only be available in particular media types. However, RFC 3986 subsequently (in the penultimate paragraph of Section 3.5) specifies that the evaluation of fragment identifiers be a client-side matter and browsers are to strip them from request URIs sent in information retrieval protocols. Based on this, contemporary web browsers do not communicate fragment identifiers to the web server but perform fragment selection locally on the returned (HTML) resource. To make things even more complicated, the most popular media type (HTML) does only allow to set markers (which are anchor points in the serialized media stream and used by browsers to identify a specific position in the content) and does not allow browsers to
regularly identify actual, conceptional fragments of the media delivered -- like, e.g., the "proper content" of a web page, excluding navigation bars etc. -- so that in practice users have got accustomed to understanding a "fragment" as actually designating a "position" in the media, not a "part" of it.

Therefore, potential usage of <fragment> components in URNs is rather limited and has to be considered very seriously by designers of URN Namespaces that would like to make use of them. URN Namespaces that rely on (unmodified) browser resolution via HTTP/HTML cannot rely on the usage of fragment identifiers to steer the resolution process. Thus, the use of fragment identifiers only seems to be useful for URN Namespaces that are intended to either (a) exclusively make use of resolution systems / clients that can cope with handing off a full-featured URN (including a possible fragment identifier) to the resolution service, or (b) exclusively employ HTML/HTTP based resolution systems / clients, i.e., where the resolution results are returned as HTML such that web browsers can perform the fragment selection, or as some other media type that better supports the identification and actual selection of embedded fragments, even in off-the-shelf web browsers -- perhaps possible for certain variants of XML-based media types.

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.

The <query> part MUST NOT be present in any "assigned" URN. This specification reserves its use for future standardization related to URN services and resolution. 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 accordance with the general specification of this part in RFC 3986, its purpose is restricted to indicate the requested URN resolution service and/or particular service aspects of the intended resolution response, e.g., to select the kind 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 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 such that there is a need to make reference of parts of such resources in typical network access environments without incurring the effort to assign and maintain different (assigned) NSSs in such cases.

URN Namespaces will deal with various kinds of fragments. For instance, publications can be divided into smaller parts -- journals consist of volumes, issues and articles, and books may contain chapters. These logical fragments are usually not fragments in the sense of the deliberations in the URI Generic Syntax, and if so, <fragment> MUST NOT be used. However, namespaces MAY have internal means for identification of logical fragments such as journal articles. For instance, the ISBN (International Standard Book Number) system allows assignment of ISBN numbers to book chapters if they are available as separate items. Namespace specific fragment identification practices are beyond the scope of this document, since they do not rely on URI Generic Syntax, and their application is the primary RECOMMENDED way to deal with fragment identification. If a namespace lacks this possibility, a URN Namespace definition SHOULD define syntactical parts of its NSSs that amend the original identifiers of the underlying namespace in a readily parseable way and serve to allow assignment of URNs in that namespace to the intended abstract fragments. A URN Namespace registration MAY forbid all kind of fragment identification (even if it were possible on the basis of URI Generic Syntax), if the application rules and syntax of the identifier does not allow identification of fragments. ISSN (International Standard Serial Number) is an example of this kind of identifier / namespace.

The use of <fragment> as specified in RFC 3986 is possible if and only if (a) the URN Namespace is based on an existing identifier scheme that designates objects of reasonable complexity that there is a need to make reference of parts of such resources in typical network access environments; and (b) these parts will be identified in the canonical manner of the media type(s) delivered upon URN resolution. Direct resolution to them SHOULD be possible and sustainable.

If in a given namespace URNs are never assigned to a particular manifestation of a resource (for instance, a PDF version of a book), but can be transferred from one manifestation to the next or apply to all of them, <fragment> usage is forbidden. This applies also to the situation when identified resources are works (without any references
to physical embodiments of the work).
The use of <fragment> SHOULD NOT be opted for if the underlying namespace provides for the intrinsic possibility to identify such parts or if there is a readily usable method to construct NSSs by combining the existing identifiers with a component (or components) to identify such parts in an easily discernable manner.

Whether the URI Generic Syntax is applied or not, there are various ways in which fragment identifiers can be generated:

(a) Fragment identifiers (if any) are assigned individually to the relevant fragments of a larger entity during the URN assignment process. If a URN Namespace opts for this model, its specification SHOULD describe the additional syntax restrictions to be adhered to 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 automated 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 SHOULD 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 (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:

\[
\text{NID} = (\text{ALPHA} / \text{DIGIT}) 0^{*}30(\text{ALPHA} / \text{DIGIT} / "-") (\text{ALPHA} / \text{DIGIT})
\]
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.

ISSUE:
The above rule still allows NIDs that contain multiple adjacent hyphens or have the form of decimal numbers or decimal number ranges.

Should this be further restricted _in this document_ or is it sufficient to defer to the additional (NID kind specific) rules in RFC 3406bis and the common sense of URN Namespace authors and the designated IANA experts? Anyhow, such restrictions would be fully backward compatible -- as is the above tightened rule -- 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[bis].

The document editor senses the low level of discussion of this issue as an indication that this Issue can be closed.

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.

Note:
This reservation is carried over unchanged from RFC 2141, for historical reasons.

ISSUE:
Further possible reservations and/or details are out of scope for this document, but might be within the scope of RFC 3406bis. It has been suggested that no additional reservations should be codified and the final decision in any case should be left to the common sense of URN Namespace authors and the designated IANA experts.

The document editor senses the low level of discussion of this issue as an indication that this Issue can be closed.
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.

ISSUE (for the record):
In comparison to RFC 2141, essentially now "&" and "~" are allowed in the NSS syntax, in full conformance with the generic URI syntax. On the other hand, the <reserved> characters are no more part of the formal syntax -- unfortunately (or erroneously) these were included in the formal syntax rules of RFC 2141 and only excluded after that fact in the prose, which at least in one instance has lead to a URNNamespace definition document that allows <reserved> in the formal NSS syntax but does _not_ properly exclude their use in the prose. The interpretation of "%" was ambiguous in RFC 2141; it is now only allowed (in the formal syntax and in the prose) in <pct-encoded> constructs.

The document editor senses that this change of the NSS syntax has found consensus and that hence this Issue is regarded as closed.

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 2.3.2 below for more details.)

2.3. 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.

### 2.3.1. Delimiter Characters

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

\[\text{gen-delims} = ":" / "/" / "?" / "#" / ":[" / "]" / ":@\]

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 <segment> 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.

### 2.3.2. 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 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 repertoire 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 -- cf. RFC 3406bis [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg].

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 repertoire, and if used in an URN, MUST be percent-encoded.

```plaintext
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 Naming Systems 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], 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 a given 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 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).

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 2.3.2, 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 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;
4. remove the <query> part of the URI, if present.

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. Note also that <fragment> MUST NOT be removed, since there is no lexical equivalence between the "base" URN and one which uses <fragment> -- the former identifies the resource as the whole; the latter just a part of it.

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
7. urn:foo:a123,456?xyz
8. urn:foo:a123,456#xyz

URNs 1, 2, 3, and 7 are all lexically equivalent. URN 4 is not lexically equivalent to any of the other URNs of the above set. The same holds for URN 8. URN 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 URN 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, 3406bis, 3629 (STD 63), and 3986 (STD 66) ([RFC2196] [RFC2276] [RFC2608] [RFC3401] [RFC3402] [RFC3403] [RFC3404] [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg] [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, 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 2.3.2 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 protocols.

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

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.
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.)

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]. 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.

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 and Mykyta Yevstifeyev.

11. References

11.1. Normative References


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-02 (work in progress), March 2012.

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

[IANA-URI] IANA, "URI Schemes Registry",

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


RFC 1808, June 1995.


RFC 2169, June 1997.


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.

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]:

\[
\text{URI} = \text{scheme} "." \text{path-rootless} [ "?" \text{query} ] [ "#" \text{fragment} ]
\]

\[
\text{scheme} = \text{ALPHA} *( \text{ALPHA} / \text{DIGIT} / "+" / "-" / "." )
\]

\[
\text{path-rootless} = \text{segment-nz} *( "/" \text{segment} )
\]

\[
\text{query} = *( \text{pchar} / "/" / "?" )
\]

\[
\text{fragment} = *( \text{pchar} / "/" / "?" )
\]

\[
\text{segment-nz} = 1*pchar
\]

\[
\text{segment} = *pchar
\]

\[
\text{pchar} = \text{unreserved} / \text{pct-encoded} / \text{sub-delims} / ":" / "@"
\]

\[
\text{unreserved} = \text{ALPHA} / \text{DIGIT} / "-" / "." / "_" / "@"
\]

\[
\text{pct-encoded} = "%" \text{HEXDIG} \text{HEXDIG}
\]

\[
\text{sub-delims} = ":" / ":$" / ":&" / ":" / "(" / "")" / "*" / ":*" / ":" / ":;" / ":="
\]
In the case of URNs, the above rules are subject to more specific restrictions:

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

urn-path = NID ":" NSS ; to be superimposed on <path-rootless>

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

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

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 = ":" / ";" / "&" / "!' / "(" / ")"
; / "&" / "&" / ";" / ";" / ";=
; TheURNbis WG arrived at unanimous consensus that "&" can be
; allowed without harm to backward compatibility for existing
; URN Namespaces.

; From RFC 3986:
; unreserved = ALPHA / DIGIT
; / ":" / ";" / "&" / "~"
; TheURNbis 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. ]

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.

s2.3, s2.3.1*: 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 1.2 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.
Appendix E.  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/>.

Author's Address

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

EMail: ah@TR-Sys.de