URNBIS
Internet-Draft
Obsoletes: 2141 (if approved)
Intended status: Standards Track
Expires: January 13, 2014

P. Saint-Andre, Ed.
Cisco Systems, Inc.
R. Moats

July 12, 2013

Uniform Resource Name (URN) Syntax
draft-ietf-urnbis-rfc2141bis-urn-05

Abstract

A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) that is intended to serve as a persistent, location-independent resource identifier. This document defines the canonical syntax for URIs under the "urn" scheme, guidelines for URN namespaces, requirements for URN presentation and transmission, and methods for determining URN equivalence. This document obsoletes RFC 2141.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) [RFC3986] that is intended to serve as a persistent, location-independent resource identifier. This document defines the canonical syntax for URIs under the "urn" scheme, guidelines for URN namespaces, requirements for URN presentation and transmission, and methods for determining URN equivalence.

URNs were originally defined in [RFC2141]. The goal of this document is to specify URNs with the smallest reasonable set of changes from the original definition while ensuring consistency with the updated specification of URIs in [RFC3986].

This document obsoletes RFC 2141.

2. Terminology

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

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

3. Requirements

The requirements for URNs are specified in [RFC1737]. This document does not modify or update those requirements.

4. URN Syntax

The syntax for a URN is defined as follows using the Augmented Backus-Naur Form (ABNF) as specified in [RFC5234].

```
      namestring    = assigned-name [ "?" query ] [ "#" fragment ]
                    ;
                    ; query and fragment are defined in RFC 3986
                    ;
      assigned-name = "urn" ":" NID ":" NSS
                    ;
                    ; the URI scheme ("urn") is case-insensitive
                    ;
      NID           = (alphanum) 0*30(ldh) (alphanum)
                    ;
                    ; alphanum is defined in RFC 3986
                    ;
      ldh           = alphanum / "-"
      NSS           = 1*(pchar)
                    ;
                    ; pchar is defined in RFC 3986
                    ;
```

The following sections describe provide additional information about these rules.

4.1. Namespace Identifier Syntax

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

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

4.2. Namespace Specific String Syntax

Depending on the rules governing a namespace, names that are valid in a namespace might contain characters that are not allowed in URNs according to the "pchar" rule (e.g., characters outside the ASCII range or characters that are reserved in URIs, such as "/", "?", and "#"). Such a string MUST be translated into a conformant NSS before using it as a protocol element or otherwise passing it on to other applications. Translation is done by percent-encoding each disallowed character using the method defined in Section 2.1 of [RFC3986]. Note that the "%" character is allowed only for the purpose of percent-encoding.

If a namespace designates one or more characters conforming to the "pchar" rule as having special meaning for that namespace (e.g., "@") and the namespace also uses that character in a literal sense, when used in a literal sense the character MUST be percent-encoded (e.g., "%40"). For related considerations with regard to NID registration, see [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg].

4.3. Query Component and Fragment Identifier Component

The URI specification [RFC3986] allows a query component, a fragment identifier component, or both after the path component of a URI, where the character '?' is used as a separator to denote the beginning of the query component and the character '#' is used as a separator to denote the beginning of the fragment identifier component. The original URN syntax specification [RFC2141] reserved the '?' and '#' characters for future developments. This specification aligns URN syntax with URI syntax by allowing the query component and fragment identifier component after (not within) the Namespace Specific String (NSS).

This specification does not define the applicability and semantics of the query component or the fragment identifier component in URNs. Additional specifications might establish these matters for URN-related services (such as resolution) or for individual URN namespaces. For example, it is possible that the query component might be used in requests to URN resolution services, or that the fragment identifier component might be used to distinguish the integral parts of resources named by URNs. However, defining such usage is left to specifications for URN resolution services, namespace registration requests and specifications for individual namespaces (which might use some namespace-specific syntax instead of the URI fragment identifier component), and other appropriate documentation (such as policy documents governing the management of a given URN namespace).

Although URN assignment is often a managed process (see [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg]), the query component or fragment identifier component can be appended after the NSS once a URN has been assigned in accordance with the rules for a given namespace.

5. URN Presentation and Transport

The URN syntax defines the canonical format for URNs. 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 might support display of URNs in a more human-friendly form and might use a character set that includes characters that are not permitted in URN syntax as defined in this RFC (i.e., when displaying URNs to humans, such applications might replace percent-encoded strings with characters in an extended character set such as Unicode).

6. Lexical Equivalence in URNs

6.1. Procedure

For various purposes such as caching, often it is desirable to determine if two URNs are "the same". This is done by testing for "lexical equivalence".

Two URNs are lexically equivalent if they are octet-by-octet equal after applying case normalization (as specified in Section 6.2.2.1 of [RFC3986]) to the following constructs:

1. the URI scheme "urn"
2. the NID
3. any percent-encoded characters (see Section 2.1 of [RFC3986])

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

If a query component, fragment identifier component, or both have been appended to the assigned URI, they MUST be ignored for purposes of determining lexical equivalence.

URN namespaces MAY define additional rules for lexical equivalence, such as case-insensitivity of the NSS (or parts thereof). Such rules MUST always have the effect of eliminating some of the false negatives obtained by the procedure above and MUST NOT result in treating two URNs as not equivalent if the procedure here says they are equivalent. For related considerations with regard to NID registration, see [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg].

6.2. Examples

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

1. URN:example:a123,456
2. urn:example:a123,456
3. urn:EXAMPLE:a123,456
4. urn:example:A123,456
5. urn:example:a123%2C456
6. URN:EXAMPLE:a123%2c456

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

7. Functional Equivalence in URNs

Functional equivalence is determined within a given namespace and managed by resolvers for that namespace, and thus is beyond the scope of this document. For related considerations with regard to NID registration, see [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg].

8. Handling of URNs by URI Processors

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

A URN MUST be considered an opaque URI by URI resolvers and passed (with the "urn" scheme) to a URN resolver for resolution. The URN resolver can either be an external resolver that the URI resolver knows of, or it can be functionality built-in to the URI resolver.

To minimize user confusion, a URI browser SHOULD display the complete URN (including the "urn" scheme) to ensure that there is no confusion between URN namespace identifiers and URL scheme identifiers.

9. Security Considerations

This document specifies the syntax for URNs. While some namespaces resolvers might assign special meaning to certain of the characters of the Namespace Specific String, any security considerations resulting from such assignment are outside the scope of this document. For related considerations with regard to NID registration, see [I-D.ietf-urnbis-rfc3406bis-urn-ns-reg].

10. IANA Considerations

This section formally registers a URI scheme of ’urn’.

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

URI Scheme Name: urn
Status: permanent

URI Scheme Syntax: See Section 4 of RFCXXXX.
URI Scheme Semantics: The ’urn’ scheme identifies Uniform Resource Names, which are persistent, location-independent resource identifiers.
Encoding Considerations: See Section 4.2 of RFCXXXX.
Applications/Protocols That Use This URI Scheme Name: Uniform Resource Names are used in a wide variety of applications, including bibliographic reference systems and as names for Extensible Markup Language (XML) namespaces.
Interoperability Considerations: There are no known interoperability concerns related to use of the ’urn’ URI scheme.
Security Considerations: See Section 9 of RFCXXXX.
Contact: URNBIS WG [mailto:urn@ietf.org]
Author/Change Controller: This scheme is registered under the IETF tree. As such, the IETF maintains change control.
References None.

11. References

11.1. Normative References

[I-D.ietf-urnbis-rfc3406bis-urn-ns-reg] Saint-Andre, P., Daigle, L., Iannella, R., and P. Faltstrom, "Uniform Resource Name (URN) Namespace Definition Mechanisms", draft-ietf-urnbis-rfc3406bis-urn-ns-reg-06 (work in progress), July 2013.

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

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

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

11.2. Informative References

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

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

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

Appendix A. Changes from RFC 2141

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

- Allowed the URI query component after the URN as assigned.
- Allowed the URI fragment identifier component after the URN as assigned.
- Disallowed "-" at the end of a NID.
- Allowed the "~" and "&" characters in an NSS.
- Formally registered 'urn' as a URI scheme.

Appendix B. Acknowledgements

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

Many thanks to Leslie Daigle, Martin Duerst, Juha Hakala, Ted Hardie, Alfred Hoenes, John Klensin, Keith Moore, Julian Reschke, Lars Svensson, and other participants in the URNBIS WG for their input.

Julian Reschke provided helpful corrections to the ABNF.

Authors' Addresses

Peter Saint-Andre (editor)
Cisco Systems, Inc.
1899 Wynkoop Street, Suite 600
Denver, CO 80202
USA

Phone: +1-303-308-3282
Email: psaintan@cisco.com

Ryan Moats