URN Syntax
Filename: draft-ietf-urn-syntax-01.txt

Status of This Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

To learn the current status of any Internet-Draft, please check the "id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Abstract

Uniform Resource Names (URNs) are intended to serve as persistent resource identifiers. This document sets forward the canonical syntax for URNs. Support for both existing legacy and new namespaces is discussed. Requirements for URN presentation and transmission are presented. Finally, there is a discussion of URN equivalence and how to determine it.

1. Introduction

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

All URNs have the following syntax:

```plaintext
<URN> ::= ["urn:"] <NID> ":" <NSS>
```

<NID> is the Namespace Identifier, and <NSS> is the Namespace Specific String. The leading case-insensitive "urn:" sequence is currently optional, as no closure on its definite presence or absence has been reached. The Namespace ID is used to determine the _syntactic_ interpretation of the Namespace Specific String (as discussed in [1]).

RFC 1737 [2] presents additional requirements on URN encoding, which all have implications as far as limiting syntax. On the other hand, the requirement to support existing legacy naming systems has the effect of broadening syntax. Thus, we discuss the acceptable syntax for both the Namespace Identifier and the Namespace Specific String separately.

2.1 Namespace Identifier Syntax

The following is the syntax for the Namespace Identifier. To (a) be consistent with all potential resolution schemes and (b) not put any undue constraints on any potential resolution scheme, the syntax for the Namespace Identifier is:

```plaintext
<NID> ::= <letter> [ <let-hyp> ]
<let-hyp> ::= <letter> | "-" | <let-hyp>
<letter> ::= any one of the 52 alphabetic characters A through Z in upper case and a through z in lower case
```

This is slightly more restrictive that what is stated in RFC 1738 [4] (which allows the period "."). Further, the Namespace Identifier is case insensitive, so that "ISBN" and "isbn" refer to the same namespace.

To avoid confusion with the optional "urn:" identifier, the NID "urn" is reserved and may not be used.

2.2 Namespace Specific String Syntax

As required by 1737, there is a single canonical representation of the NSS portion of an URN. The format of this single canonical form follows:
<NSS> ::= <URN chars>*

<URN chars> ::= <trans> | "%" <hex> <hex>

<trans> ::= <upper> | <lower> | <number> | <other>

<hex> ::= <number> | "A" | "B" | "C" | "D" | "E" | "F"

<upper> ::= "A" | "B" | "C" | "D" | "E" | "F" | "G" | "H" | "I" | "J" | "K" | "L" | "M" | "N" | "O" | "P" | "Q" | "R" | "S" | "T" | "U" | "V" | "W" | "X" | "Y" | "Z"

<lower> ::= "a" | "b" | "c" | "d" | "e" | "f" | "g" | "h" | "i" | "j" | "k" | "l" | "m" | "n" | "o" | "p" | "q" | "r" | "s" | "t" | "u" | "v" | "w" | "x" | "y" | "z"

<number> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<other> ::= "(" | ")" | "+" | "," | ":" | ";" | "?" | "/" | ";" |

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

Namespaces MAY designate one or more characters from the URN character set as having special meaning for that namespace. If the namespace also uses that character in a literal sense as well, the character used in a literal sense must be encoded with "%" followed by the hexadecimal representation of that octet. Therefore, the process of registering a namespace identifier shall include publication of a definition of which characters have a special meaning and how to encode these characters if used in a literal sense.
3. Support of existing legacy naming systems and new naming systems

URN-aware applications MAY accept as input other resource identifiers from existing legacy namespaces. If such identifiers contain characters that are not members of the URN character set specified in section 2.2, the identifier MUST be translated to canonical format as discussed in section 2.2.

Some existing name spaces that have the properties of the URN-space contain some human-significant components, and these exist in a wide variety of languages. However, URNs are NOT intended to convey information that is significant to humans. While the translation rule in section 2.2 is provided for existing namespaces, new namespaces, as part of their registration documentation, MUST define a discipline for assigning new URNs that does not simplify the generation of human-significant names.

4. URN presentation and transport

URN-aware applications MAY support "natural" display of URNs which contain characters encoded using "%" notation. However, they MUST provide for display of URNs in canonical form (i.e. in a format suitable for transcription).

URNs may only be transported in canonical format.

5. Equivalence in URNs

URNs are considered equivalent if they return the same resource. For various purposes, such as caching, a test is necessary to determine equivalence without actually resolving the URNs and fetching/comparing the underlying resources. "Lexical equivalence" is a stricter condition that the equivalence described above (functional equivalence).

5.1 Lexical Equivalence

Lexical equivalence may be determined by comparing two URNs without making any network accesses. Two URNs are lexically equivalent if they are octet-by-octet equal after the following preprocessing:

1. drop any preceding "urn:" token
2. normalize the case of the NID

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 have the effect of eliminating some of the false negatives obtained by the procedure above, and MUST NEVER says that
two URNs are not equivalent if the procedure above says they are equivalent.

5.2 Functional Equivalence

Resolvers determine functional equivalence based on specific rules for the namespace. Therefore, namespace registration must include documentation on how to determine functional equivalence for that namespace.

5.3 Examples

The following URN comparisons highlight the difference between these types of equivalence:

\[
\begin{align*}
\text{urn:isbn:1-23485-8-29, isbn:1-23485-8-29} & \text{ are lexically equiv.} \\
\text{urn:isbn:1-23485-8-29, ISBN:1-23485-8-29} & \text{ are lexically equiv.} \\
\text{urn:isbn:1-23485-8-29, isbn:123485829} & \text{ are not lexically equiv.} \\
\text{but may be functionally equivalent.}
\end{align*}
\]

6. Security considerations

Because of the number of potential namespaces, it must be restated that certain of the characters in the Namespace Specific String may have special meaning to certain namespace resolvers. The process of registering a namespace identifier shall therefore include publication of a definition of which characters have a special meaning.

7. Acknowledgments

Thanks to various members of the URN working group and <<your name here!!>> for comments on earlier drafts of this document. This document is partially supported by the National Science Foundation.

8. References

Request For Comments (RFC) and Internet Draft documents are available from <URL:ftp://ftp.internic.net> and numerous mirror sites.


9. Editor’s address

Ryan Moats  
AT&T  
15621 Drexel Circle  
Omaha, NE 68135-2358  
USA

Phone:  +1 402 894-9456  
EMail:  jayhawk@ds.internic.net

This Internet Draft expires May 19, 1997.