

# Exhibit H

Network Working Group  
Request for Comments: 3986  
STD: 66  
Updates: 1738  
Obsoletes: 2732, 2396, 1808  
Category: Standards Track

T. Berners-Lee  
W3C/MIT  
R. Fielding  
Day Software  
L. Masinter  
Adobe Systems  
January 2005

## Uniform Resource Identifier (URI): Generic Syntax

### Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The Internet Society (2005).

### Abstract

A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource. This specification defines the generic URI syntax and a process for resolving URI references that might be in relative form, along with guidelines and security considerations for the use of URIs on the Internet. The URI syntax defines a grammar that is a superset of all valid URIs, allowing an implementation to parse the common components of a URI reference without knowing the scheme-specific requirements of every possible identifier. This specification does not define a generative grammar for URIs; that task is performed by the individual specifications of each URI scheme.

### 1.1.2. Examples

The following example URIs illustrate several URI schemes and variations in their common syntax components:

<ftp://ftp.is.co.za/rfc/rfc1808.txt>

<http://www.ietf.org/rfc/rfc2396.txt>

ldap://[2001:db8::7]/c=GB?objectClass?one

mailto:John.Doe@example.com

news:comp.infosystems.www.servers.unix

tel:+1-816-555-1212

telnet://192.0.2.16:80/

urn:oasis:names:specification:docbook:dtd:xml:4.1.2

### 1.1.3. URI, URL, and URN

A URI can be further classified as a locator, a name, or both. The term "Uniform Resource Locator" (URL) refers to the subset of URIs that, in addition to identifying a resource, provide a means of locating the resource by describing its primary access mechanism (e.g., its network "location"). The term "Uniform Resource Name" (URN) has been used historically to refer to both URIs under the "urn" scheme [RFC2141], which are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable, and to any other URI with the properties of a name.

An individual scheme does not have to be classified as being just one of "name" or "locator". Instances of URIs from any given scheme may have the characteristics of names or locators or both, often depending on the persistence and care in the assignment of identifiers by the naming authority, rather than on any quality of the scheme. Future specifications and related documentation should use the general term "URI" rather than the more restrictive terms "URL" and "URN" [RFC3305].

coordinates on a map. Likewise, a URI scheme may define components with additional encoding requirements that are applied prior to forming the component and producing the URI.

When a new URI scheme defines a component that represents textual data consisting of characters from the Universal Character Set [UCS], the data should first be encoded as octets according to the UTF-8 character encoding [STD63]; then only those octets that do not correspond to characters in the unreserved set should be percent-encoded. For example, the character A would be represented as "A", the character LATIN CAPITAL LETTER A WITH GRAVE would be represented as "%C3%80", and the character KATAKANA LETTER A would be represented as "%E3%82%A2".

### 3. Syntax Components

The generic URI syntax consists of a hierarchical sequence of components referred to as the scheme, authority, path, query, and fragment.

```
URI           = scheme ":" hier-part [ "?" query ] [ "#" fragment ]

hier-part     = "//" authority path-abempty
               / path-absolute
               / path-rootless
               / path-empty
```

The scheme and path components are required, though the path may be empty (no characters). When authority is present, the path must either be empty or begin with a slash ("/") character. When authority is not present, the path cannot begin with two slash characters ("//"). These restrictions result in five different ABNF rules for a path (Section 3.3), only one of which will match any given URI reference.

The following are two example URIs and their component parts:

```
foo://example.com:8042/over/there?name=ferret#nose
  \_/      \_____/ \_____/ \_____/ \_____/
  |         |         |         |         |
scheme  authority  path      query  fragment
  |         |         |         |         |
  / \      /_____| \_____|
urn:example:animal:ferret:nose
```

### 3.1. Scheme

Each URI begins with a scheme name that refers to a specification for assigning identifiers within that scheme. As such, the URI syntax is a federated and extensible naming system wherein each scheme's specification may further restrict the syntax and semantics of identifiers using that scheme.

Scheme names consist of a sequence of characters beginning with a letter and followed by any combination of letters, digits, plus ("+"), period ("."), or hyphen ("-"). Although schemes are case-insensitive, the canonical form is lowercase and documents that specify schemes must do so with lowercase letters. An implementation should accept uppercase letters as equivalent to lowercase in scheme names (e.g., allow "HTTP" as well as "http") for the sake of robustness but should only produce lowercase scheme names for consistency.

```
scheme      = ALPHA *( ALPHA / DIGIT / "+" / "-" / "." )
```

Individual schemes are not specified by this document. The process for registration of new URI schemes is defined separately by [BCP35]. The scheme registry maintains the mapping between scheme names and their specifications. Advice for designers of new URI schemes can be found in [RFC2718]. URI scheme specifications must define their own syntax so that all strings matching their scheme-specific syntax will also match the <absolute-URI> grammar, as described in [Section 4.3](#).

When presented with a URI that violates one or more scheme-specific restrictions, the scheme-specific resolution process should flag the reference as an error rather than ignore the unused parts; doing so reduces the number of equivalent URIs and helps detect abuses of the generic syntax, which might indicate that the URI has been constructed to mislead the user ([Section 7.6](#)).

### 3.2. Authority

Many URI schemes include a hierarchical element for a naming authority so that governance of the name space defined by the remainder of the URI is delegated to that authority (which may, in turn, delegate it further). The generic syntax provides a common means for distinguishing an authority based on a registered name or server address, along with optional port and user information.

The authority component is preceded by a double slash ("//") and is terminated by the next slash ("/"), question mark ("?"), or number sign("#") character, or by the end of the URI.