Network Working Group
Request for Comments: 1630
Category: Informational

T. Berners-Lee
CERN

June 1994

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

Status of this Memo

This memo provides information for the Internet community. This memo does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

IESG Note:

Note that the work contained in this memo does not describe an Internet standard. An Internet standard for general Resource Identifiers is under development within the IETF.

Introduction

This document defines the syntax used by the World-Wide Web initiative to encode the names and addresses of objects on the Internet. The web is considered to include objects accessed using an extendable number of protocols, existing, invented for the web itself, or to be invented in the future. Access instructions for an individual object under a given protocol are encoded into forms of address string. Other protocols allow the use of object names of various forms. In order to abstract the idea of a generic object, the web needs the concepts of the universal set of objects, and of the universal set of names or addresses of objects.

A Universal Resource Identifier (URI) is a member of this universal set of names in registered name spaces and addresses referring to registered protocols or name spaces. A Uniform Resource Locator (URL), defined elsewhere, is a form of URI which expresses an address which maps onto an access algorithm using network protocols. Existing URI schemes which correspond to the (still mutating) concept of IETF URLs are listed here. The Uniform Resource Name (URN) debate attempts to define a name space (and presumably resolution protocols) for persistent object names. This area is not addressed by this document, which is written in order to document existing practice and provide a reference point for URL and URN discussions.

The world-wide web protocols are discussed on the mailing list www-talk-request@info.cern.ch and the newsgroup comp.infosystems.www is preferable for beginner's questions. The mailing list uri-request@bunyip.com has discussion related particularly to the URI issue. The author may be contacted as timbl@info.cern.ch.

This document is available in hypertext form at:

http://info.cern.ch/hypertext/WWW/Addressing/URL/URI_Overview.html

The Need For a Universal Syntax

This section describes the concept of the URI and does not form part of the specification.

Many protocols and systems for document search and retrieval are currently in use, and many more protocols or refinements of existing protocols are to be expected in a field whose expansion is explosive.

These systems are aiming to achieve global search and readership of documents across differing computing platforms, and despite a plethora of protocols and data formats. As protocols evolve, gateways can allow global access to remain possible. As data formats evolve, format conversion programs can preserve global access. There is one area, however, in which it is impractical to make conversions, and that is in the names and addresses used to identify objects. This is because names and addresses of objects are passed on in so many ways, from the backs of envelopes to hypertext objects, and may have a long life.

A common feature of almost all the data models of past and proposed systems is something which can be mapped onto a concept of "object" and some kind of name, address, or identifier for that object. One can therefore define a set of name spaces in which these objects can be said to exist.

Practical systems need to access and mix objects which are part of different existing and proposed systems. Therefore, the concept of the universal set of all objects, and hence the universal set of names and addresses, in all name spaces, becomes important. This allows names in different spaces to be treated in a common way, even though names in different spaces have differing characteristics, as do the objects to which they refer.

URIs

This document defines a way to encapsulate a name in any registered name space, and label it with the the name space, producing a member of the universal set. Such an encoded and labelled member of this set is known as a Universal Resource Identifier, or URI.

The universal syntax allows access of objects available using existing protocols, and may be extended with technology.

The specification of the URI syntax does not imply anything about the properties of names and addresses in the various name spaces which are mapped onto the set of URI strings. The properties follow from the specifications of the protocols and the associated usage conventions for each scheme.

URLs

For existing Internet access protocols, it is necessary in most cases to define the encoding of the access algorithm into something concise enough to be termed address. URIs which refer to objects accessed with existing protocols are known as "Uniform Resource Locators" (URLs) and are listed here as used in WWW, but to be formally defined in a separate document.

URNs

There is currently a drive to define a space of more persistent names than any URLs. These "Uniform Resource Names" are the subject of an IETF working group's discussions. (See Sollins and Masinter, Functional Specifications for URNs, circulated informally.)

The URI syntax and URL forms have been in widespread use by World-Wide Web software since 1990.

Design Criteria and Choices

This section is not part of the specification: it is simply an explanation of the way in which the specification was derived.

Design criteria

The syntax was designed to be:

Extensible	New naming schemes may be added later.
Complete	It is possible to encode any naming scheme.
Printable	It is possible to express any URI using 7-bit ASCII characters so that URIs may, if necessary, be passed using pen and ink.

Choices for a universal syntax

For the syntax itself there is little choice except for the order and punctuation of the elements, and the acceptable characters and escaping rules.

The extensibility requirement is met by allowing an arbitrary (but registered) string to be used as a prefix. A prefix is chosen as left to right parsing is more common than right to left. The choice of a colon as separator of the prefix from the rest of the URI was arbitrary.

The decoding of the rest of the string is defined as a function of the prefix. New prefixed are introduced for new schemes as necessary, in agreement with the registration authority. The registration of a new scheme clearly requires the definition of the decoding of the URI into a given name space, and a definition of the properties and, where applicable, resolution protocols, for the name space.

The completeness requirement is easily met by allowing particularly strange or plain binary names to be encoded in base 16 or 64 using the acceptable characters.

The printability requirement could have been met by requiring all schemes to encode characters not part of a basic set. This led to many discussions of what the basic set should be. A difficult case, for example, is when an ISO latin 1 string appears in a URL, and within an application with ISO Latin-1 capability, it can be handled intact. However, for transport in general, the non-ASCII

characters need to be escaped.

The solution to this was to specify a safe set of characters, and a general escaping scheme which may be used for encoding "unsafe" characters. This "safe" set is suitable, for example, for use in electronic mail. This is the canonical form of a URI.

The choice of escape character for introducing representations of non-allowed characters also tends to be a matter of taste. An ANSI standard exists in the C language, using the back-slash character "\". The use of this character on unix command lines, however, can be a problem as it is interpreted by many shell programs, and would have itself to be escaped. It is also a character which is not available on certain keyboards. The equals sign is commonly used in the encoding of names having attribute=value pairs. The percent sign was eventually chosen as a suitable escape character.

There is a conflict between the need to be able to represent many characters including spaces within a URI directly, and the need to be able to use a URI in environments which have limited character sets or in which certain characters are prone to corruption. This conflict has been resolved by use of an hexadecimal escaping method which may be applied to any characters forbidden in a given context. When URLs are moved between contexts, the set of characters escaped may be enlarged or reduced unambiguously.

The use of white space characters is risky in URIs to be printed or sent by electronic mail, and the use of multiple white space characters is very risky. This is because of the frequent introduction of extraneous white space when lines are wrapped by systems such as mail, or sheer necessity of narrow column width, and because of the inter-conversion of various forms of white space which occurs during character code conversion and the transfer of text between applications. This is why the canonical form for URIs has all white spaces encoded.

Reommendations

This section describes the syntax for URIs as used in the WorldWide Web initiative. The generic syntax provides a framework for new schemes for names to be resolved using as yet undefined protocols.

URI syntax

A complete URI consists of a naming scheme specifier followed by a string whose format is a function of the naming scheme. For locators of information on the Internet, a common syntax is used for the IP

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.