WEBDAV Working Group
Internet-Draft
Expires: May 17, 2006

J. Reschke
greenbytes

November 13, 2005

Web Distributed Authoring and Versioning (WebDAV) URL constraints
draft-reschke-webdav-url-constraints-00

Status of this Memo

Copyright Notice

Abstract

Both WebDAV servers and clients frequently map URI-escaped characters inside a path segment to non-ASCII characters. These mappings can only be interoperable if there is a consensus about the appropriate character encoding. This document specifies a default encoding that is compatible with both the recommendations for URIs in HTML content and the "Internationalized Resource Identifiers" (IRI) specification.

Furthermore, servers that implement a mapping to locally constrained names frequently do not support specific names, or silently map

"similar" names to the same resource (for instance when content is stored in a filesystem that is case-preserving, but not case-sensitive). For these cases, discovery and error signalling features are defined.

Editorial Note

Distribution of this document is unlimited. Please send comments to the Distributed Authoring and Versioning (WebDAV) working group at w3c-dist-auth@w3.org [1], which may be joined by sending a message with subject "subscribe" to w3c-dist-auth-request@w3.org [2].

Discussions of the WEBDAV working group are archived at URL: <http://lists.w3.org/Archives/Public/w3c-dist-auth/>.

Table of Contents

1. Introduction

Both WebDAV servers and clients frequently map URI-escaped characters (see [RFC3986]) inside a path segment to non-ASCII characters. These mappings can only be interoperable if there is a consensus about the appropriate character encoding. This document specifies a default encoding that is compatible with both the recommendations for URIs in HTML content (see [HTML], Appendix B.2.1) and the IRI specification [RFC3987].

Furthermore, servers that implement a mapping to locally constrained names frequently do not support specific names, or silently map "similar" names to the same resource (for instance when content is stored in a filesystem that is case-preserving, but not case-sensitive). For these cases, discovery and error signalling features are defined.

2. Notational Conventions

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 [RFC2119].

3. Terminology

The terminology used here follows that in WebDAV [RFC2518], HTTP [RFC2616] and "Versioning Extensions to WebDAV" [RFC3253]. Definitions of the terms resource, Uniform Resource Identifier (URI), and Uniform Resource Locator (URL) are provided in [RFC3986].

This document uses the terms "precondition" and "postcondition" as defined in [RFC3253]. Servers SHOULD report pre-/postcondition failures as described in section 1.6 of this document.

4. Name to URL segment mapping

In proposing a common mapping, the following requirements were taken into account:

R1 For URL characters inside the US-ASCII range (0..127), the mapping should be the identity mapping.

R2 The mapping should provide support for all characters defined in the Unicode character set.

The only widely-deployed character encoding fulfilling these requirements is the UTF-8 character decoding, defined in [RFC3629]. Consequently, it's also the encoding recommended for URLs in HTML

content ([HTML], Appendix B.2.1) and for IRIs ([RFC3987]).

Therefore, clients and servers SHOULD use the UTF-8 character encoding to map non-ASCII characters to/from character sequences in URL segments.

5. Server Considerations

When mapping HTTP URL segments (see [RFC3986], section 3.3) to local storage, the server's behaviour usually depends on the API used to access that storage. In practice, two styles are widely deployed: binary and character-based. The sections below discuss the implications of each and also describe an "identity" mapping.

5.1 Overview of common mapping methods

5.1.1 Mapping URL segments to byte sequences

A typical scenario for this case is when the server does a direct mapping between URLs and objects in a filesystem, and the filesystem uses filenames based on byte sequences. This is the case for typical Unix filesystem implementations.

In this case, mapping between URL segments and local names is straightforward:

- To map from URL segments, just apply URL unescaping to obtain a byte sequence (see [RFC3986], section 2.1)

- To map to URL segments, just apply URL escaping to obtain a sequence of characters suitable for use in a URL segment

The advantage of this simple mapping is that it faithfully stores whatever the original URL contained. On the other hand, this is a binary encoding, and programs that display filenames usually have to map the byte sequence to a character sequence for display. Unless both character encodings match, the results will be either inaccurate (incorrect characters) or the display function will break completely (for instance when an attempt is made to UTF-8-decode a byte stream that was originally encoded using an incompatible encoding such as ISO-8859-1).

Things get even more complicated when there is no single character encoding being used on the server. For instance, in a Unix system multiple users may use different character encodings for filenames. However, the filesystem does not preserve information about what character encoding the filename was encoded with; thus, depending on their "locale" settings, different users will see different names for

the same filesystem object.

5.1.2 Mapping URL segments to character sequences

This scenario is similar to the one discussed in the previous section (5.1.1). For instance it occurs when objects are stored locally in a way that allows Unicode characters in names, such as filenames in the Windows filesystem.

However, in addition to the mapping to byte sequences, an additional mapping to a character sequence is required. As discussed in Section 4, this mapping should use the UTF-8 character encoding ([RFC3629]). Thus, here the mapping can be described as:

- To map from URL segments, apply URL unescaping to obtain a byte sequence (see [RFC3986], section 2.1), then UTF-8-decode to a sequence of characters.

- To map to URL segments, UTF-8-encode the character sequence to a sequence of bytes, then apply URL escaping to obtain a sequence of characters suitable for use in a URL segment

5.1.3 Identity mapping

Finally, it's also possible to simply store the URL segments character by character, in which case no special mapping considerations apply. Note that this approach may be inefficient in case the names contain many URL-escaped sequences (such as when asian characters have been encoded using UTF-8).

5.2 Caveats

The non-trivial mappings have the common drawback that certain sets of legal HTTP URLs can not be mapped to local names (and therefore usually need to be rejected). For the byte sequence mapping described in Section 5.1.1, this will usually be just the null character.

However, when using the character mapping described in Section 5.1.2, whole Unicode character ranges may either be impossible to represent (such as when the underlying filesystem does only support a Unicode subset), or explicitly disallowed (such as non-normalized character sequences, see [CNORM], section 3.2).

In cases like these, servers SHOULD reject operations that attempt to create those non-mappable URLs. Appropriate precondition names are defined in Section 7.1.

6. Client Considerations

In general, the mappings discussed in Section 5.1.2 apply to clients as well. Whether a client maps segments to byte or character sequences usually depends on the platform it runs on, and what system layer it uses. For instance, a filesystem driver for a Unix system usually will have to translate to byte sequences (because that's how many Unix system internally represent filenames).

However, if the client needs to do any mapping it all, there may be sitations where parts of a URL segment can't be mapped to what the client needs internally. In cases like these, it is recommended that the client signals the problem, and provides a way to repair the problem (such as renaming the resource).

7. Additional Method Semantics

7.1 Additional Preconditions

7.1.1 DAV:name-allowed precondition

The name specified by the HTTP request as path segment is available for use as a new binding name (see [draft-ietf-webdav-bind], section 4 and 6).

8. Compatibility Considerations

Servers that use a non-identity mapping may not be able to create new resources with the URLs specified by the client (such as in an MKCOL or a PUT request).

Clients that use a non-identity mapping may not be able to handle all URLs returned by a server (such as a result of a PROPFIND request).

9. Security Considerations

All of the security considerations of HTTP/1.1 and the WebDAV Distributed Authoring Protocol specification also apply to this protocol specification.

TBD: add notes about the inherent security risks when a backend storage maps multiple notations to the same physical object (file), think uppercase/lowercase, trailing blanks/dots, resolution of relative paths ("./", "../").

10. Internationalization Considerations

All internationalization considerations mentioned in [RFC2518] also

apply to this document.

11. IANA Considerations

There are no IANA Considerations.

12. References

12.1 Normative References

[HTML] Raggett, D., Hors, A., and I. Jacobs, "HTML 4.01 Specification", W3C REC REC-html401-19991224, December 1999, <http://www.w3.org/TR/1999/REC-html401-19991224>.

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

[RFC2518] Goland, Y., Whitehead, E., Faizi, A., Carter, S., and D. Jensen, "HTTP Extensions for Distributed Authoring -- WEBDAV", RFC 2518, February 1999.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC3253] Clemm, G., Amsden, J., Ellison, T., Kaler, C., and J. Whitehead, "Versioning Extensions to WebDAV", RFC 3253, March 2002.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 3629, STD 63, November 2003.

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

12.2 Informative References

[CNORM] Duerst, M., Yergau, F., Ishida, R., Wolf, M., Texin, T., and A. Phillips, "Character Model for the World Wide Web 1.0: Normalization", W3C WD-charmod-norm-20040225, February 2004, <http://www.w3.org/TR/2004/WD-charmod-norm-20040225>.

[RFC3987] Duerst, M. and M. Suignard, "Internationalized Resource Identifiers (IRIs)", RFC 3987, January 2005.

[draft-ietf-webdav-bind]
Clemm, G., Crawford, J., Reschke, J., and J. Whitehead, "Binding Extensions to Web Distributed Authoring and Versioning (WebDAV)", draft-ietf-webdav-bind-12 (work in progress), July 2005, <http://greenbytes.de/tech/webdav/draft-ietf-webdav-bind-12.html>.

URIs

[1] <mailto:w3c-dist-auth@w3.org>

[2] <mailto:w3c-dist-auth-request@w3.org?subject=subscribe>

Author's Address

Julian F. Reschke
greenbytes GmbH
Hafenweg 16
Muenster, NW 48155
Germany

Phone: +49 251 2807760
Fax: +49 251 2807761
Email: julian.reschke@greenbytes.de
URI: http://greenbytes.de/tech/webdav/

Index

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.