

The "data" URL scheme

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 (1998). All Rights Reserved.

[1.](#) Abstract

A new URL scheme, "data", is defined. It allows inclusion of small data items as "immediate" data, as if it had been included externally.

[2.](#) Description

Some applications that use URLs also have a need to embed (small) media type data directly inline. This document defines a new URL scheme that would work like 'immediate addressing'. The URLs are of the form:

data:[<mediatype>][;base64],<data>

The <mediatype> is an Internet media type specification (with optional parameters.) The appearance of ";base64" means that the data is encoded as base64. Without ";base64", the data (as a sequence of octets) is represented using ASCII encoding for octets inside the range of safe URL characters and using the standard %xx hex encoding of URLs for octets outside that range. If <mediatype> is omitted, it defaults to text/plain;charset=US-ASCII. As a shorthand, "text/plain" can be omitted but the charset parameter supplied.

The "data:" URL scheme is only useful for short values. Note that some applications that use URLs may impose a length limit; for example, URLs embedded within <A> anchors in HTML have a length limit determined by the SGML declaration for HTML [[RFC1866](#)]. The LITLEN (1024) limits the number of characters which can appear in a single

[RFC 2397](#)

The "data" URL scheme

August 1998

attribute value literal, the ATTSPLEN (2100) limits the sum of all lengths of all attribute value specifications which appear in a tag, and the TAGLEN (2100) limits the overall length of a tag.

The "data" URL scheme has no relative URL forms.

3. Syntax

```
dataurl      := "data:" [ mediatype ] [ ";base64" ] "," data
mediatype    := [ type "/" subtype ] *( ";" parameter )
data         := *urlchar
parameter    := attribute "=" value
```

where "urlchar" is imported from [\[RFC2396\]](#), and "type", "subtype", "attribute" and "value" are the corresponding tokens from [\[RFC2045\]](#), represented using URL escaped encoding of [\[RFC2396\]](#) as necessary.

Attribute values in [\[RFC2045\]](#) are allowed to be either represented as tokens or as quoted strings. However, within a "data" URL, the "quoted-string" representation would be awkward, since the quote mark is itself not a valid urlchar. For this reason, parameter values should use the URL Escaped encoding instead of quoted string if the parameter values contain any "tspecial".

The ";base64" extension is distinguishable from a content-type parameter by the fact that it doesn't have a following "=" sign.

4. Examples

A data URL might be used for arbitrary types of data. The URL

```
data: ,A%20brief%20note
```

encodes the text/plain string "A brief note", which might be useful in a footnote link.

The HTML fragment:

```
<IMG
SRC="
AAAC8IyPqcvT3wCcDkiLc7C0qwyGHhSWpjQu5yqmCYsapyuvUULvONm0ZtfzgFz
ByTB10Qgx0R0TqBQejhRNz0fkVJ+5YiUqrXF5Y5lKh/DeuNcP5yLWGsEbtLi0Sp
a/TPg7JpJHxyendzWTBfX0cx0nKPjgBzi4diinWGdkF8kjdfnycQZXZeYGejmJl
```

ZeGl9i2icVqaNVailT6F5iJ90m6mvuTS40K05M0vDk0Q4XUtwvK0zrcd3iq9uis
F81M10IcR7lEewwcLp7tuNNkM3uNna3F2JQFo97Vriy/Xl4/f1cf5VWzXyym7PH
hhx4dbgYKAAA7"
ALT="Larry">

could be used for a small inline image in a HTML document. (The embedded image is probably near the limit of utility. For anything else larger, data URLs are likely to be inappropriate.)

A data URL scheme's media type specification can include other parameters; for example, one might specify a charset parameter.

`data:text/plain;charset=iso-8859-7,%be%fg%be`

can be used for a short sequence of greek characters.

Some applications may use the "data" URL scheme in order to provide setup parameters for other kinds of networking applications. For example, one might create a media type

`application/vnd-xxx-query`

whose content consists of a query string and a database identifier for the "xxx" vendor's databases. A URL of the form:

`data:application/vnd-xxx-
query,select_vcount,fc0l_from_fielddtable/local`

could then be used in a local application to launch the "helper" for application/vnd-xxx-query and give it the immediate data included.

[5.](#) History

This idea was originally proposed August 1995. Some versions of the data URL scheme have been used in the definition of VRML, and a version has appeared as part of a proposal for embedded data in HTML. Various changes have been made, based on requests, to elide the media type, pack the indication of the base64 encoding more tightly, and eliminate "quoted printable" as an encoding since it would not easily yield valid URLs without additional %xx encoding, which itself is sufficient. The "data" URL scheme is in use in VRML, new applications of HTML, and various commercial products. It is being used for object

parameters in Java and ActiveX applications.

[6.](#) Security

Interpretation of the data within a "data" URL has the same security considerations as any implementation of the given media type. An application should not interpret the contents of a data URL which is marked with a media type that has been disallowed for processing by the application's configuration.

Sites which use firewall proxies to disallow the retrieval of certain media types (such as application script languages or types with known security problems) will find it difficult to screen against the inclusion of such types using the "data" URL scheme. However, they should be aware of the threat and take whatever precautions are considered necessary within their domain.

The effect of using long "data" URLs in applications is currently unknown; some software packages may exhibit unreasonable behavior when confronted with data that exceeds its allocated buffer size.

[7.](#) References

- [RFC2396] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", [RFC 2396](#), August 1998.
- [RFC1866] Berners-Lee, T., and D. Connolly, "Hypertext Markup Language - 2.0.", [RFC 1866](#), November 1995.
- [RFC2045] Freed N., and N. Borenstein., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", [RFC 2045](#), November 1996.

Author contact information:

Larry Masinter
Xerox Palo Alto Research Center
3333 Coyote Hill Road

Palo Alto, CA 94304

EMail: masinter@parc.xerox.com

Masinter

Standards Track

[Page 4]

[RFC 2397](#)

The "data" URL scheme

August 1998

Full Copyright Statement

Copyright (C) The Internet Society (1998). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an

"AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.