Network Working Group
Internet-Draft
Updates: 4180 (if approved)
Intended status: Informational
Expires: June 5, 2014

M. Hausenblas
DERI, NUI Galway
E. Wilde
EMC Corporation
J. Tennison
Open Data Institute

December 2, 2013

URI Fragment Identifiers for the text/csv Media Type
draft-hausenblas-csv-fragment-08

Abstract

This memo defines URI fragment identifiers for text/csv MIME entities. These fragment identifiers make it possible to refer to parts of a text/csv MIME entity, identified by row, column, or cell. Fragment identification can use single items, or ranges.

Note to Readers

This draft should be discussed on the apps-discuss mailing list [1].

Online access to all versions and files is available on github [2].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on June 5, 2014.

Copyright Notice

Table of Contents

1. Introduction

This memo updates the text/csv media type defined in RFC 4180 [RFC4180] by defining URI fragment identifiers for text/csv MIME entities.

The change to the text/csv media type registration required IESG approval, as the IESG is the change controller for that registration. The IESG has, after consultation with the IETF community, approved the change, which is specified in Section 5 of this document.

This section gives an introduction to the general concepts of text/csv MIME entities and URI fragment identifiers, and discusses the need for fragment identifiers for text/csv and deployment issues. Section 2 discusses the principles and methods on which this memo is based. Section 3 defines the syntax, and Section 4 discusses processing of text/csv fragment identifiers.

1.1. What is text/csv?

Internet Media Types (often referred to as "MIME types") as defined in RFC 2045 [RFC2045] and RFC 2046 [RFC2046] are used to identify different types and sub-types of media. The text/csv media type is defined in RFC 4180 [RFC4180], using US-ASCII [ASCII] as the default character encoding (other character encodings can be used as well). Apart from a media type parameter for specifying the character encoding ("charset"), there is a second media type parameter ("header") that indicates whether there is a header row in the CSV document or not.

1.2. Why text/csv Fragment Identifiers?

URIs are the identification mechanism for resources on the Web. The URI syntax specified in RFC 3986 [RFC3986] optionally includes a so-called "fragment identifier", separated by a number sign ("#"). The fragment identifier consists of additional reference information to be interpreted by the client after the retrieval action has been successfully completed. The semantics of a fragment identifier is a property of the media type resulting from a retrieval action, regardless of the URI scheme used in the URI reference. Therefore, the format and interpretation of fragment identifiers is dependent on the media type of the retrieval result.

1.2.1. Motivation

Similar to the motivation in RFC 5147 [RFC5147], which defines fragment identifiers for plain text files, referring to specific parts of a resource can be very useful, because it enables users and

applications to create more specific references. Users can create references to the part they really are interested in or want to talk about, rather than always pointing to a complete resource. Even though it is suggested that fragment identification methods are specified in a media type's registration (see [RFC6838]), many media types do not have fragment identification methods associated with them.

Fragment identifiers are only useful if supported by the client, because they are only interpreted by the client. Therefore, a new fragment identification method will require some time to be adopted by clients, and older clients will not support it. However, because the URI still works even if the fragment identifier is not supported (the resource is retrieved, but the fragment identifier is not interpreted), rapid adoption is not highly critical to ensure the success of a new fragment identification method.

1.2.2. Use Cases

Fragment identifiers for text/csv as defined in this memo make it possible to refer to specific parts of a text/csv MIME entity. Use cases include, but are not limited to, selecting a part for visual rendering, stream processing, making assertions about a certain value (provenance, confidence, comments, etc.), or data integration.

1.3. Incremental Deployment

As long as text/csv fragment identifiers are not supported universally, it is important to consider the implications of incremental deployment. Clients (for example, Web browsers) not supporting the text/csv fragment identifier described in this memo will work with URI references to text/csv MIME entities, but they will fail to understand the identification of the sub-resource specified by the fragment identifier, and thus will behave as if the complete resource was referenced. This is a reasonable fallback behavior, and in general users should take into account the possibility that a program interpreting a given URI will fail to interpret the fragment identifier part. Since fragment identifier evaluation is local to the client (and happens after retrieving the MIME entity), there is no reliable way for a server to determine whether a requesting client is using a URI containing a fragment identifier.

1.4. Notation Used in this Memo

The capitalized 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 RFC

2119 [RFC2119].

2. Fragment Identification Methods

This memo specifies fragment identification using following methods: "row" for row selections, "col" for columns selections, and "cell" for cell selections.

Throughout the sections below, the following example table in CSV (having 7 rows, including one header row, and 3 columns) is used:

```
date,temperature,place
2011-01-01,1,Galway
2011-01-02,-1,Galway
2011-01-03,0,Galway
2011-01-01,6,Berkeley
2011-01-02,8,Berkeley
2011-01-03,5,Berkeley
```

2.1. Row-based selection

To select a specific record, the "row" scheme followed by a single number is used (the first row is at position 1).

```
http://example.com/data.csv#row=4
```

The above CSV fragment identifies the fourth row:

```
2011-01-03,0,Galway
```

Fragments can also select ranges of rows:

```
http://example.com/data.csv#row=5-7
```

The above CSV fragment identifies three consecutive rows:

```
2011-01-01,6,Berkeley
2011-01-02,8,Berkeley
2011-01-03,5,Berkeley
```

The value "*" can be used to indicate the last row, so the previous URI is equivalent to:

```
http://example.com/data.csv#row=5-*
```

2.2. Column-based selection

To select values from a certain column, the "col" scheme is used, followed by a position (the first column is at position 1):

```
http://example.com/data.csv#col=2
```

The above CSV fragment addresses the second column, identifying the column:

```
temperature
1
-1
0
6
8
5
```

The "col" scheme can also be used to identify ranges of columns:
http://example.com/data.csv#col=1-2

The above CSV fragment addresses the first and second column:

```
date,temperature
2011-01-01,1
2011-01-02,-1
2011-01-03,0
2011-01-01,6
2011-01-02,8
2011-01-03,5
```

As for rows, the value "*" can be used to indicate the last column.

2.3. Cell-based selection

To select particular fields, the "cell" scheme is used, followed by a row number, a comma, and a column number.
http://example.com/data.csv#cell=4,1

The above CSV fragment addresses the field in the first column within the fourth row, yielding:
2011-01-03

It is also possible to select cell-based fragments that have more than just one cell, in which case the cell selection uses the same range syntax as for row and column range selections. For these selections, the syntax uses the upper-lefthand cell as the starting point of the selection, followed by a minus sign, and then the lower-righthand cell as the end point of the selection.
http://example.com/data.csv#cell=4,1-6,2

The above CSV fragment selects a region that starts at the fourth row and the first column, and ends at the sixth row and the second column:

```
2011-01-03,0
2011-01-01,6
2011-01-02,8
```

2.4. Multi-Selections

Row, column, and cell selections can make more than one selection, in which case the individual selections are separated by semicolons. In these cases, the resulting fragment may be a disjoint fragment, such as the selection "#row=3;6" for the example CSV, which would select the third and the sixth row. It is up to the user agent to decide how to handle disjoint fragments, but since they are allowed, user agents should be prepared to handle disjoint fragments.

3. Fragment Identification Syntax

The syntax for the text/csv fragment identifiers is as follows.

The following syntax definition uses ABNF as defined in RFC 4234 [RFC4234], including the rule DIGIT.

NOTE: In the descriptions that follow, specified text values MUST be used exactly as given, using exactly the indicated lower-case letters. In this respect, the ABNF usage differs from [RFC4234].

```
csv-fragment =  rowsel / colsel / cellsel
rowsel       =  "row=" singlespec 0*( ";" singlespec)
colsel       =  "col=" singlespec 0*( ";" singlespec)
cellsel      =  "cell=" cellspec 0*( ";" cellspec)
singlespec   =  position [ "-" position ]
cellspec     =  cellrow "," cellcol [ "-" cellrow "," cellcol ]
cellrow      =  position
cellcol      =  position
position     =  number / "*"
number       =  1*( DIGIT )
```

4. Fragment Identifier Processing

Applications implementing support for the mechanism described in this memo MUST behave as described in the following sections.

4.1. Syntax Errors in Fragment Identifiers

If a fragment identifier contains a syntax error (i.e., does not conform to the syntax specified in Section 3), then it MUST be ignored by clients. Clients MUST NOT make any attempt to correct or guess fragment identifiers. Syntax errors MAY be reported by clients.

4.2. Semantics of Fragment Identifiers

Rows and columns in CSV are counted from one. Positions thus refer to the rows and columns starting from position 1, which identifies the first row or column of a CSV. The special character "*" can be used to refer to the last row or column of a CSV, thus allowing fragment identifiers to easily identify ranges that extend to the last row or column.

If single selections refer to non-existing rows or columns (i.e., beyond the size of of the CSV), they MUST be ignored.

If ranges extend beyond the size of the CSV (by extending to rows or columns beyond the size of the CSV), they MUST be interpreted to only extend to the actual size of the CSV.

If selections of ranges of rows or columns or selections of cell ranges are specified in a way so that they select "inversely" (i.e., "#row=10-5" or "#cell=10,10-5,5"), they MUST be ignored.

Each specification of an identified region is processed independently, and ignored specifications (because of reason listed in the previous paragraphs) do not cause the whole fragment identifier to fail, they just mean that this single specification is ignored. For the example file, the fragment identifier "#row=1-2;5-4;13-16" does identify the first two rows: the second specification is an "inverse" specification and thus is ignored, and the third specification targets rows beyond the actual size of the CSV and thus is also ignored.

The complete fragment identifier identifies all the successfully evaluated identified parts as a single identified fragment. This fragment can be disjoint because of multiple selections. Multiple selections also can result in overlapping individual parts, and it is up to the user agent how to process such a fragment, and whether the individual parts are still made accessible (i.e., visualized in visual user agents), or are presented as one unit. For example, the fragment identifier "#row=3-6;4-5" contains a second identified part that is completely contained in the first identified part. Whether a user agent maintains this selection as two parts, or simply signals that the identified fragment spans from the third to the sixth row, is up for the user agent to decide.

5. IANA Considerations

Note to RFC Editor: Please change this section to read as follows after the IANA action has been completed: "IANA has added a reference

to this specification in the text/csv Media Type registration."

IANA is requested to update the registration of the MIME Media type text/csv at http://www.iana.org/assignments/media-types/text/ with the fragment identifier defined in this memo by adding a reference to this memo (with the appropriate RFC number once it is known).

5.1. The text/csv media type

The Internet media type [RFC6838] for a CSV document is text/csv. The following registration has been copied from the original registration of text/csv [RFC4180], with the exception of the added fragment identification considerations, and added security considerations for fragment identifiers.

Type name: text

Subtype name: csv

Required parameters: none

Optional parameters: charset, header

> The "charset" parameter specifies the charset employed by the CSV content. In accordance with RFC 6657 [RFC6657], the charset parameter SHOULD be used, and if it is not present, UTF-8 SHOULD be assumed as the default (this implies that US-ASCII CSV will work, even when not specifying the "charset" parameter). Any charset defined by IANA for the "text" tree may be used in conjunction with the "charset" parameter.
>
> The "header" parameter indicates the presence or absence of the header line. Valid values are "present" or "absent". Implementors choosing not to use this parameter must make their own decisions as to whether the header line is present or absent.

Encoding considerations: CSV MIME entities consist of binary data [RFC6838]. As per section 4.1.1. of RFC 2046 [RFC2046], this media type uses CRLF to denote line breaks. However, implementers should be aware that some implementations may use other values.

Security considerations:

> Text/csv consists of nothing but passive text data that should not pose any direct risks. However, it is possible that malicious data may be included in order to exploit buffer overruns or other bugs in the program processing the text/csv

data.

The text/csv format provides no confidentiality or integrity protection, so if such protections are needed they must be supplied externally.

The fact that software implementing fragment identifiers for CSV and software not implementing them differs in behavior, and the fact that different software may show documents or fragments to users in different ways, can lead to misunderstandings on the part of users. Such misunderstandings might be exploited in a way similar to spoofing or phishing.

Implementers and users of fragment identifiers for CSV text should also be aware of the security considerations in RFC 3986 [RFC3986] and RFC 3987 [RFC3987].

Interoperability considerations: Due to lack of a single specification, there are considerable differences among implementations. Implementers should "be conservative in what you do, be liberal in what you accept from others" (RFC 793 [RFC0793]) when processing CSV files. An attempt at a common definition can be found in Section 2. Implementations deciding not to use the optional "header" parameter must make their own decision as to whether the header is absent or present.

Published specification: While numerous private specifications exist for various programs and systems, there is no single "master" specification for this format. An attempt at a common definition can be found in Section 2 of RFC 4180 [RFC4180].

Applications that use this media type: Spreadsheet programs and various data conversion utilities.

Fragment identifier considerations: Fragment identification for text/csv is supported by using fragment identifiers as specified by RFC XXXX (Note to RFC Editor: Please update with RFC number once it is known).

Additional information:

Magic number(s): none

File extension(s): CSV

Macintosh file type code(s): TEXT

Person & email address to contact for further information: Yakov Shafranovich <ietf@shaftek.org> and Erik Wilde <dret@berkeley.edu>

Intended usage: COMMON

Restrictions on usage: none

Author: Yakov Shafranovich <ietf@shaftek.org> and Erik Wilde <dret@berkeley.edu>

Change controller: IESG

6. Security Considerations

The security considerations for text/csv fragment identifiers are listed in the respective section of the media type registration Section 5.1.

7. Change Log

Note to RFC Editor: Please remove this section before publication.

7.1. From -07 to -08

- Added IESG approval note.
- Removed "Implementation Status" section.

7.2. From -06 to -07

- Changing "charset" parameter to "SHOULD be used" and UTF-8 as default value.
- Changing encoding to be binary.

7.3. From -05 to -06

- Adding complete media type registration by copying and editing the registration from RFC 4180.
- Moving "Security Considerations" text to media type registration.

7.4. From -04 to -05

- Updating "Implementation Status" section to refer to RFC 6982 [RFC6982].
- Switching to <?rfc symrefs="yes" ?>

7.5. From -03 to -04

- Switched category from "std" to "info".
- Changed the definition of positions to start counting from 1 instead of 0.

7.6. From -02 to -03

- Added "Implementation Status" section.
- Added examples of ranges of rows and columns.
- Corrected errors in examples.

7.7. From -01 to -02

- Removed slices ("#where:") as fragment identification method.
- Removed any special support for headers, which means that they are now treated as a regular (the first) row (if a header row is present).
- Changed semantics and syntax to allow multiple selection of rows, columns, and cells, and to allow ranges of rows and columns.

7.8. From -00 to -01

- Added cell-based selections.
- Added Jeni Tennison as author; updated Erik Wilde's affiliation to EMC.

8. References

8.1. Normative References

[RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

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

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

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

[RFC4180] Shafranovich, Y., "Common Format and MIME Type for Comma-Separated Values (CSV) Files", RFC 4180, October 2005.

[RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234, October 2005.

[RFC6657] Melnikov, A. and J. Reschke, "Update to MIME regarding "charset" Parameter Handling in Textual Media Types", RFC 6657, July 2012.

8.2. Non-Normative References

[ASCII] ANSI X3.4-1986, "Coded Character Set - 7-Bit American National Standard Code for Information Interchange", STD 63, RFC 3629, 1992.

[RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[RFC5147] Wilde, E. and M. Duerst, "URI Fragment Identifiers for the text/plain Media Type", RFC 5147, April 2008.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

[RFC6982] Sheffer, Y. and A. Farrel, "Improving Awareness of Running Code: The Implementation Status Section", RFC 6982, July 2013.

URIs

[1] <https://www.ietf.org/mailman/listinfo/apps-discuss>

[2] <https://github.com/dret/I-D/tree/master/csv-fragment>

Appendix A. Acknowledgements

Thanks for comments and suggestions provided by Nevil Brownlee, Richard Cyganiak, Ian Davis, Gannon Dick, Leigh Dodds, and Barry Leiba.

Authors' Addresses

Michael Hausenblas
DERI, NUI Galway
IDA Business Park
Galway
Ireland

Phone: +353-91-495730
Email: michael.hausenblas@deri.org
URI: http://sw-app.org/about.html

Erik Wilde
EMC Corporation
6801 Koll Center Parkway
Pleasanton, CA 94566
U.S.A.

Phone: +1-925-6006244
Email: erik.wilde@emc.com
URI: http://dret.net/netdret/

Jeni Tennison
Open Data Institute
65 Clifton Street
London EC2A 4JE
U.K.

Phone: +44-797-4420482
Email: jeni@jenitennison.com
URI: http://www.jenitennison.com/blog/