                 Parameter Value Encoding in iCalendar and vCard
                    draft-daboo-ical-vcard-parameter-encoding-03

Abstract

   This specification updates the data formats for iCalendar (RFC 5545)
   and vCard (RFC 6350) to allow parameter values to include certain
   characters forbidden by the existing specifications.

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 July 12, 2013.

Copyright Notice

Table of Contents

1. Introduction

   The iCalendar [RFC5545] specification defines a standard way to
   describe calendar data. The vCard [RFC6350] specification defines a
   standard way to describe contact data. Both of these use a similar
   text-based data format. Each iCalendar and vCard data object can
   include "properties" which have "parameters" and a "value". The
   value of a "parameter" is typically a token or URI value, but a
   "generic" text value is also allowed. However, the syntax rules for
   both iCalendar and vCard prevent the use of a double-quote character
   or control characters in such values, though double-quote and some
   subset of control characters are allowed in the actual property
   values.

   As more and more extensions are being developed for these data
   formats, there is a need to allow at least double-quotes and line
   feeds to be included in parameter values. The \-escaping mechanism
   used for property text values is not defined for use with parameter
   values and cannot be easily used in a backwards compatible manner.
   This specification defines a new character escaping mechanism,
   compatible with existing parsers and chosen to minimise any impact on
   existing data.

2. Conventions Used in This Document

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in
   [RFC2119].

3. Parameter Value Encoding Scheme

   This specification defines the ^ character (U+005E - Circumflex
   Accent) as an escape character in parameter values whose value type
   is defined using the "param-value" syntax element (Section 3.1 of
   iCalendar [RFC5545] and Section 3.3 of vCard [RFC6350]). The
   ^-escaping mechanism can be used when the value is either unquoted or
   quoted (i.e., whether or not the value is surrounded by double-
   quotes). When used, the combination of ^ plus one other character is
   interpreted as a single character as per the table below:

```
       +---------------------+----------------------------+
       | Escape Sequence     | Character                  |
       +---------------------+----------------------------+
       | ^' (U+005E, U+0027) | U+0022 (Quotation Mark)    |
       | ^n (U+005E, U+006E) | U+000A (Line Feed)         |
       | ^^ (U+005E, U+005E) | U+005E (Circumflex Accent) |
       +---------------------+----------------------------+
```

   When generating iCalendar or vCard parameter values, the characters
   in the second column above MUST be escaped using the sequence from
   the first column. When parsing iCalendar or vCard parameter values,
   the two character sequence from the first column MUST be replaced by
   the single character from the second column. If a ^ (U+005E)
   character is followed by any other character than the ones above,
   parsers MUST leave both the ^ and the following character in place.

   In addition to the escaping procedure outlined above, text parameter
   values MUST also have line break character sequences canonicalized
   into a single line feed (U+000A) character when the parameter value
   is generated. Specifically, any single carriage return character
   (U+000D) or the two character combination of carriage return followed
   by line feed (U+000D, U+000A) MUST be mapped to ^n in the generated
   parameter value. When parsing a parameter value with ^n, clients can
   replace that with the appropriate line break character sequence for
   the system they are running on.

   When converting between iCalendar and vCard text-based data formats
   and alternative data-format representations such as XML (as described
   in [RFC6321] and [RFC6351] respectively), implementations MUST ensure
   that parameter value escape sequences are generated correctly in the
   text-based format and are decoded when the parameter values appear in
   the alternate data formats.

3.1. iCalendar Example

   An "ATTENDEE" property with a "CN" parameter whose value includes two
   double-quote characters. The parameter value is not quoted as there
   are no characters in the value that would trigger quoting as required
   by iCalendar.

   ATTENDEE;CN=George Herman ^'Babe^' Ruth:mailto:babe@example.com

   The unescaped parameter value is

   George Herman "Babe" Ruth

3.2. vCard Examples

A "GEO" property with a "X-ADDRESS" parameter whose value includes
several line feed characters. The parameter value is also quoted
since it contains a comma which triggers quoting as required by
vCard.

GEO;X-ADDRESS="Pittsburgh Pirates^n115 Federal St^nPitt
 sburgh, PA 15212":geo:40.446816,-80.00566

The unescaped parameter value is (where each line is terminated by a
line break character sequence)

Pittsburgh Pirates
115 Federal St
Pittsburgh, PA 15212

4. Security Considerations

There are no additional security issues beyond those of iCalendar
[RFC5545] and vCard [RFC6350].

5. IANA Considerations

There are none.

6. Acknowledgments

Thanks to Michael Angstadt, Tim Bray, Mike Douglass, Barry Leiba, and
Simon Perreault for feedback on this specification.

7. Normative References

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

[RFC5545]   Desruisseaux, B., "Internet Calendaring and Scheduling
            Core Object Specification (iCalendar)", RFC 5545,
            September 2009.

[RFC6321]   Daboo, C., Douglass, M., and S. Lees, "xCal: The XML
            Format for iCalendar", RFC 6321, August 2011.

[RFC6350]   Perreault, S., "vCard Format Specification", RFC 6350,

August 2011.

[RFC6351]   Perreault, S., "xCard: vCard XML Representation",
            RFC 6351, August 2011.

Appendix A. Choice of Quoting Mechanism

Having recognized the need for escaping parameters values, the
question is what mechanism to use? One obvious choice would be to
adopt the \-escaping used for property values. However, that could
not be used as-is, because it escapes a double-quote as the sequence
of \ followed by double-quote. Consider what the example in
Section 3.1 might look like using \-escaping:

ATTENDEE;CN="George Herman \"Babe\" Ruth":mailto:babe@example.com

Existing iCalendar/vCard parsers know nothing about escape sequences
in parameters. So they would parse the parameter value as:

George Herman \

i.e. the text between the first and second occurrence of a double-
quote. However, the text after the second double-quote ought to be
either a : or a ; (to delimit the parameter value from the following
parameter or property) but is not - so the parser could legitimately
throw an error at that point because the data is syntactically
invalid. Thus for backwards compatibility reasons, a double-quote
cannot be escaped using a sequence that itself includes a double-
quote. Hence the choice of using a single-quote in this
specification.

Another option would be to use a form of \-escaping modified for use
in parameter values only. However, some incorrect, non-interoperable
use of \ in parameter values has been observed, and thus it is best
to steer clear of that to achieve guaranteed, reliable
interoperability. Also, given that double-quote gets changed to
single-quote in the escape sequence for a parameter, but not for a
value, it is better to not give the impression that the same escape
mechanism (and thus code) can be used for both (which could lead to
other issues such as an implementations incorrectly escaping a ; as
\; as opposed to quoting the parameter value).

The choice of ^ as the escape character was made based on the
requirement that an ASCII symbol (non-alphanumeric) character be
used, and it ought to be one least likely to be found in existing
data.

Appendix B. Change History (To be removed by RFC Editor before
             publication)

   Changes in -03:

   1. Tweak wording and change SHOULD to MUST for leaving ^+any other
       character sequences unchanged.

   2. Added paragraph on interaction with other data-format
       representations such as XML.

   Changes in -02:

   1. Removed unnecessary HTAB escaping.

   2. Added appendix to describe why ^ was chosen as the escape
       mechanism.

   Changes in -01:

   1. Tweaked vCard example line-folding.

Author's Address

   Cyrus Daboo
   Apple Inc.
   1 Infinite Loop
   Cupertino, CA 95014
   USA

   Email: cyrus@daboo.name
   URI: http://www.apple.com/