[CSV]- No quote char unsupported scenarios

Last updated by | Jackie Huang | Jan 4, 2022 at 12:24 AM PST

Contents

Issue

There are several cases which is not supported in csv 'no quote char' scenario:

1. When quote character is set to 'no quote char', multi-char column delimiter can't start and end with the same letters
2. When quote character is set to 'no quote char', multi-char column delimiter can't contain escapeChar: \
3. When quote character is set to 'no quote char', column value can't contain row delimiter
4. Quote character and escape character cannot both be empty (no quote and no escape) if column value contains column delimiter.

Root Cause

It will has ambiguity. For example:

1. Start and end with the same letters

```
column delimiter: $*^$*
column value: abc$*^    def
csv sink: abc$*^$*^$*def
will be read as "abc" and "^&*def"
```

2. Multi-char delimiter contains escapeChar

```
column delimiter: \x
escape char:\
column value: "abc\\xdef"
```

escape char will either escape column delimiter or escape escape char.

3. Column value contain row delimiter

```
We need quote character to tell if row delimiter is inside column value or not.
```

4. Quote character and escape character both be empty and column value contains column delimiter

```
Column delimiter: \t
column value: 111\t222\t33\t3
It will be ambigious if it contains 3 columns 111,222,33\t3 or 4 columns 111,222,33,3.
```

Resolution

1. By design
2. We will unblock in near future.
3. Never use no quote char for a multiline csv file.
4. Set either quote character or escape character as non-empty. Or remove all column delimiter inside customer's data.

Additional Information:

- Icm Reference: N/A
- Author: Zhuoyang Zhang; Xiaojin Wang
- Reviewer: Shawn Xiao; Zhuoyang Zhang
- Keywords: DelimitedText

How good have you found this content?

🙂 🙁