SixArm / **usv** Public

Code

Issues 6

Pull requests

Actions

Projects

Wiki

Security

Insights

main

joelparkerhenderson Merge pull request #7 from mgred/improve-docs

…

`58f70a0` May 9, 2022

33 commits

bin
Add USVX info about final newline
May 6, 2022

doc
Rework BNF for more specificity
May 8, 2022

README.md
Improve Bash script in README
May 9, 2022

README.md

Unicode Separated Values (USV)

Unicode separated values (USV) is a data format that places Unicode characters between data parts.

- ␟ = U+241F = US = Unit Separator. Use between each data unit, database column, spreadsheet cell, etc.
- ␞ = U+241E = RS = Record Separator. Use between each data record, database row, spreadsheet line, etc.
- ␝ = U+241D = GS = Group Separator. Use between each data group, database table, spreadsheet grid, etc.
- ␜ = U+241C = FS = File separator. Use between each data file, database schema, spreadsheet folio, etc.

The USV file name extension is ".usv".

The USV repo is https://github.com/sixarm/usv.

If you know about comma separated values (CSV), or tab separated values (TSV), or ASCII separated values (ASV), then you already know most of USV.

If you want more capabilties, then try Unicode Separated Values Extensions (USVX), which uses the same Unicode characters, with extensions that provide whitespace trim, backslash escape, and final newline. See https://github.com/sixarm/usvx.

Documentation

Documentation files here:

- Frequently asked questions
- How to type USV Unicode characters
- TODO list
- BNF: Backus-Naur form for standarization
- Comparisons with CSV, TSV, TDF, ASV, DEL
- Objections to USV and USVX
- History of ASCII separated values (ASV)

Examples

USV with 2 units:

```
a␟b
```

USV with 2 units by 2 records:

```
a␟b␞c␟d
```

USV with 2 units by 2 records by 2 groups:

```
a␟b␞c␟d␝e␟f␞g␟h
```

USV with 2 units by 2 records by 2 groups by 2 files:

```
a␟b␞c␟d␝e␟f␞g␟h␜i␟j␞k␟l␝m␟n␞o␟p
```

USV is easy and friendly

USV is intended to be easy to use and friendly to try:

USV works with many kinds of data. Any data can contain any characters except the four USV characters.

USV works with many kinds of editors. Any editor that can render the USV characters will work. We use vi, emacs, Coda, Notepad++, TextMate, Sublime, VS Code, etc.

USV works with many kinds of tools. Any tool that can parse the USV characters will work. We use awk, sed, grep, rg, miller, etc.

USV works with many kinds of languages. Any language that can handle UTF-8 character encoding and rendering should work. We use C, Go, Java, JavaScript, Julia, Python, R, Ruby, Rust, Swift, PHP, etc.

Why use USV?

USV can handle data that contains commas, semicolons, quotes, tabs, newlines, and other special characters, all without escaping.

USV can format units/columns/cells and records/rows/lines (similar to CSV) and groups/tables/grids and files/schemas/folios (similar to ASV).

USV is aiming to be an international standard.

USV uses Unicode characters that are semantically meaningful.

USV works well with any typical modern editor, font, terminal, shell, search, and language.

USV uses visible letter-width characters, and these are easy to view, select, copy, paste, search.

USV is simpler and clearer to use and write because there are no escape characters.

Example USV scripts

Create an example USV file with 2 units by 2 records:

```
$ echo 'a␟b␞c␟d' > example.usv
```

Convert USV to CSV by using `sed` :

```
$ cat example.usv | sed 's/␟/,/g; s/␞/\n/g;'
a,b
c,d
```

Convert USV to TSV by using `tr` :

```
$ cat example.usv | tr ␟␞ ',\n'
a,b
c,d
```

Create an example USV file with 2 units by 2 records by 2 groups by 2 files:

```
echo "a␟b␞c␟d␝e␟f␞g␟h␜i␟j␞k␟l␝m␟n␞o␟p" > example.usv
```

Convert USV to TSV by using `awk` :

```
$ cat example.usv | awk 'BEGIN { FS="␟"; RS="␞"; OFS=","; ORS="\n"; } {$1=$1}1' | grep -v ^$
```

Print USV with typical shell commands and pretty output:

```
$ cat example.usv | sed 's/␟/,/g; s/␞/\n/g; s/␝/\n---\n/g; s/␜/\n===\n/g;'
a,b
c,d
---
e,f
g,h
===
i,j
k,l
---
m,n
o,p
```

Print USV characters by using a shell script with `bash` :

```
#!/usr/bin/env bash
set -euf -o pipefail

# USV example shell script that demonstrates the use of USV characters.
# This script reads STDIN one character at a time, and prints output.
# There is a similar USVX example shell script that demonstrates
# extensions for whitespace trimming and backslash escaping.

while IFS= read -n1 -r c; do
    case  "$c" in
    "␟")
        printf "\nunit separator\n"
        ;;
    "␞")
        printf "\nrecord separator\n"
        ;;
    "␝")
        printf "\ngroup separator\n"
        ;;
    "␜")
        printf "\nfile separator\n"
        ;;
    *)
        printf %s "$c"
        ;;
    esac
done
```

Conclusion

USV is simple and fast, without extras.

USVX adds extensions for whitespace trim, backslash escape, and final newline.

USV and USVX are helping us with data format projects, and we hope USV and USVX may be able to help you too.

We welcome constructive feeback about USV and USVX, as well as git issues, pull requests, and standardization help.

https://github.com/sixarm/usv

https://github.com/sixarm/usvx

About

USV: Unicode Separated Values

Readme

36 stars

3 watching

3 forks

Releases

No releases published

Packages

No packages published

Contributors 3

joelparkerhenderson Joel Parker Henderson

mgred Marc Redemske

junichi-chiba

Languages

● **Shell** 100.0%