

構造化テキストをCSV変換

Structured textというほどではないが単にタグをつけたようなテキストデータは結構あるのではないだろうか。
そういったデータをCSV化するスクリプトである。

入力テキスト

[st.txt](#)

```
Name: Convert-StructuredTextToCsv.ps1
Length: 1749
Mode: -a----
LastWriteTime: 2023/07/29 11:04:51

Name: output.csv
Length: 0
Mode: -a----
LastWriteTime: 2023/07/28 21:11:24

Name: st.txt
Length: 33
Mode: -a----
LastWriteTime: 2023/07/28 21:11:19
```

変換コマンドレット

- オプション
 - -StructuredText 構造化テキストファイル
 - -ExportCSV 出力CSVファイル

```
PS > .\Convert-StructuredTextToCsv.ps1 -StructuredText .\st.txt -ExportCSV
.\output.csv
```

CSVファイル

[output.csv](#)

```
"Name","Length","Mode","LastWriteTime"
"Convert-StructuredTextToCsv.ps1","1749","-a----","2023/07/29 11:04:51"
"output.csv","0","-a----","2023/07/28 21:11:24"
```

PowerShellスクリプト

Convert-StructuredTextToCsv.ps1

```
# 2023/07/26 Structured-TextからCSVへの変換スクリプト
param (
    [string]$StructuredText, [string]$ExportCSV
)
if (($StructuredText.Length -eq 0) -or -not (Test-Path -Path
$StructuredText)) {
    Write-Host "File not found: $StructuredText"
    exit
}
$header = @()
$name2num = @{}
$data = @()
Get-Content -Path $StructuredText | ForEach-Object {
    $line = $_
    # 拡張タグ処理
    if ($line -match "^#:([\w\$]+)\s*(\w+)") {
        $tag = $Matches[1]
        $key = $Matches[2]
        # Key定義
        if (($tag -eq "define") -and -not
($name2num.ContainsKey($key))) {
            $name2num.Add($key, ($header.Length))
            $header += $key
        }
    }
    # コメントアウト処理
    } elseif ($line -match "^#") {
        continue
    }
    # レコード区切り処理
    } elseif (($line -match "^$") -or ($line -match "^\..$")) {
        if ($data.Length -gt 0) {
            (($data -join ',') | ConvertFrom-Csv -Header $header)
            $data.Clear()
        }
    }
}
# Structured-Textの解析
if ($line -match "^(([\w\$]+): *(.*)$") {
    $key = $Matches[1]
    $value = $Matches[2]
    if ($value -match ",") {
        $value = (-join ('"', $value, '"'))
    }
    if ($name2num.ContainsKey($key)) {
```

```
        $data[($name2num.$key)] = $value
    } else {
        $name2num.Add($key, ($header.Length))
        if ($data.Length -le $header.Length) {
            $data += $value
        } else {
            $data[($name2num.$key)] = $value
        }
        $header += $key
    }
}
} | Export-Csv -Path $ExportCSV -Encoding Default -NoTypeInfoation
Get-Item -Path $ExportCSV
```

Perlスクリプト

[str2csv.pl](#)

```
#!/usr/bin/perl
# StructuredText形式からCSV形式に変換
#
$line = 0;
while(<>) {
    chomp;
    chomp;
    s/\s+$//;
    # StructurTextの要素分解
    ($key, $val) = (split(/:\s*/, $_, 2));
    if ($key =~ m/^[\\w\\-]+$/) {
        #カンマを含む行のクォーティング
        if ($val =~ m/,/) {
            $val = sprintf("\"%s\"", $val);
        }
        # 出現位置への格納
        if (defined($key2Num{$key})) {
            $csvList[$key2Num{$key}] = $val;
        }
        # 項目名出現順の記録
        else {
            push(@keyList, $key);
            push(@csvList, $val);
            $key2Num{$key} = scalar(@keyList) - 1;
        }
    }
}
# 空行位置でのCSV出力
elsif ($key eq '') {
    if ($line == 0) {
        printf(" %s\n", join(',', @keyList));
```

```
}  
if (scalar(@csvList)>0) {  
    printf("%s\n", join(',', @csvList));  
    @csvList = ();  
    $line++;  
}  
}  
}
```

From:
<https://kijima.mydns.jp/dokuwiki/> - Kijima's private wiki

Permanent link:
<https://kijima.mydns.jp/dokuwiki/doku.php?id=%E6%A7%8B%E9%80%A0%E5%8C%96%E3%83%86%E3%82%AD%E3%82%B9%E3%83%88%E3%82%92csv%E5%A4%89%E6%8F%9B>

Last update: 2023/07/29 04:43

