

# Sjisutf16 ユーザーズガイド

sjisutf16 コマンドは、シフト JIS のテキストファイルを UTF-16 のファイルに変換します。

## コマンドプロンプト

Windows のコマンドプロンプトは、シフト JIS が多用されます。MS932 は、いわゆるシフト JIS ですが、規格外です。たとえば、丸付きの数字 (①②③⑩⑳)、ローマ数字 (ⅠⅡⅢⅣⅤ)、1 バイトのカタカナは、EUC-JP で表現しにくい。

### 珍しい漢字

MS932 は、名前の漢字が不十分であるため、名前に含まれた珍しい漢字を表現しにくい。UTF-16 のファイルには、珍しい漢字を挿入しやすい。

## インストール

sjisutf16 コマンドは、Windows のコンソールアプリケーションです。32 ビット版の Windows Vista において、Free Pascal Compiler 3.0.4 で構築しました。

## パスを通す手順の例

1. トリシーカー (<http://tori.tobihiro.jp/PDF-zip-7z.html>) というウェブサイトから Sjisutf16.zip をダウンロードしてください。
2. Sjisutf16.zip を展開してください。
3. Cドライブに Sjisutf16 フォルダを移動してください。利用者は、Sjisutf16 フォルダの移動および改名が可能です。
4. Windows ロゴキー (田キー) を押しながら Pause を押してください。
5. メニューからシステムの詳細設定を選択してください。
6. 詳細設定タブをクリックしてください。
7. [環境変数] ボタンを押してください。キーワード: システム、詳細、環境変数
8. システム環境変数として、Path が見えるまでスクロールしてください。
9. Path をクリックして選択してください。
10. [編集] ボタンを押してください。
11. Windows 10 の場合、[テキストの編集] ボタンを押してください。
12. 右向き矢印 (→) キーまたは End キーを押してください。カーソルが末尾に移動します。
13. フォルダのことをディレクトリとも言います。セミコロン ( ; ) は、ディレクトリどうしを区切る記号です。  
`;C:¥Sjisutf16` を入力してください。
14. 各ウィンドウで OK ボタンを押してください。

## フォルダごと削除

アンインストール(プログラムの削除)を行うには、Sjisutf16 フォルダごと削除してください。Path から C:¥Sjisutf16 を削除してください。

コマンド	文字コード	説明
obot	バイナリファイル	ファイルの最初の部分を十六進数で表現する。
sjisutf16	シフト JIS	MS932~UTF16.bin にしたがって、シフト JIS のテキストファイルを UTF-16 のファイルに変換する。
surro	シフト JIS	surro.txt から surro.csv を作成する。surro.csv は、UTF-16 のファイルである。
unarfi_j	UTF-16	CSV を読み込んで、ある欄のみ選択して、細長いファイルを書き出す。surro.csv および UTF16~MS932.bin にしたがって、UTF-16 のファイルをシフト JIS のテキストファイルに変換する。
utf8to16	UTF-8	UTF-8 のテキストファイルを UTF-16 のファイルに変換する。
utf16sjis	UTF-16	surro.csv および UTF16~MS932.bin にしたがって、UTF-16 のファイルをシフト JIS のテキストファイルに変換する。

## 著作権

Sogaya(そがや)は、Sjisutf16 の著作権を保有しています。

## 免責条項

Sjisutf16 コマンドのご利用によって発生するいかなる損害も、Sogaya は、責任を負わないものとします。

## 構築

32ビット版の Windows Vista において、Free Pascal Compiler 3.0.4 で Sjisutf16 を構築しました。ソースコードを添付しました。

## ユーザーズガイド

2020 年 5 月 18 日 Revision 1.013



Sjisutf16

## Sjisutf16 のパラメータ

sjisutf16 コマンドは、シフト JIS のテキストファイルを UTF-16 のファイルに変換します。

### 設定ファイル

sjisutf16 コマンドの設定ファイルは、シフト JIS のファイルです。利用者は、設定ファイルとしてカレントディレクトリに `set_sjisutf16.txt` を用意できます。`set_sjisutf16.txt` の第 1 行に変換表のパス名を記入できます。第 2 行、第 3 行、第 4 行、…に sjisutf16 コマンドのパラメータを記入できます。

#### エラーファイル

カレントディレクトリで設定ファイルを開くことができない場合、sjisutf16 コマンドは、エラーファイルを書き出しますが、変換処理を続行します。エラーファイルの名前は、`err 通し番号-分-秒.txt` の形式で命名されます。

### 読み込みたいファイル

sjisutf16 コマンドのパラメータとして、シフト JIS ファイルを付加してください。sjisutf16 コマンドは、そのシフト JIS ファイルを読み込んで、UTF-16 のファイルとして、`sjisutf16.txt` を作成します。

### 2 個のファイル

利用者は、`sjisutf16.txt` ではなく新しいファイルの作成および命名を行うことができます。sjisutf16 コマンドのパラメータとして、2 個のファイルが付加された場合、sjisutf16 コマンドは、シフト JIS のファイルとして第 1 のファイルを読み込んで、UTF-16 のファイルとして第 2 のファイルを書き出します。

### COMP コマンド

バイナリファイルとして比較するには、FC コマンドのパラメータとして、`/B` を付加してください。Windows 2000, Windows XP, Windows Vista, Windows 7, Windows 10 では、COMP コマンドを試すことができます。

#### 先に新しいファイルの名前を指定したい

上記のとおり、2 個のファイルが付加した場合、第 2 のファイルが新しいファイルになりますが、先に新しいファイルの名前を指定するには、sjisutf16 コマンドのパラメータとして、`-o text.tmp` を付加してください。`-o` および `text.tmp` の間に空白を挿入し

てください。text.tmp は、新しいファイルの名前です。  
sjisutf16 -o text.tmp Gongitsune.txt

```
C:¥Q>sjisutf16 Gongitsune.txt
C:¥Q>sjisutf16 -o text.tmp Gongitsune.txt
C:¥Q>COMP text.tmp sjisutf16.txt
text.tmp と sjisutf16.txt を比較しています...
ファイルに違いはありません
(ほかのファイルを比較しますか (Y/N)? n
C:¥Q>
```

## MORE コマンド

コマンドプロンプトでシフト JIS のテキストファイルを閲覧する方法として、MORE コマンドが利用できます。スペースキーで進行します。Q で終了します。どの Windows も最初から MORE コマンドを用意しています。

## 中止したい合図

コマンドプロンプトに、中止したい合図を入力するには、Ctrl を押しながら、C を押してください。Ctrl + C は、この操作を意味します。^C が画面に表示される場合があります。

## 標準入力

ファイルが付加されていない場合、sjisutf16 コマンドは、標準入力から行を読み込むことができます。

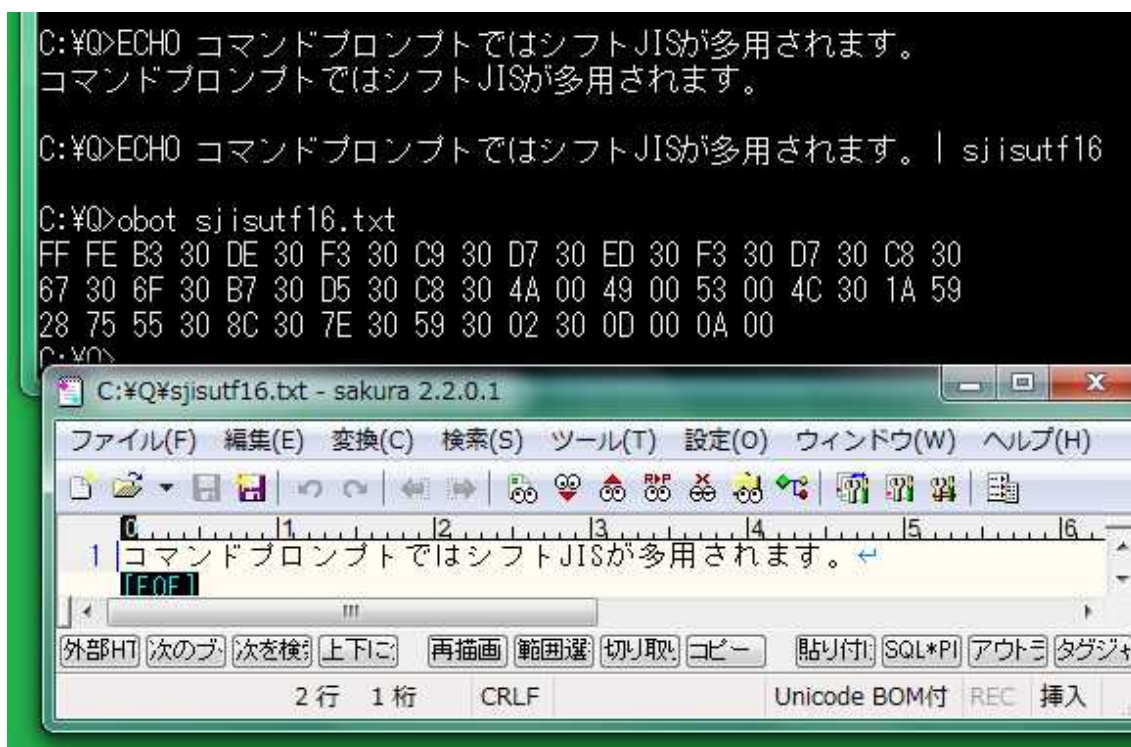
### ECHO コマンド

どの Windows も最初から ECHO コマンドを用意しています。ECHO コマンドのパラメータとして入力した内容は、コマンドプロンプトにシフト JIS のテキストとして表示されます。改行がシフト JIS のテキストに付加されます。

### パイプ

ECHO コマンドおよび sjisutf16 コマンドで構成したパイプにより、sjisutf16.txt を作成した例を下図に示します。添付の obot コマンドは、ファイルの最初の部分を十六進数で表現できます。sjisutf16.txt は、FF FE で開始することを obot コマンドで確認できます。すなわち、sjisutf16.txt は、BOM 付きの UTF-16LE のファイルです。0D 00 0A 00 で改行されていることも確認できます。このユーザーズガイドで

は、UTF-16 は、BOM 付きの UTF-16LE を意味します。



## 改行

0Dh は、十六進数 (hexadecimal) の 0D を意味します。テキストファイルに関して、Windows では、0Dh および 0Ah で改行しますが、UNIX では、0Ah のみで改行します。sjisutf16 コマンドは、各行から 0Dh を削除できます。

### 設定を無効にする

-0 は、-L より優先されるため、-L を指定したい場合、-0 を無効にしなければなりません。初期段階で、sjisutf16 コマンドは、0Dh および 0Ah で改行するように設定されています。この設定を無効にするには、sjisutf16 コマンドのパラメータとして、-0- を付加してください。

パラメータ	改行	説明
-0	0Dh および 0Ah で改行する。	省略された場合、sjisutf16 コマンドは、-0 が付加されたとみなす。
-0-	シフト JIS のファイルにしたがって改行する	シフト JIS ファイルの行が、0D 0A で改行されていれば、sjisutf16 コマンドは、0Dh および 0Ah で改行する。
-0- -L	0Ah のみで改行する。	インターネットでは、これでも問題ない。

### 0A のみで改行

元のシフト JIS ファイルに 0Dh があろうがなかろうが、0Ah のみで改行して UTF-16 の

ファイルを書き出すには、sjisutf16 コマンドのパラメータとして、`-LF` を付加してください。`-LF` は、`-0- -L`(マイナスゼロマイナス空白マイナスエル)と同じ結果を生じません。

## ファイルの日時

元のファイルが新しいファイルと同じ日時を有するように命令するには、sjisutf16 コマンドのパラメータとして、`-t` を付加してください。

### 一時ファイルを削除しない

`-t` 付きの sjisutf16 コマンドは、一時ファイルとして `binary.tmp` を作成します。sjisutf16 コマンドが自動的に一時ファイルを削除しないように命令するには、sjisutf16 コマンドのパラメータとして、`-t -y` を付加してください。

```
C:¥Q>DIR /O:D "Gon*.txt" | FINDSTR "Gon"
2018/10/11 13:53          10,636 Gongitsune.txt

C:¥Q>sjisutf16 -t -y Gongitsune.txt Gon_UTF-16.txt

C:¥Q>COMP binary.tmp Gongitsune.txt
binary.tmp と Gongitsune.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n

C:¥Q>DIR /O:D "Gon*.txt" | FINDSTR "Gon"
2020/03/15 13:18          10,858 Gon_UTF-16.txt
2020/03/15 13:18          10,636 Gongitsune.txt

C:¥Q>_
```

### 日時が新しい

1. 利用者は、カレントディレクトリにシフト JIS のファイルを用意する。
2. sjisutf16 コマンドは、シフト JIS のファイルを読み込む。
3. `-t` 付きの sjisutf16 コマンドは、一時ファイルとして `binary.tmp` を作成する。
4. `binary.tmp` は、内容もサイズもシフト JIS ファイルと同じである。
5. `-t` 付きの sjisutf16 コマンドは、`binary.tmp` を読み込んで新しいシフト JIS ファイルを作成する。
6. `-y` が付加されていないならば、sjisutf16 コマンドは、`binary.tmp` を削除する。
7. sjisutf16 コマンドは、新しいシフト JIS ファイルを読み込んで、UTF-16 のファイルを書き出す。
8. 新しいシフト JIS ファイルは、日時が UTF-16 のファイルと同じである。

## 元のファイル名で命名する

いったん `binary.tmp` を作成して、`binary.tmp` を元のファイル名に改名するには、`sjisutf16` コマンドのパラメータとして、`-b` を付加してください。UTF-16 のファイルは、名前が元のファイルと同じですが、内容もサイズも元のファイルと異なります。

```
C:¥Q>obot WikiTable.txt
7C 7E 82 A2 82 C2 7C 7E 8E E5 8C EA 7C 7E 93 AE 8E 8C 7C 7E
96 DA 93 49 8C EA 7C 0D 0A 7C 31 38 37 37 94 4E 7C 83 47 83
57 83 5C 83 93 7C 94 AD 96 BE 82 B5 82 BD 7C 92 7E 89 B9 8B
40 7C 0D 0A 7C 96 BE 8E A1 8E 9E 91 E3 7C 8E 99 93 87 8C D5
8E 9F 98 59 7C 8E FB 8F 57 82 B5 82 BD 7C 8C 7C 8F 70 95 69
7C 0D 0A
C:¥Q>sjisutf16 WikiTable.txt WikiTable_UTF-16.txt

C:¥Q>sjisutf16 -b WikiTable.txt WikiTable.txt

C:¥Q>COMP WikiTable.txt WikiTable_UTF-16.txt
WikiTable.txt と WikiTable_UTF-16.txt を比較しています...
ファイルに違いはありません

(ほかのファイルと比較しますか (Y/N)? n

C:¥Q>obot WikiTable.txt
FF FE 7C 00 7E 00 44 30 64 30 7C 00 7E 00 3B 4E 9E 8A 7C 00
7E 00 D5 52 5E 8A 7C 00 7E 00 EE 76 84 76 9E 8A 7C 00 0D 00
0A 00 7C 00 31 00 38 00 37 00 37 00 74 5E 7C 00 A8 30 B8 30
BD 30 F3 30 7C 00 7A 76 0E 66 57 30 5F 30 7C 00 C4 84 F3 97
5F 6A 7C 00 0D 00 0A 00 7C 00 0E 66 BB 6C 42 66 E3 4E 7C 00
50 51 F6 5C 4E 86 21 6B CE 90 7C 00 CE 53 C6 96 57 30 5F 30
7C 00 B8 82 53 88 C1 54 7C 00 0D 00 0A
C:¥Q>
```

### UTF-16 の特徴

`FF FE` がファイルの最初に出現して、`0D 00 0A 00` で改行されている場合、シフト JIS のファイルではないことが判明します。

### シフト JIS の特徴

`0D 00 0A 00` ではなく `0D 0A` で改行されている場合、UTF-16 ではないことが判明します。2 バイト文字、3 バイト文字、4 バイト文字の第 1 バイトは、`7Fh` より大きい。EUC-JP も UTF-8 も、`7Fh` より大きい数は、2 個以上、連続します。シフト JIS の特徴として、2 バイト文字の第 2 バイトが `7Fh` より小さい場合があります。すなわち、シフト JIS は、`7Fh` より大きい数は、連続しないで 1 個のみ出現する場合があります。

## シフト JIS の例

上図の場合、たとえば、37 94 4E から、シフト JIS であると認識できます。**2 バイト文字の第 1 バイトが小さい数どうしの間**に出現しているからです。37h は、ASCII の数字の 7 です。94h は、7Fh より大きい数であり、ASCII の数字の直後に出現したため、2 バイト文字の第 1 バイトです。37h および 4Eh は、7Fh より小さい数です。4Eh は、2 バイト文字の第 1 バイトの直後にあるため、2 バイト文字の第 2 バイトであるとみなしません。(4Eh が 2 バイト文字の第 1 バイトの直後に出現したのではない場合、4Eh は、ASCII のアルファベットの N です。)2 バイト文字の第 1 バイトが小さい数どうしの間

十六進数	シフト JIS	2 バイト文字の第 1 バイトである理由
37 94 4E	7 年	94 は、ASCII の数字の直後に出現した。
83 47 83 57 83 5C	エジソ	全角カタカナの第 2 バイトは、7F より小さい。83 は、その直後に出現した。
8F 76 8D 48	竣工	8D は、漢字の第 2 バイトの直後に出現した。
83 54 83 43 83 66	サイデ	全角カタカナの第 2 バイトは、7F より小さい。83 は、その直後に出現した。
89 5E 93 5D	運転	93 は、漢字の第 2 バイトの直後に出現した。
83 5E 83 4E 83 56 81 5B	タクシー	全角カタカナの第 2 バイトは、7F より小さい。83 も 81 もその直後に出現した。
8B 60 90 6C	義人	90 は、漢字の第 2 バイトの直後に出現した。
92 6E 90 6B	地震	90 は、漢字の第 2 バイトの直後に出現した。
8C 76 8E 5A	計算	8E は、漢字の第 2 バイトの直後に出現した。
34 89 7E 81 7E	4 円×	89 は、ASCII の数字の直後に出現した。81 は、漢字の第 2 バイトの直後に出現した。
90 48 97 7E	食欲	97 は、漢字の第 2 バイトの直後に出現した。
83 7C 83 70 83 43	ポパイ	全角カタカナの第 2 バイトは、7F より小さい。83 は、その直後に出現した。



### 7Fh より大きいなら下線を表示

-w127 付きの obot コマンドは、7Fh より大きい数の代用として下線を表示します。Gongitsune.txt の第 7 行まで表示してみました。「中山《なかやま》というところに小さなお城があつて」の山《は、下図の 52 \_\_ 73 に対応しています。7Fh より大きい数は、連続しないで 1 個のみ出現しています。中山《の二重山括弧(ギュメ)の第 1 バイトは、81h であり 7Fh より大きい。二重山括弧の第 2 バイトは、73h であり 7Fh より小さい。

```
C:¥Q>obot -w127 -m7 Gongitsune.txt
_ _ _ _ _ 0D 0A _ 56 _ _ _ _ _ 67 0D 0A 0D 0A
_ _ _ _ _ 0D 0A 0D 0A _ 40 _ _ _ _ _ 41 _ _ _ _ _ 73
_ _ _ _ _ 74 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ 41 _ _ _ _ _ _ _ _ _ _ _ 73 _ _ _ _ _
_ _ _ _ _ 74 _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ 40 _ _ _ _ _ 62 _ _ _ _ _ 42 0D 0A
_ _ _ _ _ _ _ _ _ _ _ 41 _ _ _ _ _ 52 _ _ 73
_ _ _ _ _ _ _ _ _ _ _ 74 _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ 41 _ _ _ _ _ 52 _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ 41 _ _ _ _ _ _ _ _ _ _ _
_ _ _ _ _ _ _ _ _ _ _ 42 0D 0A
C:¥Q>
```

## sjisutf16 コマンドの変換表

MS932~UTF16.bin にしたがって、sjisutf16 コマンドは、シフト JIS のテキストファイルを UTF-16 のファイルに変換します。

### どの変換表を使用するか

## C:¥Ssjisutf16

に sjisutf16.exe および MS932~UTF16.bin をインストールしました。通常は、sjisutf16 コマンドをインストールしたディレクトリにある MS932~UTF16.bin にアクセスできます。どの変換表を使用するか表示するには、sjisutf16 コマンドのパラメータとして、-w を付加してください。このデータは、data.txt に保存されています。

```
C:¥Q>sjisutf16 -w Gongitsune.txt
C:¥Sjisutf16¥MS932~UTF16.bin is primarily accessible to Sjisutf16.

C:¥Q>TYPE data.txt
C:¥Sjisutf16¥MS932~UTF16.bin is primarily accessible to Sjisutf16.
C:¥Q>_
```

### カレントディレクトリ

現在のディレクトリをカレントディレクトリと言います。sjisutf16 コマンドをインストールしたディレクトリから作業ディレクトリ(C:¥Q)に、MS932~UTF16.bin を移動できます。もはやインストールディレクトリから MS932~UTF16.bin を発見できません。CD コマンドで、作業ディレクトリをカレントディレクトリにしてください。

### 削除されるおそれ

カレントディレクトリでは、何らかの誤操作のため、MS932~UTF16.bin が削除されるおそれがあります。バックアップを目的として、事前に適当なディレクトリに MS932~UTF16.bin のコピーを作成することを推奨します。カレントディレクトリの変換表を使用する場合、sjisutf16 コマンドは、変換表のパス名ではなくファイル名を表示することを下図に示します。

```
C:¥Q>CD ¥Sjisutf16

C:¥Sjisutf16>MOVE MS932~UTF16.bin C:¥Q
1 個のファイルを移動しました。

C:¥Sjisutf16>CD ¥Q

C:¥Q>sjisutf16 -w Gongitsune.txt
MS932~UTF16.bin is primarily accessible to Sjisutf16.

C:¥Q>_
```

## 環境変数 Path

インストールディレクトリにもカレントディレクトリにも変換表を発見できなかった場合、sjisutf16 コマンドは、環境変数 Path に記載の各ディレクトリを探索します。発見されるまで多数のエラーファイルを作成します。err-通し番号-分-秒.txt 形式で各エラーファイルが命名されます。

### 通し番号をリセット

sjisutf16 コマンドは、serial.txt を読み込んで、数値に 1 を加算して、serial.txt に加算の結果を書き出します。serial.txt を削除するか、

serial.txt に 0 を記入することのいずれかによって、通し番号をリセットできます。

## 設定ファイル

sjisutf16 コマンドは、設定ファイルが不要ですが、カレントディレクトリで設定ファイルを開くことができない場合、エラーファイルを作成します。設定ファイルは、set\_sjisutf16.txt です。これは、シフト JIS のテキストファイルです。

### 第 1 行に変換表のパス名

設定ファイルの第 1 行に変換表のパス名を記入してください。第 2 行には、たとえば、-0 を記入してください。-0 は、0Dh および 0Ah で改行することを意味します。第 3 行には、必要に応じて、-w を記入してください。以下同様です。コマンドプロンプトには、パラメータを空白で区切って並べますが、設定ファイルには、各行に 1 個のパラメータを記入してください。各行は、パラメータ、タブ、注釈の順に記入してください。



## 変換表を探索する順序

sjisutf16 コマンドは、下記の順序で変換表を探索します。

1. sjisutf16 コマンドがインストールされたディレクトリ
2. カレントディレクトリ
3. 環境変数 Path に記載の各ディレクトリ
4. 設定ファイルの第 1 行に記載のパス名

### 設定ファイルに記載のパス名を優先

インストールディレクトリ、カレントディレクトリ、Path に記載の各ディレクトリに変換表があろうがなかろうが、設定ファイルの第 1 行からパス名を捕捉して、変換表を開くには、sjisutf16 コマンドのパラメータとして、-a を付加してください。

```
C:¥Q>TYPE set_sjisutf16.txt
C:¥Narfi¥MS932~UTF16.bin
-0      CRLF

C:¥Q>sjisutf16 -a -w -o date.txt Gongitsune.txt
C:¥Narfi¥MS932~UTF16.bin is primarily accessible to Sjisutf16.

C:¥Q>DIR dat*.txt | FINDSTR "dat"
2020/05/13  18:38                62 data.txt
2020/05/13  18:38            10,858 date.txt

C:¥Q>
```

### 変換表の探索を抑制

sjisutf16 コマンドが変換表を探索することを抑制するには、sjisutf16 コマンドのパラメータとして、`-d` を付加してください。

### 変換表を生成

UTF16~MS932.bin が削除されるおそれがあるなど、バックアップを目的として、事前に適当なディレクトリに UTF16~MS932.bin のコピーを作成することを推奨しますが、コマンドプロンプトに入力するとき、何もパラメータが付加されていない utf16sjis コマンドは、MS932~UTF16.bin から UTF16~MS932.bin を生成できます。カレントディレクトリに MS932~UTF16.bin を用意してください。

## 制御文字

制御文字としては、MS-DOS テキストファイルの終わり(1Ah)、エスケープシーケンスの切替え(1Bh)、ベル(07h)、改ページ、タブ、改行、復帰などがあります。

### ワードパッドは改ページを認識する

Windows のアクセサリには、メモ帳だけではなくワードパッドが含まれています。ワードパッドで FormFeed.txt を開くことができます。サクラエディタと異なり、ワードパッドは、改ページを認識します。



十六進数	数値	英語	略号	説明
07h	7	alert	BEL	ベル、警告
08h	8	back space	BS	バックスペース、削除
09h	9	horizontal tab	HT	タブ、水平タブ
0Ah	10	line feed	LF	改行
0Bh	11	vertical tab	VT	垂直タブ
0Ch	12	form feed	FF	改ページ
0Dh	13	carriage return	CR	復帰
1Ah	26	end of file	EOF	MS-DOS テキストファイルの終わり
		substitute	SUB	置換
1Bh	27	escape	ESC	エスケープシーケンスの切替え

## 制御文字を削除

09h, 0Ch, 1Bh などの制御文字を削除するには、sjisutf16 コマンドのパラメータとして、`-s` を付加してください。`-s` 付きの `sjisutf16` コマンドで 0Ch が削除された例を下図に示します。

```

C:¥Q>obot FormFeed.txt
50 61 67 65 2D 31 0C 50 61 67 65 2D 32 0C 45 4F 46
C:¥Q>sjisutf16 -s FormFeed.txt FormFeed_UTF-16.txt

C:¥Q>obot FormFeed_UTF-16.txt
FF FE 50 00 61 00 67 00 65 00 2D 00 31 00 50 00 61 00 67 00
65 00 2D 00 32 00 45 00 4F 00 46 00
C:¥Q>

```

### 0Ah のみで改行

0Dh を削除するには、`sjisutf16` コマンドのパラメータとして、`-LF` を付加してください。

# UTF-8 の HTML

近年、インターネットでは、シフト JIS でも EUC-JP でもなく UTF-8 が多用されます。Chrome, Edge, Firefox, Safari などのウェブブラウザは、UTF-8 のテキストファイルをサポートしています。UTF-8 は、Unicode(ユニコード)の一種であり、橋の名前、地名、人名、屋号など、名前に含まれた珍しい漢字を表現できます。

## HTML を保存

たとえば、Firefox でウェブページを表示しているとき、ウェブページのどこか適当な場所で右クリックすれば、メニューが表示されます。メニューから「名前を付けてページを保存」という項目を選択して、UTF-8 のテキストファイルをダウンロードできます。

### ハンバーガーメニュー

右クリックしないで、ハンバーガー(三、hamburger.png)をクリックしてもメニューから「名前を付けてページを保存」という項目を選択できます。

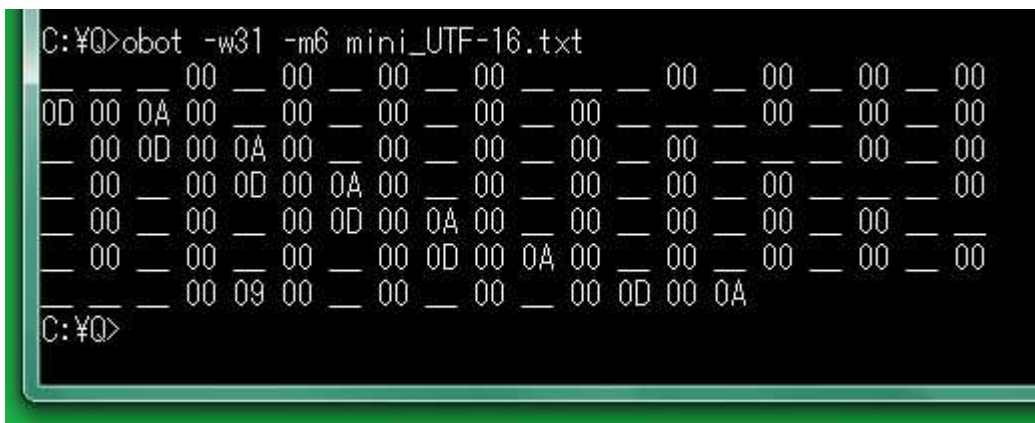
## obot コマンド

添付の obot コマンドは、ファイルの最初の部分を十六進数で表現します。

### 上限

上限より大きい数の代用として下線を表示できます。たとえば、十六進数で制御文字のみ表現するには、obot コマンドのパラメータとして、-w31 を付加してください。

mini\_UTF-16.txt の第 6 行で、09h が出現している例を下図に示します。



```
C:¥Q>obot -w31 -m6 mini_UTF-16.txt
  00 00 00 00 00 00 00 00 00 00 00 00 00 00
0D 00 0A 00 00 00 00 00 00 00 00 00 00 00
  00 0D 00 0A 00 00 00 00 00 00 00 00 00 00
  00 00 0D 00 0A 00 00 00 00 00 00 00 00 00
  00 00 00 00 0D 00 0A 00 00 00 00 00 00 00
  00 00 00 00 00 0D 00 0A 00 00 00 00 00 00
  00 00 09 00 00 00 00 00 0D 00 0A
C:¥Q>
```

## NOTEPAD コマンド

どの Windows も最初からテキストエディタとしてメモ帳を用意しています。NOTEPAD コマンドでメモ帳を起動できます。メモ帳には、0Dh および 0Ah で改行されたテキストが適しています。

# 珍しい漢字

Windows のコマンドプロンプトは、シフト JIS が多用されます。MS932 は、いわゆるシフト JIS ですが、規格外です。たとえば、丸付きの数字(①②③⑩⑳)、ローマ数字(ⅠⅡⅢⅣⅤ)、1 バイトのカタカナは、EUC-JP で表現しにくい。一方、MS932 は、名前の漢字が不十分であるため、名前に含まれた珍しい漢字を表現しにくい。

## 橋の名前

江戸幕府直轄の領地、すなわち天領として倉敷が栄えました。明治時代に、倉敷では、画家の児島虎次郎がエル・グレコ、モネらの芸術品を収集しました。倉敷川にかかる石橋として、今橋、中橋、高砂橋、前神橋、入船橋があります。大原美術館は、今橋のたもとにあります。令和元年 12 月 21 日、今橋に行って撮影しました。



### 符号位置の上位サロゲートおよび下位サロゲート

この珍しい漢字、上位サロゲート、下位サロゲート、符号位置(コードポイント)は、以下の通りです。

文字	上位サロゲート	下位サロゲート	符号位置
今	D86D	DF46	2B746

## UTF-16

UTF-8 と同様に、UTF-16 も、橋の名前、地名、人名、屋号など、名前に含まれた珍しい漢字を表現できます。このユーザズガイドでは、UTF-16 は、UTF-16LE を意味します。すなわち、BOM 付き UTF-16 のファイルの最初に FFh, FEh の 2 バイトがこの順に出現します。

## テキストファイル

テキストファイルは、テキストエディタで編集しやすいファイルです。EUC-JP, シフト JIS, UTF-8 などのテキストファイルは、00h が含まれていませんが、UTF-16 のファイルは、00h が含まれている場合があります。

### テキストエディタ

Windows 用のテキストエディタとしては、EmEditor, K2, Mery, MIFES, TeraPad, WZ, サクラエディタ、秀丸、メモ帳などがあります。インターネットからダウンロードした HTML のテキストファイルを開覧するには、0Ah のみで改行されたテキストも正しく表示できるテキストエディタを選択してください。

## UTF-8 から UTF-16 に変換

添付の utf8to16 コマンドは、UTF-8 のテキストファイルを UTF-16 のファイルに変換します。BOM 付き UTF-16 のファイルが FFh, FEh で開始することを obot コマンドで確認できます。下記の UTF-16 のファイルは、00h が含まれています。

### HTML ファイルの文字コード

UTF-8 のテキストファイルを UTF-16 のファイルに変換することで HTML ファイルを作成した場合、もはや UTF-8 の HTML ファイルではないため、ファイルの全体にわたって、UTF-8 を UTF-16 に訂正してください。UTF-16 の CSS ファイルを用意してください。

### CSS ファイルの文字コード

UTF-8 のテキストファイルを UTF-16 のファイルに変換することで CSS ファイルを作成した場合、もはや UTF-8 の CSS ファイルではないため、CSS ファイルの文字コード指定として、UTF-8 を UTF-16 に訂正してください。

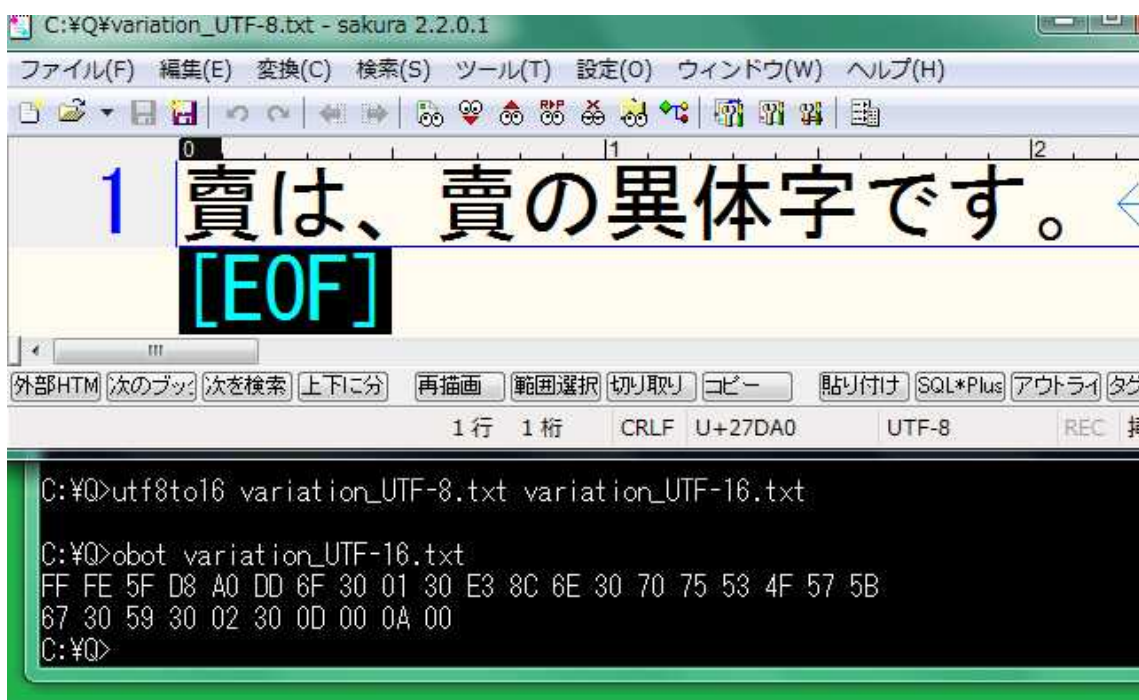


### 変換表が不要である

UTF-8 も UTF-16 も Unicode の一種であるため、utf8to16 コマンドは、まず符号位置を算出して、次に符号位置から計算して UTF-16 の形式で出力します。変換表を使用しません。

### サロゲートペアをサポート

utf8to16 コマンドは、サロゲートペアをサポートします。サクラエディタのフォント設定で、サイズを 36 に設定して、27DA0 の符号位置にある漢字を表示した例を示します。サロゲートペアは、D85F,DDA0 です。なお、賣は、8CE3 の符号位置にあります。



## BOM

Windows パソコンは、メモリ、ディスクに 16 ビットの整数を記憶するとき、下位 8 ビット、上位 8 ビットの順に並べて記憶します。すなわち、2 バイトの整数を記憶するとき、下位バイト、上位バイトの順に並べて記憶します。これをリトルエンディアンのアーキテクチャーと言います。Intel Core i5 は、リトルエンディアンのプロセッサです。FFh, FEh のバイト順マーク (BOM, byte order mark) を UTF-16 ファイルの先頭に付与することで、リトルエンディアンのアーキテクチャーに適していることを明示できます。

### ビッグエンディアン

FEh, FFh のバイト順マークが付与されていれば、2 バイトずつ記憶されたデータは、ビッグエンディアンのアーキテクチャーに適しています。たとえば、Solaris 10 という OS の Sun SPARC は、ビッグエンディアンのプロセッサです。

## iconv コマンド

文字コードを変換するために、iconv コマンドが GNU Win32 に用意されています。インターネットで、たとえば、libiconv-1.9.2-1 などのインストーラーを検索してください。接頭辞の lib は、ライブラリを意味します。このインストーラーをダウンロードして、実行して、32 ビット版 Windows 7 のパソコンに iconv.exe をインストールしました。

## DIR コマンド

DIR コマンドで、サイズに関して昇順に行を並べ替えるには、DIR コマンドのパラメータとして、/O:S を付加してください。どの Windows も最初から DIR コマンドを用意しています。

### 日時

DIR コマンドで、日時に関して昇順に行を並べ替えるには、DIR コマンドのパラメータとして、/O:D を付加してください。

## COPY コマンド

COPY コマンドで 2 個のファイルをバイナリファイルとして連結する方法で、BOM 付き UTF-16 のファイルを作成しました。COPY コマンドは、どの Windows にも最初から用意されています。utf8to16 コマンドの結果と一致したことを下図に示します。

```
C:\>obot FFFE.bin
FF FE
C:\>iconv -f UTF-8 -t UTF-16LE WikiTable_UTF-8.txt > text.tmp

C:\>COPY /B FFFE.bin+text.tmp text_with_BOM.tmp
FFFE.bin
text.tmp
    1 個のファイルをコピーしました。

C:\>utf8to16 -LF WikiTable_UTF-8.txt WikiTable_UTF-16.txt

C:\>COMP text_with_BOM.tmp WikiTable_UTF-16.txt
text_with_BOM.tmp と WikiTable_UTF-16.txt を比較しています...
ファイルに違いはありません

(ほかのファイルを比較しますか (Y/N)? n
C:\>
```

### UTF-8 のファイルは大きい

下記の例では、iconv コマンドでシフト JIS のテキストファイルを変換しました。0Dh およ

び 0Ah の両方で改行しています。さらに、utf8to16 で UTF-8 のテキストファイルを UTF-16 のファイルに変換しました。たいてい UTF-16 のファイルでは、1 文字の漢字に 2 バイトが必要ですが、UTF-8 のテキストファイルでは、1 文字の漢字に 3 バイトも必要です。UTF-16 のファイルは、UTF-8 のテキストファイルより小さい。

```
C:¥Q>iconv -f Shift_JIS -t EUC-JP Gongitsune.txt > Gon_EUC-JP.txt
C:¥Q>iconv -f Shift_JIS -t UTF-8 Gongitsune.txt > Gon_UTF-8.txt
C:¥Q>utf8to16 Gon_UTF-8.txt Gon_UTF-16.txt

C:¥Q>DIR /O:S Gon*.txt | FINDSTR "Gon
2020/03/24 10:17          10,636 Gon_EUC-JP.txt
2018/10/11 13:53          10,636 Gongitsune.txt
2020/03/24 10:18          10,858 Gon_UTF-16.txt
2020/03/24 10:17          15,844 Gon_UTF-8.txt

C:¥Q>
```

#### utf16sjis コマンド

surro.csv および UTF16~MS932.bin にしたがって、utf16sjis コマンドは、UTF-16 のファイルをシフト JIS のテキストファイルに変換します。UTF16~MS932.bin と異なり、surro.csv は、utf16sjis コマンドによって探索されません。カレントディレクトリで surro.csv を開くことができない場合、utf16sjis コマンドは、自動的に surro.csv を作成します。

#### UTF-16 のファイルは大きい

下記の例では、utf8to16 コマンドで UTF-8 のテキストファイルを UTF-16 のファイルに変換しました。この UTF-8 ファイルは、内容が ASCII のみからなり、しかも 0Ah のみで改行されているため、シフト JIS のテキストファイルより小さい。UTF-8 のファイルでは、1 文字の ASCII に 1 バイトが必要ですが、UTF-16 のテキストファイルでは、1 文字の ASCII に 2 バイトも必要です。UTF-16 のファイルは、UTF-8 のテキストファイルより大きい。utf16sjis コマンドで UTF-16 のテキストファイルをシフト JIS のファイルに変換しました。

```
C:¥Q>obot -w127 -m1 isbn_UTF-8.csv
32 30 38 2C 32 30 31 31 2C 39 2E 39 30 6D 6D 2C 49 53 42 4E
20 39 37 38 2D 31 2D 38 34 39 30 35 2D 31 35 33 2D 38 2C 4A
65 73 73 69 63 61 20 4B 69 6E 67 73 6C 65 79 20 50 75 62 6C
69 73 68 65 72 73 2C 53 65 63 75 72 69 6E 67 20 41 70 70 72
6F 70 72 69 61 74 65 20 45 64 75 63 61 74 69 6F 6E 20 50 72
6F 76 69 73 69 6F 6E 20 66 6F 72 20 43 68 69 6C 64 72 65 6E
20 77 69 74 68 20 41 75 74 69 73 6D 20 53 70 65 63 74 72 75
6D 20 44 69 73 6F 72 64 65 72 73 2C 41 20 47 75 69 64 65 20
66 6F 72 20 50 61 72 65 6E 74 73 20 61 6E 64 20 50 72 6F 66
65 73 73 69 6F 6E 61 6C 73 2C 42 79 2C 41 6C 6C 69 73 6F 6E
20 48 6F 70 65 2D 57 65 73 74 2C 0A 32 30 33 2C 32 30 30 39
2C 32 32 2E 30 30 6D 6D 2C 49 53 42 4E 20 39 37 38 2D 31 2D
34 30 32 30 2D 39 38 33 30 2D 37 2C 53 70 72 69 6E 67 65 72
2C 22 54 68 65 20 48 61 6E 64 62 6F 6F 6B 20 6F 66 20 4E 65
75 72 6F 70 73 79 63 68 69 61 74 72 69 63 20 42 69 6F 6D 61
72 6B 65 72 73 2C 20 45 6E 64 6F 70 68 65 6E 6F 74 79 70 65
73 20 61 6E 64 20 47 65 6E 65 73 22 2C 4E 65 75 72 6F 61 6E
61 74 6F 6D 69 63 61 6C 20 61 6E 64 20 4E 65 75 72 6F 69 6D
61 67 69 6E 67 20 45 6E 64 6F 70 68 65 6E 6F 74 79 70 65 73
20 61 6E 64 20 42 69 6F 6D 61 72 6B 65 72 73 2C 42 79 2C 4D
69 63 68 61 65 6C 20 53 2E 20 52 69 74 73 6E 65 72 2C 0A
C:¥Q>utf8to16 isbn_UTF-8.csv isbn_UTF-16.csv

C:¥Q>utf16sjis isbn_UTF-16.csv isbn_Shift_JIS.csv

C:¥Q>DIR /O:S isbn*.csv | FINDSTR "isbn"
2019/12/24 15:59          62,665 isbn_UTF-8.csv
2020/03/24 10:27          63,185 isbn_Shift_JIS.csv
2020/03/24 10:26        126,372 isbn_UTF-16.csv

C:¥Q>
```

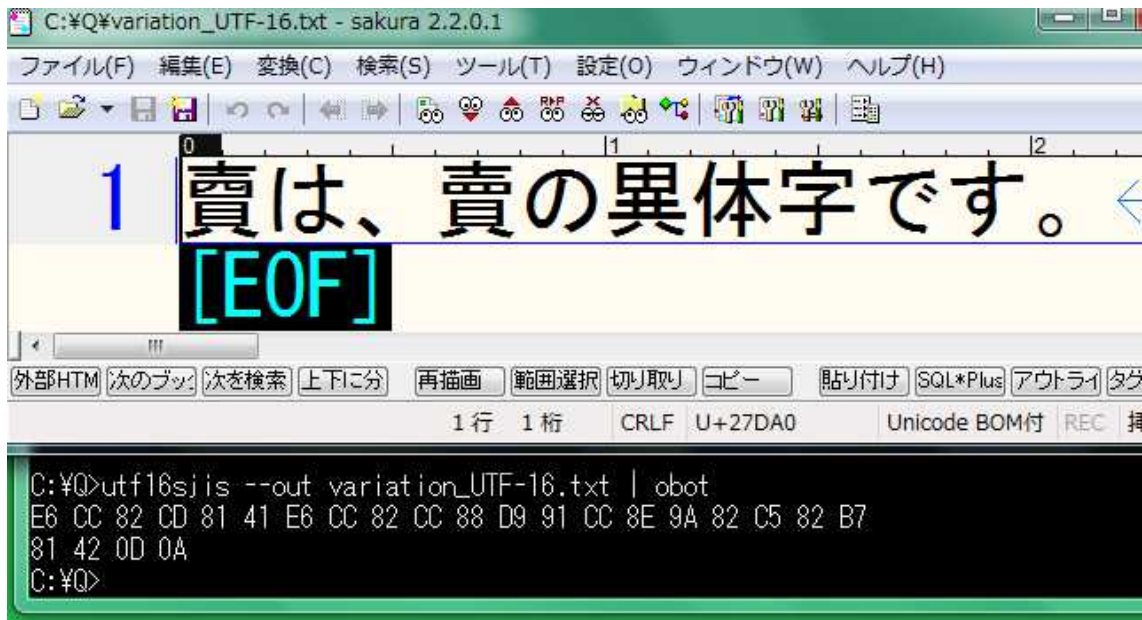
### 元に戻った

変換、逆変換で元に戻った例を下図に示します。まず sjisutf16 コマンドでシフト JIS のテキストファイルである Gongitsune.txt を BOM 付き UTF-16 ファイルに変換して、次に utf16sjis コマンドで UTF-16 ファイルをシフト JIS テキストファイルに変換しました。結果のファイルを Gon\_Shift\_JIS.txt と命名しました。Gon\_Shift\_JIS.txt は、内容が Gongitsune.txt に一致しました。サイズも同一でした。

```
C:¥Q>sjisutf16 Gongitsune.txt Gon_UTF-16.txt
C:¥Q>utf16sjis Gon_UTF-16.txt Gon_Shift_JIS.txt
C:¥Q>COMP Gongitsune.txt Gon_Shift_JIS.txt
Gongitsune.txt と Gon_Shift_JIS.txt を比較しています...
ファイルに違いはありません
(ほかのファイルを比較しますか (Y/N)? n
C:¥Q>
```

## 標準出力

utf16sjis コマンドは、標準入力から読み込むことができません。標準出力にシフト JIS のデータを書き出すには、utf16sjis コマンドのパラメータとして、--out を付加してください。--out は、-o- -z と同じ結果を生じます。27DA0 の符号位置にある文字(サロゲートペア)をシフト JIS にある文字で代替した例を下図に示します。シフト JIS として E6CC が 2 回も出現しています。シフト JIS のバイト列は、上位バイトが下位バイトより先に出現します。文意は、無意味になりましたが、この変換結果は、正しい。



## サロゲートペアの代替文字

utf16sjis コマンドは、surro.csv および UTF-16~MS932.bin にしたがって、UTF-16 のファイルをシフト JIS のテキストファイルに変換します。下線( \_ )は、シフト JIS の 2 バイト文字に含まれています。テキストにサロゲートペアが出現した場合、

utf16sjis コマンドは、サロゲートペアをたとえば下線に置換しますが、利用者は、シフト JIS の 2 バイト文字から新しい代替文字を選択できます。

#### surro.csv

添付の surro コマンドは、surro.csv を作成します。surro.csv は、BOM 付きの UTF-16 ファイルです。

#### 手順

1. 2000B, 20089, 200A2 など十六進数でサロゲートペアの符号位置(コードポイント)を表現してください。
2. surro.txt の各行に 1 個の十六進数を記入してください。
3. surro コマンドで、surro.csv を作成してください。
4. surro.csv の各行に新しい代替文字を記入してください。

#### 初期化される

surro コマンドで、surro.csv は、上書きされます。バックアップを目的として、事前に適切なディレクトリに surro.csv のコピーを作成することを推奨します。FPC 版の utf16sjis コマンドは、surro.csv から SurroSJIS.csv を作成します。バックアップを目的として、事前に適切なディレクトリに SurroSJIS.csv のコピーを作成することを推奨します。

```
C:¥Q>DEL surro*.*

C:¥Q>COPY C:¥Sjisutf16¥test_files¥surro*.*
C:¥Sjisutf16¥test_files¥surro.csv
C:¥Sjisutf16¥test_files¥surro.txt
    2 個のファイルをコピーしました。

C:¥Q>utf16sjis --out variation_UTF-16.txt
賣は、賣の異体字です。

C:¥Q>DIR surro*.* | FINDSTR "urro"
2019/12/23  13:54                17,790 surro.csv
2019/12/23  13:45                 2,480 surro.txt
2020/05/13  16:54                 9,204 SurroSJIS.csv

C:¥Q>DEL surro*.csv

C:¥Q>surro

C:¥Q>utf16sjis --out variation_UTF-16.txt
_ は、賣の異体字です。

C:¥Q>
```

## サロゲートペアを下線に置換

utf16sjis コマンドがサロゲートペアをシフト JIS の代替文字に変換するように、利用者は、surro.csv に代替文字を記入できますが、utf16sjis コマンドがサロゲートペアをすべて、シフト JIS の下線に変換しても利用者が困らない場合、surro.csv は、不要であるかもしれません。この場合、utf16sjis コマンドのパラメータとして、-u を付加できます。

### -u 付きの utf16sjis コマンド

-u が付加された場合、utf16sjis コマンドは、サロゲートペアをすべて、シフト JIS の下線に変換します。-u 付きの utf16sjis コマンドは、surro.csv が不要です。ただし、surro コマンドと同様に、-u 付きの utf16sjis コマンドは、surro.txt が必要です。

```
C:¥Q>DIR surro*.* | FINDSTR "urro"
2019/12/23 13:45                2,480 surro.txt

C:¥Q>utf16sjis -u --out variation_UTF-16.txt
_は、賣の異体字です。

C:¥Q>
```

### surro.csv から surro.txt を作成

カレントディレクトリで surro.txt を開くことができない場合、utf16sjis コマンドは、自動的に surro.csv を読み込んで、第 4 欄のみ選択して、surro.txt を書き出します。

```
C:¥Q>DIR surro*.* | FINDSTR "urro"
2019/12/23 13:54                17,790 surro.csv

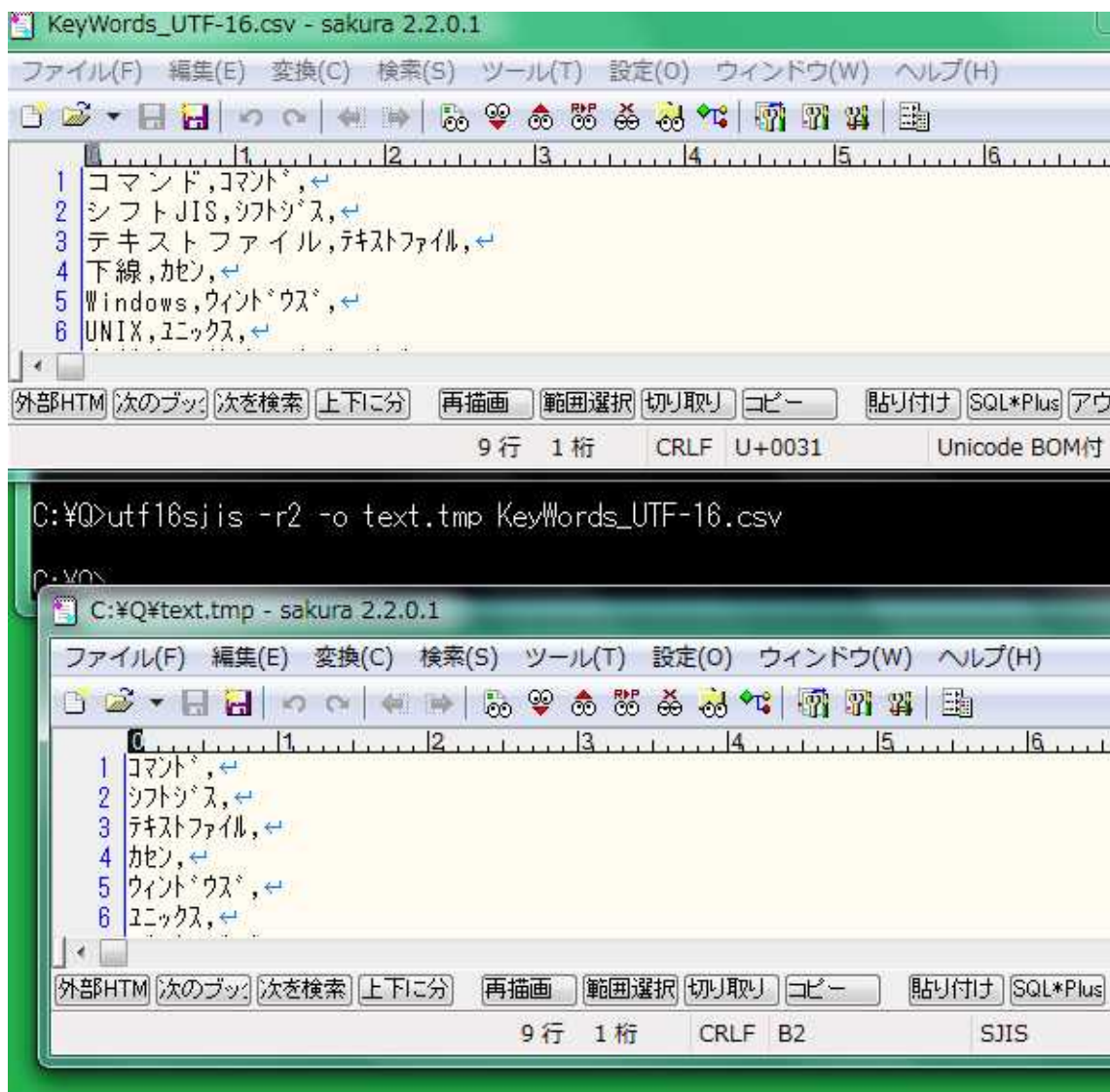
C:¥Q>utf16sjis -u --out variation_UTF-16.txt
_は、賣の異体字です。

C:¥Q>DIR surro*.* | FINDSTR "urro"
2019/12/23 13:54                17,790 surro.csv
2020/05/13 23:16                2,480 surro.txt

C:¥Q>
```

## ある欄を選択

utf16sjis コマンドが UTF-16 の CSV ファイルを読み込んで、たとえば、第 2 欄のみ選択して、シフト JIS のテキストファイルを書き出すには、utf16sjis コマンドのパラメータとして、`-r2` を付加してください。



## HTML を閲覧

まず `surro.txt` の各行に 1 個の十六進数を記入してください。`surro.txt` は、シフト JIS のテキストファイルです。次に `surro` コマンドで、`surro.csv` を作成してください。一般論として、ウェブブラウザで UTF-16 のファイルを閲覧できます。`surro` コマンドは、カレントディレクトリで `cut_here.html` を開くことができた場合、`surro.csv` だけではなく `surro.html` も作成します。

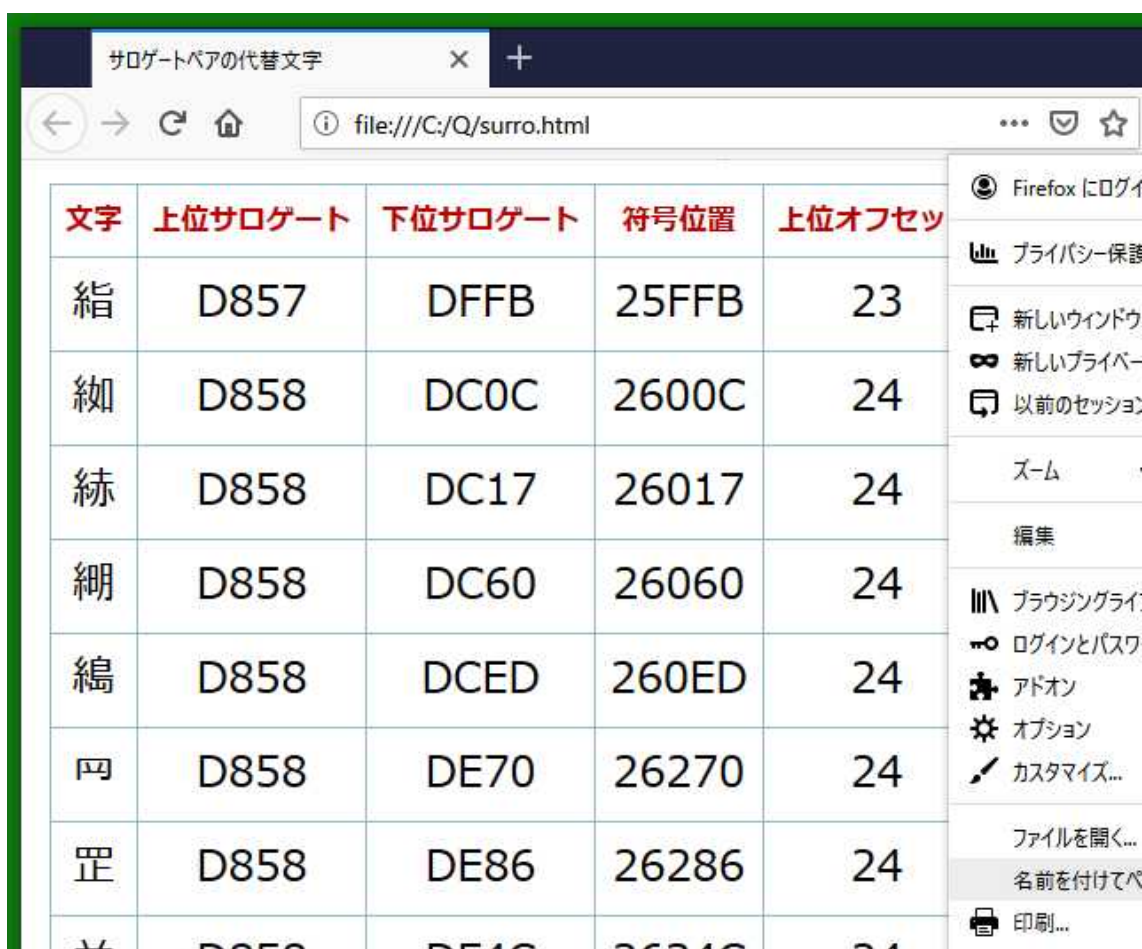


### HTML は変換に必要ではない

カレントディレクトリで cut\_here.html を開くことができない場合、surro コマンドは、surro.csv のみ作成して、正常に終了します。

### HTML は表作成が可能

ウェブブラウザで HTML を閲覧することで、**表の作成および確認**が可能です。同一のディレクトリに Cut\_here\_style\_UTF-16.css および surro.html の両方が必要です。Utf16sjis.png も用意します。surro.html を閲覧するには、たとえば、エクスプローラで surro.html を選択して、ダブルクリックしてください。Windows 10 のパソコンで、Firefox で surro.html を開いて、画面を保存して、画面の画像から表の一部を切り出した例



文字	上位サロゲート	下位サロゲート	符号位置	上位オフセット
紺	D857	DFFB	25FFB	23
紉	D858	DC0C	2600C	24
紕	D858	DC17	26017	24
紕	D858	DC60	26060	24
紕	D858	DCED	260ED	24
𠄎	D858	DE70	26270	24
𠄎	D858	DE86	26286	24
𠄎	D858	DE1C	2621C	24

### 各行に新しい代替文字を記入してください

surro コマンドを実行した結果として、surro.csv および surro.html が作成されますが、代替文字として下線を使用する設定になります。利用者が、下線ではなく異体字を記入した例を下図に示します。異体字のうち、シフト JIS に含まれる漢字を記入しました。下図は、Windows 10 のメモ帳で CSV ファイルを表示しました。



## リダイレクト

標準出力をファイルにリダイレクトすることで、ファイルを作成できます。この場合、標準出力の内容は、画面に表示されないで、ファイルに書き出されます。

### NUL にリダイレクト

NUL がファイルであるかのように、標準出力を NUL にリダイレクトすることで、標準出力の内容を画面に表示しないことが可能です。どの Windows も最初から NUL を用意しています。



## 標準エラー出力を NUL にリダイレクト

NUL がファイルであるかのように、標準エラー出力を NUL に 2> でリダイレクトできます。この場合、標準エラー出力は、画面に表示されません。

```
C:¥Q>ECHO your private life > text.tmp
C:¥Q>DEL text.tmp
C:¥Q>DEL text.tmp
C:¥Q¥text.tmp が見つかりませんでした。
C:¥Q>DEL text.tmp 2> NUL
C:¥Q>
```

## バージョン

32ビット版の Windows Vista において、Free Pascal Compiler 3.0.4 で Sjisutf16 を構築しました。バージョン番号を表示するには、sjisutf16 コマンドのパラメータとして、-V を付加してください。

### ガイド

sjisutf16 コマンドの用例および短い説明を表示するには、sjisutf16 コマンドのパラメータとして、-G または --help または -VC を付加してください。

パラメータ	ほぼ同等な入力	説明
-CRLF	-0	0Dh および 0Ah で改行する。
--help	-G	用例および短い説明を表示する。
	-VC	
-LF	-0- -L	0Ah のみで改行する。
--out	-o- -z	標準出力に書き出す。
-VC	-G	用例および短い説明を表示する。
	--help	

## 複合パラメータ

たとえば、-b -w -y の 3 個のパラメータを統合して -bwy という 1 個の複合パラメータを sjisutf16 コマンドに付加できます。この場合、sjisutf16 コマンドは、自動的に -bwy を分解して、-b -w -y の 3 個のパラメータを生成します。-byw も同一の結果を生じます。。-ybw, -ywb, -wby, -wyb の各複合パラメータも同一の結果を生じます。

```
C:¥Q>DEL binary*.tmp 2> NULL

C:¥Q>sjisutf16 -bwy Gongitsune.txt Gon_bwy.txt
C:¥Sjisutf16¥MS932~UTF16.bin is primarily accessible to Sjisutf16.

C:¥Q>DIR binary.tmp | FINDSTR "binary"
2020/03/25  20:22                10,858 binary.tmp

C:¥Q>REN binary.tmp binary_bwy.tmp

C:¥Q>sjisutf16 -ywb Gongitsune.txt Gon_ywb.txt
C:¥Sjisutf16¥MS932~UTF16.bin is primarily accessible to Sjisutf16.

C:¥Q>DIR binary.tmp | FINDSTR "binary"
2020/03/25  20:23                10,858 binary.tmp

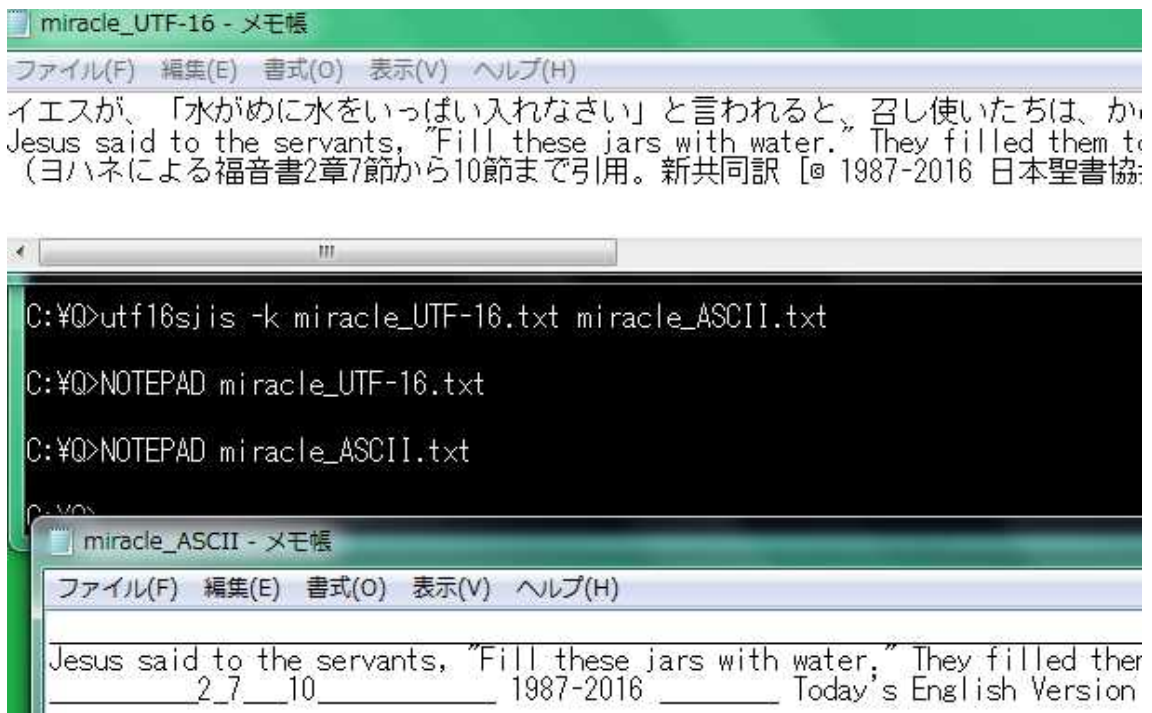
C:¥Q>FC Gon_bwy.txt Gon_ywb.txt
ファイル Gon_bwy.txt と GON_YWB.TXT を比較しています
FC: 相違点は検出されませんでした

C:¥Q>FC binary_bwy.tmp binary.tmp
ファイル binary_bwy.tmp と BINARY.TMP を比較しています
FC: 相違点は検出されませんでした

C:¥Q>_
```

## 漢字を廃止

漢字、ひらがな、カタカナを廃止して、下線(   )で代用できます。たとえば、UTF-16のファイルの一部が英文である場合、しかも英文のみ関心がある場合、漢字、ひらがな、カタカナをすべて下線に置換するには、utf16sjis コマンドのパラメータとして、-k を付加してください。



## unarfij コマンド

入力ファイルとして CSV を想定しています。ある欄のみ選択して、unarfij コマンドは、細長いファイルを書き出します。surro.csv および UTF16~MS932.bin にしたがって、unarfij コマンドは、UTF-16 のファイルをシフト JIS のテキストファイルに変換します。(unarfij コマンドは、Delphi 版の unarfi および utf16sjis の組み合わせに相当します。)

### ある欄を選択

たとえば、UTF-16 の CSV ファイルの第 5 欄のみ選択して、シフト JIS のテキストファイルを書き出すには、unarfij コマンドのパラメータとして、-j5 を付加してください。

### ある欄から制御文字を削除

unarfij コマンドが UTF-16 の CSV ファイルを読み込んで、たとえば、第 2 欄から制御文字を削除して、シフト JIS のテキストファイルを書き出すには、unarfij コマンドのパラメータとして、-o2 を付加してください。

#### タブを削除

十六進数として、09h, 0Ch, 1Bh は、それぞれ、タブ、改ページ、エスケープシーケンスの切替えを意味します。テストファイルとして、mini\_UTF-16.txt は、第 2 欄に 09h, 0Ch, 1Bh が混入しています。unarfij コマンドが第 2 欄から 09h, 0Ch, 1Bh を削除した例を下図に示します。

```
C:\>unarfij -o2 mini_UTF-16.txt mini_narrow.txt
```

```
C:\>
```

The image shows two Notepad++ windows side-by-side. The left window is titled 'mini\_UTF-16.txt' and contains 28 lines of text. The right window is titled 'C:\>mini\_narrow.txt - sakura 2.2.0.1' and contains 22 lines of text, with the last 6 lines missing. The status bars at the bottom of each window provide additional information: the right window shows '6行 1桁 CRLF 38 SJIS' and the left window shows '6行 8桁 CRLF U+0009 Unicode BOM付 REC 挿入'.

Line	mini_UTF-16.txt	mini_narrow.txt
1	4429 24,	24,
2	4647 64,	64,
3	4759 28,	28,
4	4840 27,	27,
5	4851 38,	38,
6	4881 86,	86,
7	4902 59,	59,
8	4908 14,	14,
9	4944 19,	19,
10	4954 93,	93,
11	4956 01,	01,
12	4960 25,	25,
13	4967 51,	51,
14	4972 20,	20,
15	4973 41,	41,
16	4976 18,	18,
17	4983 90,	90,
18	5000 17,	17,
19	5006 39,	39,
20	5009 74,	74,
21	5014 40,	40,
22	5025 68,	68,
23	5030 70,	70,
24	5032 83,	83,
25	5035 12,	12,
26	5037 94,	94,
27	5043 15,	15,
28	5049 32,	

## 新しいファイル

UTF-16 のファイル、新しいファイルがこの順に指定された場合、unarfij コマンドは、UTF-16 のファイルを読み込んで、新しいファイルに書き出します。新しいファイルは、MS932 のテキストファイルです。MS932 は、いわゆるシフト JIS です。

### 先に新しいファイルの名前を指定したい

上記のとおり、2 個のファイルを指定した場合、2 個目のファイルが新しいファイルになりますが、先に新しいファイルの名前を指定するには、unarfij コマンドのパラメータとして、`-F text.tmp` を付加してください。`-F` および `text.tmp` の間に空白を挿入してください。`text.tmp` は、新しいファイルの名前です。

## 0Dh および 0Ah で改行

新しいシフト JIS のファイルは、Windows のパソコンで使用されることを想定しているため、パラメータとして何も指定されていない場合も、unarfij コマンドは、`-0` が指定されたとみなします。

### 0Ah のみで改行

`-0` が指定されても、さらに`-0-`を付加することで、`-0` を無効にすることができます。`0Ah` のみで改行するように命令するには、unarfij コマンドのパラメータとして、`-LF` を付加してください。`-LF` は、`-0- -L` を意味します。

### `-LF` と同等なパラメータ

`-LF` は、`-0- -L`(マイナスゼロマイナス空白マイナスエル)と同じ結果を生じます。

パラメータ	改行	説明
<code>-0</code>	0Dh および 0Ah で改行する。	省略された場合、unarfij コマンドは、 <code>-0</code> が付加されたとみなす。
<code>-0-</code>	UTF-16 のファイルにしたがって改行する	UTF-16 ファイルの行が、000D, 000A で改行されていれば、unarfij コマンドは、0Dh および 0Ah で改行する。
<code>-0- -L</code>	0Ah のみで改行する。	インターネットでは、これでも問題ない。

### パラメータを無効にする

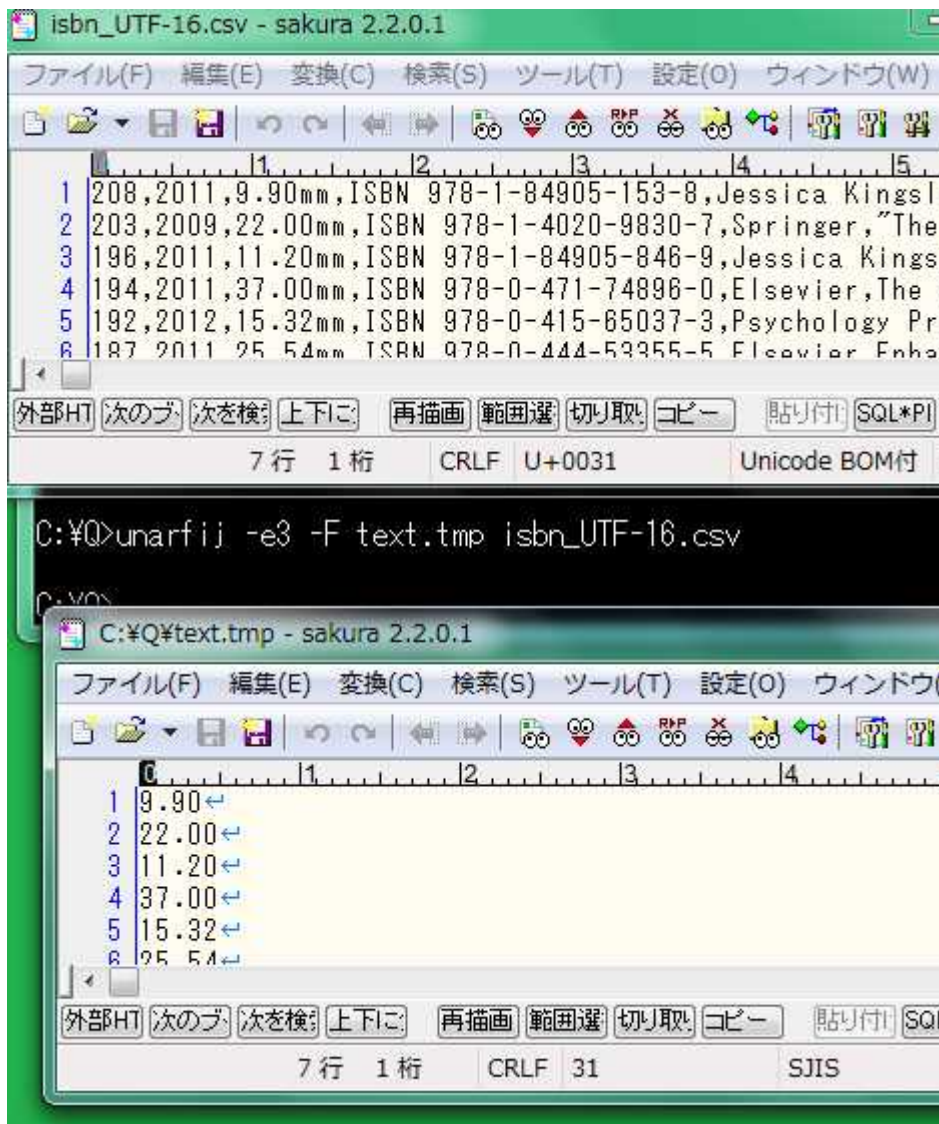
`0Ah` のみで改行したい場合、利用者は、unarfij コマンドのパラメータとして`-L` を付加できます。ただし、unarfij コマンドは、自動的に`-0` のパラメータを付加して処理します(デフォルト)。`-0` は、`-L` より優先されるため、`-L` を指定したい場合、`-0` を無効にしなければなりません。利用者は、`-0-`により`-0` を無効にすることができます。すなわち、`0Ah` のみで改行したい場合、unarfij コマンドのパラメータとして`-0- -L` を付加してください。`-LF` は、`-0- -L` と同じ結果を生じます。

## 浮動小数点数を抽出

unarfij コマンドは、ある欄から浮動小数点数を抽出できます。たとえば、第 3 欄から浮動小数点数を抽出するには、unarfij コマンドのパラメータとして、`-e3` を付加してください。

### mm が削除された

下記の例では、第 3 欄には、本の厚さとしてミリメートルの単位 (mm) が記入されていますが、`-e3` が付加された unarfij コマンドは、mm を削除しました。



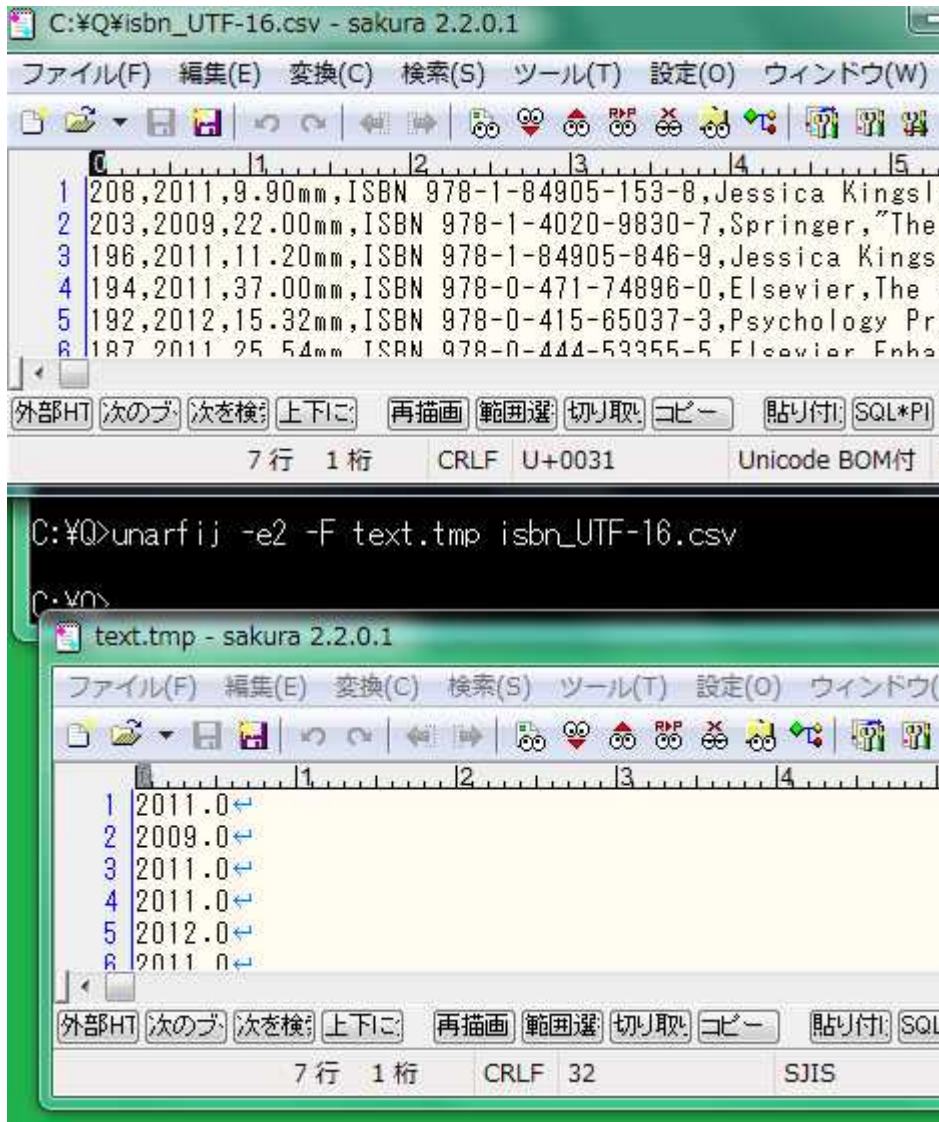
### 空白も文字も無視される

`-e3` が付加された unarfij コマンドは、浮動小数点数のみ抽出します。すなわち、欄の前半にある空白も文字も無視されます。



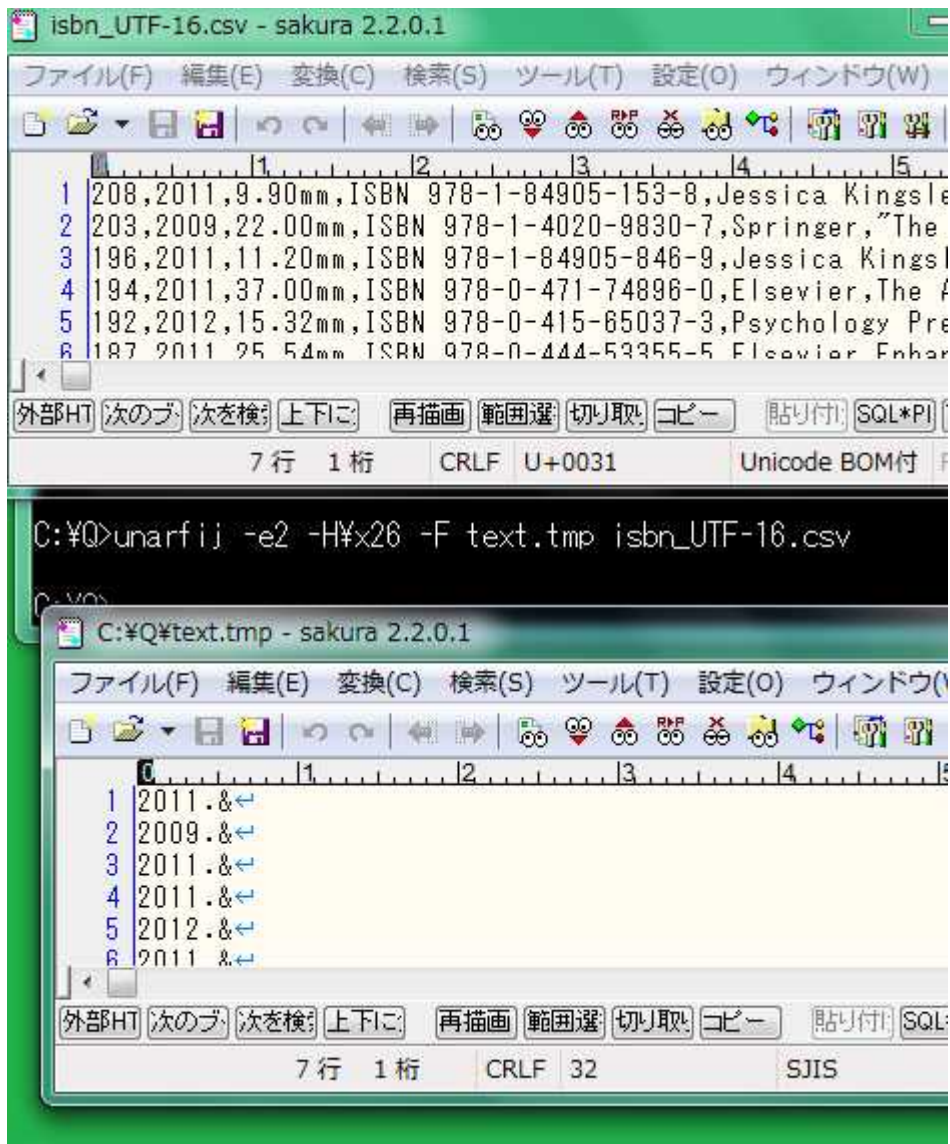
### 小数点が発見されない

利用者が、たとえば第 2 欄から浮動小数点数を抽出するように命令したにもかかわらず、第 2 欄で小数点(ドット)が出現しなかった場合、unarfij コマンドは、自動的にドットおよびゼロを付加します。



### ドットおよび記号

利用者が、たとえば第 2 欄から浮動小数点数を抽出するように命令した、すなわち `-e2` を付加したにもかかわらず、第 2 欄で小数点(ドット)が出現しなかった場合、unarfij コマンドが自動的にドットおよび&を付加するように命令するには、さらに `-H¥x26` を付加してください。十六進数として、26h は、&を意味します。下図の例のように、`-H¥x` 十六進数でドットおよび記号を付加できます。



## 設定ファイル

unarfij コマンドの設定ファイルは、`set_unarfij.txt` です。設定ファイルは、不要ですが、カレントディレクトリで設定ファイルを開くことができない場合、unarfij コマンドは、エラーファイルを書き出します。

### 第 1 行に変換表のパス名を記入してください

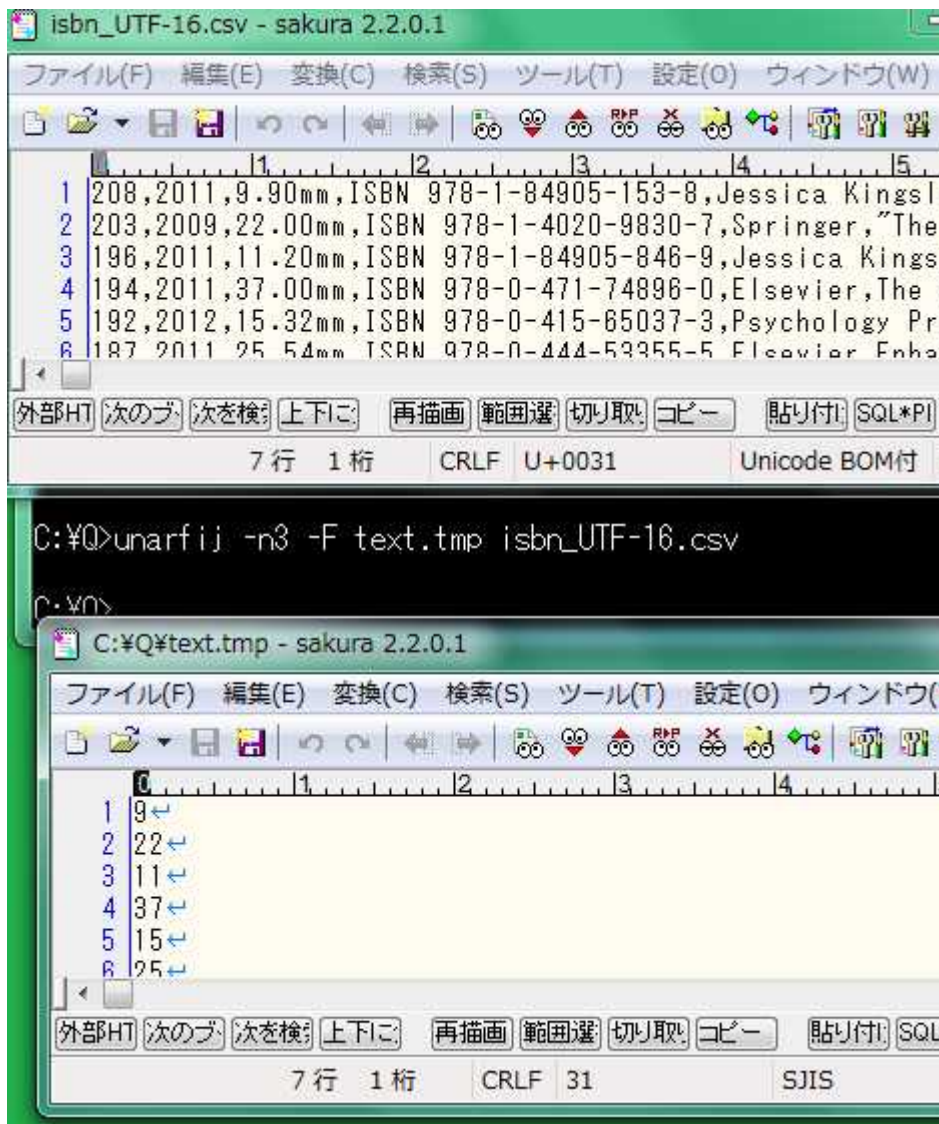
`set_unarfij.txt` の第 1 行に変換表のパス名を記入してください。第 2 行に `-0` を記入してください。必要に応じて、第 3 行に `-M` を記入してください。以下、同様です。コマンドプロンプトでは、空白でパラメータを区切りますが、設定ファイルでは、各行に 1 個のパラメータを記入してください。注釈も記入できます。パラメータ、タブ、注釈の順に記入してください。

## 整数を抽出

第3欄から整数を抽出するには、unarfij コマンドのパラメータとして、`-n3` を付加してください。

### 空白も文字も無視される

`-n3` が付加された unarfij コマンドは、先頭の数字を探索して、その位置を取得します。負号があれば、その位置を取得します。さらに、欄から整数のみ抽出します。すなわち、欄の前半にある空白も文字も無視されます。

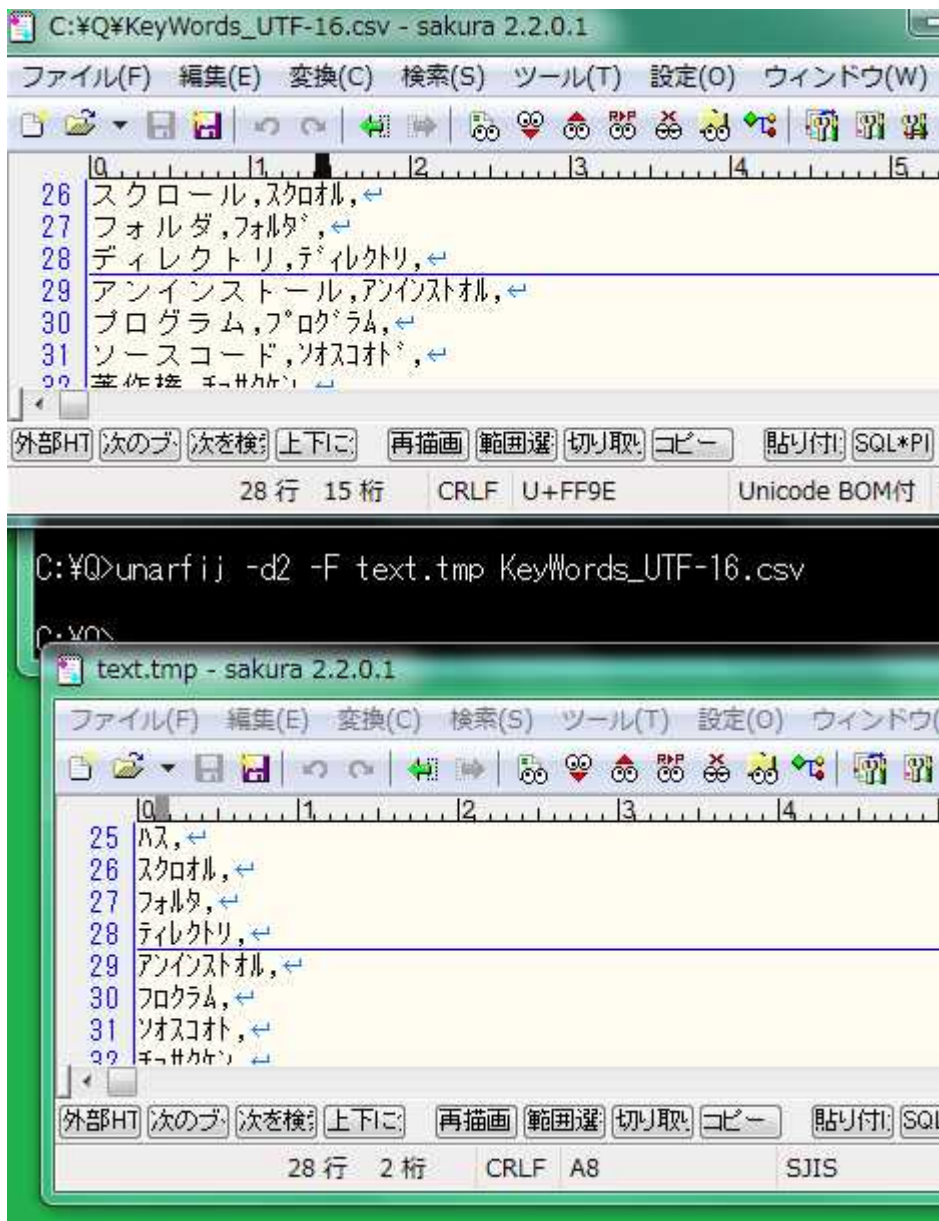


## 濁点を削除

第 2 欄から濁点、半濁点を削除するには、unarfij コマンドのパラメータとして、-d2 を付加してください。

### H および I を削除

半角カタカナの代用として ASCII を使用している場合、さらに、-p を付加してください。ASCII の H および I が削除されます。

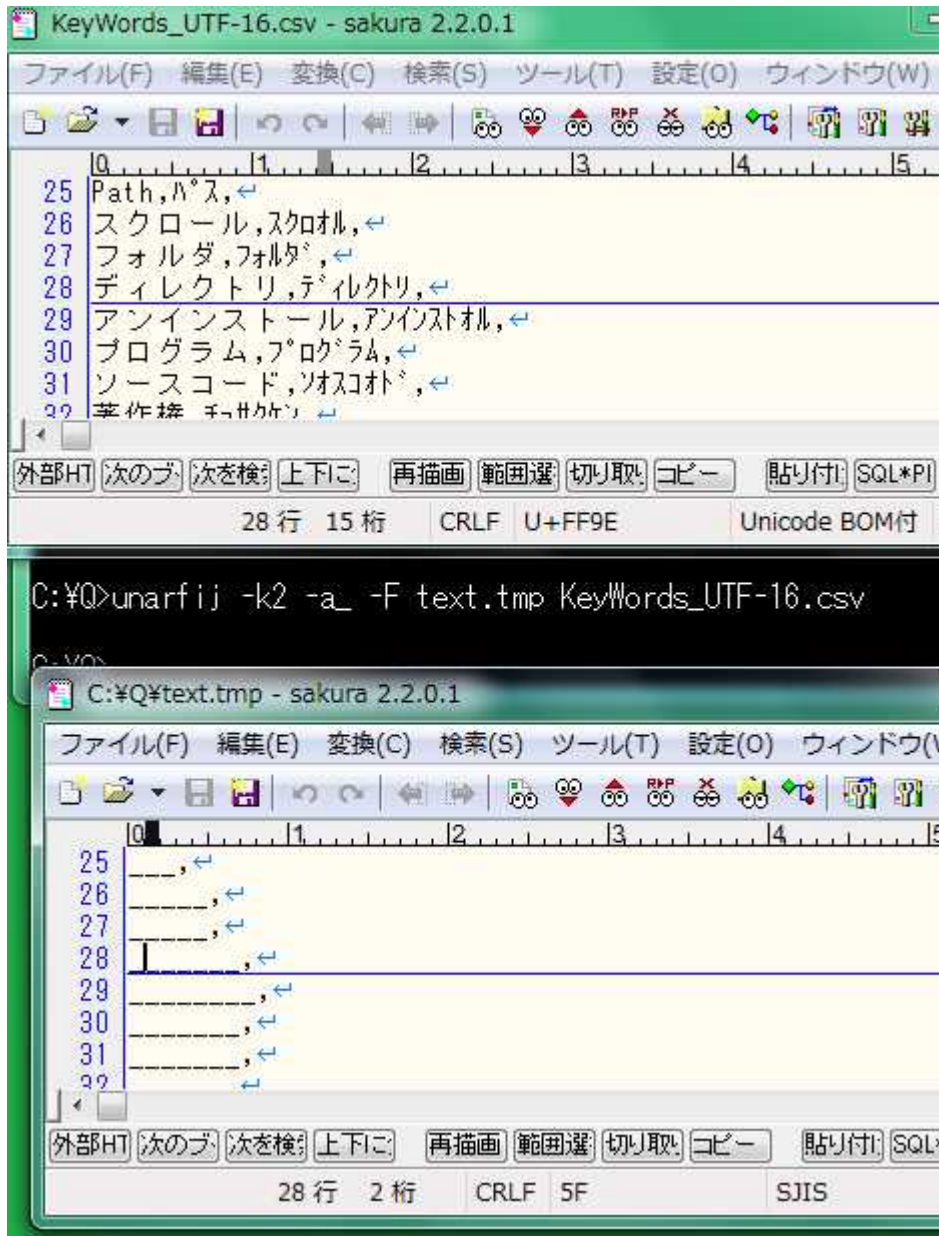


## 半角カタカナを廃止

ある欄で半角カタカナを ASCII に置換できます。

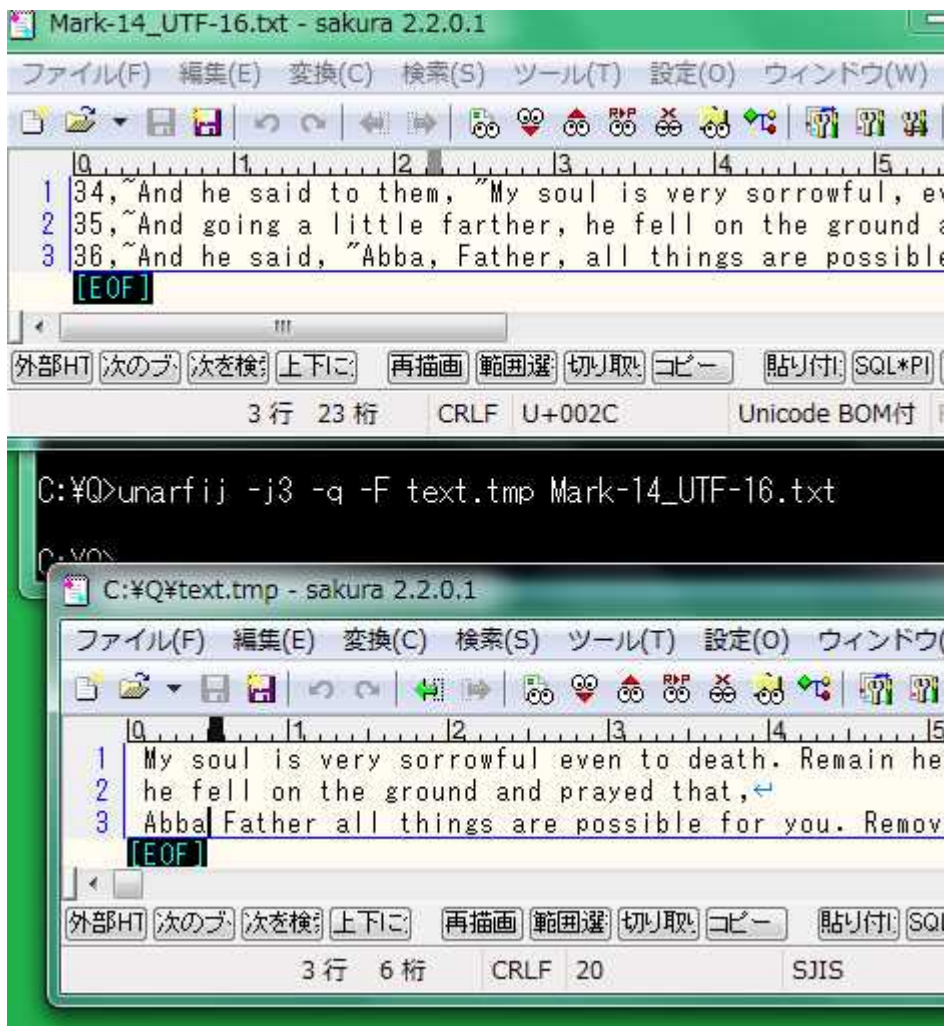
### すべて下線になる

unarfij コマンドは、どの半角カタカナも同一の文字に置換します。第 2 欄で半角カタカナを下線(5Fh)に置換するには、unarfij コマンドのパラメータとして、`-k2 -a_` を付加してください。



## 引用符で囲まれたコンマ

引用符が出現したら、unarfij コマンドは、もうひとつの引用符が出現するまで、区切り文字を無視しますが、無視するだけではなく削除するには、unarfij コマンドのパラメータとして、-q を付加してください。



## 引用符を変更したい

引用符が出現したら、unarfij コマンドは、もうひとつの引用符が出現するまで、区切り文字を無視しますが、自由な表現として、いわゆる引用符( ")が欄内に出現する場合、unarfij コマンドが認識する引用符を他の文字に変更しなければなりません。

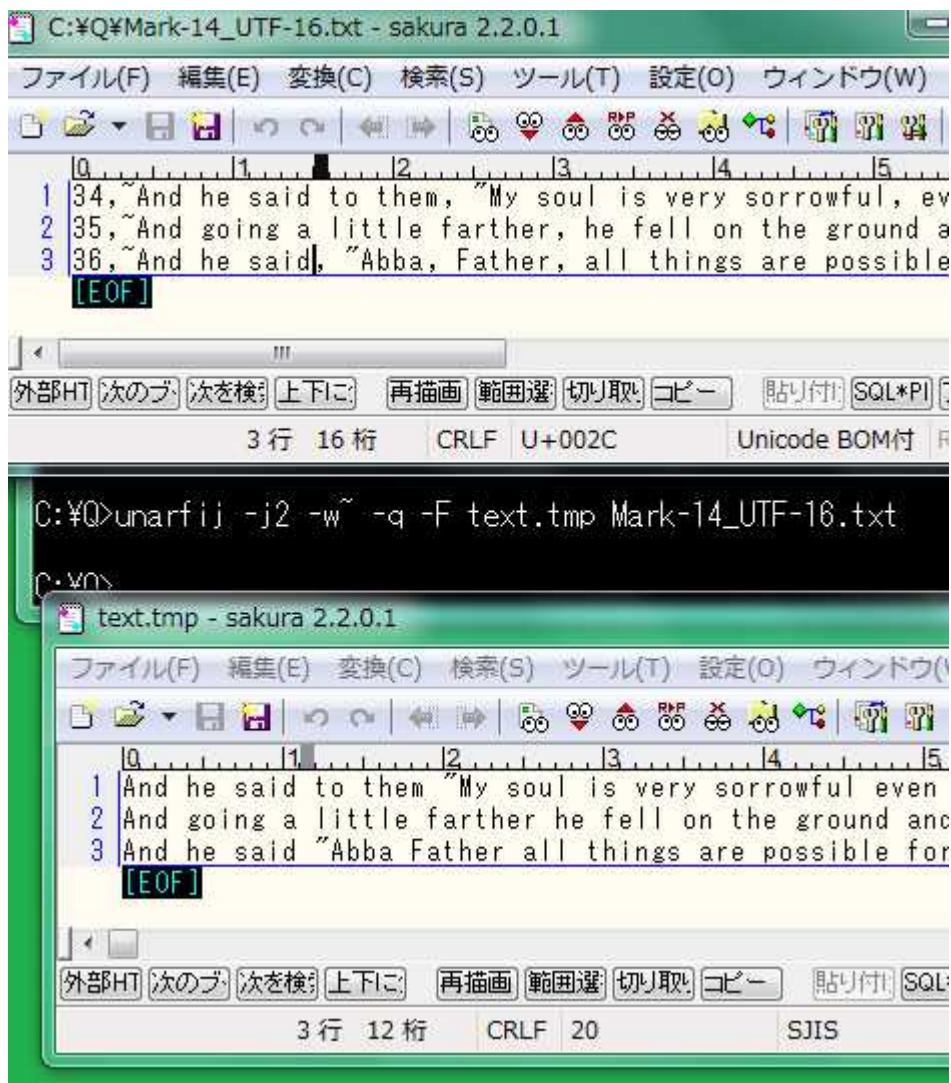
### 引用符を設定

チルダ( ~ )が出現した場合もうひとつのチルダが出現するまで区切り文字を無視するように設定するには、unarfij コマンドのパラメータとして、-w~ を付加してください。unarfij コマンドは、引用符として~を使用します。いわゆる引用符( ")は、もはや引用

符として認識されません。

### 欄からコンマを削除

-w~が付加された unarfij コマンドは、チルダに囲まれたコンマを無視しますが、無視するだけではなく削除するには、さらに-q を付加してください。下記の例では、第 2 欄の全体がチルダで囲まれているため、第 2 欄からコンマが削除されました。



### ファイル名に.csvが含まれているか

unarfij コマンドは、ファイル名に.csvが含まれている場合、区切り文字としてコンマを使用しますが、ファイル名に.csvが含まれているかどうか点検しないように命令するには、unarfij コマンドのパラメータとして、-x-を付加してください。

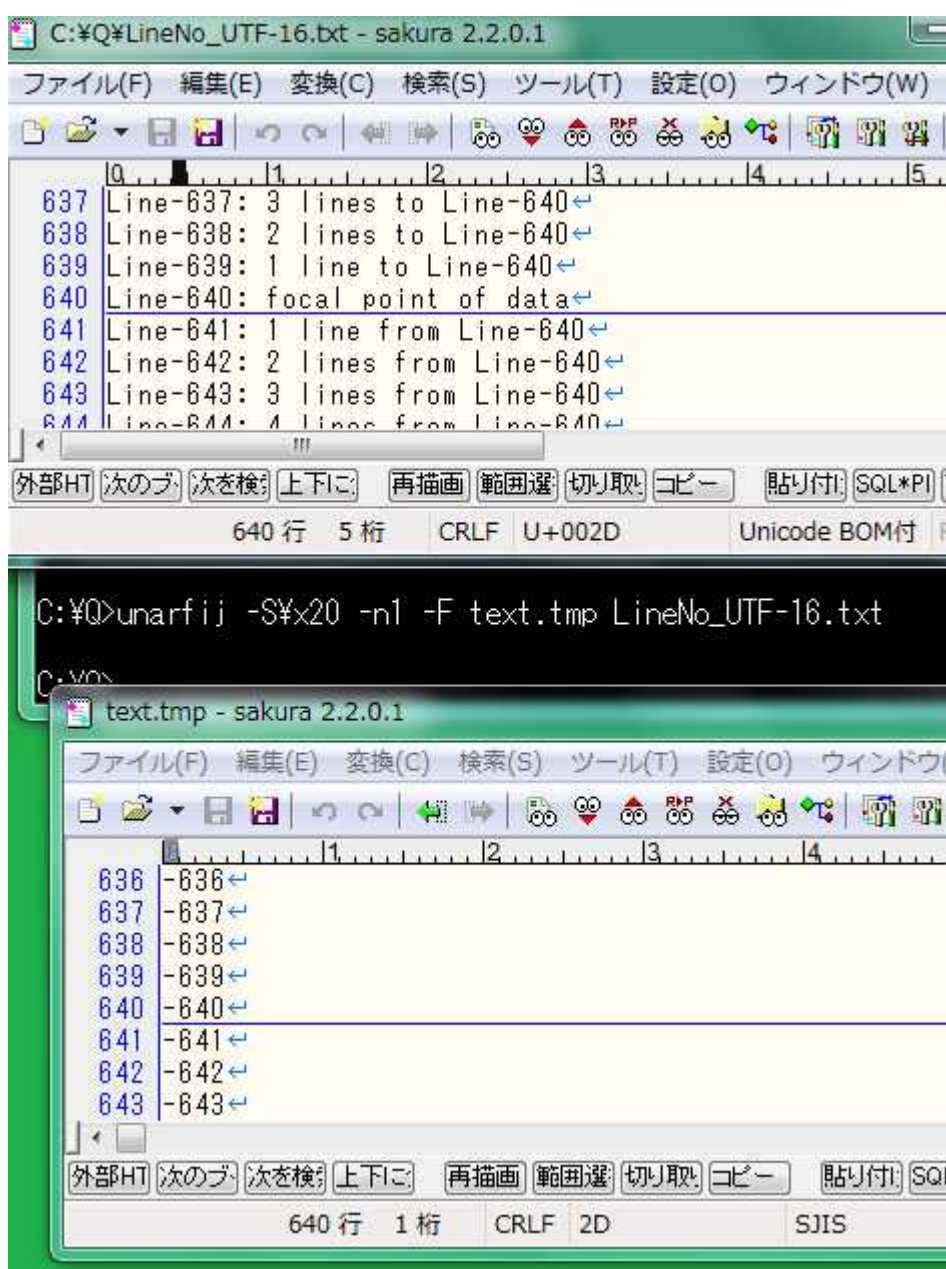
### 区切り文字を変更したい

unarfij コマンドは、CSV の UTF-16 ファイルを想定しています。すなわち、区切り文字

としてコンマを使用しますが、区切り文字として空白(20h)を使用するには、unarfij コマンドのパラメータとして、-S¥x20 を付加してください。-S¥s は、-S¥x20 と同じ結果を生じます。ファイル名に.csv が含まれている場合、さらに、-x-を付加してください。

### 空白

添付の LineNo\_UTF-16.txt の第 1 欄は、行番号が Line-1, Line-2, Line-3 の形式で記入されています。区切り文字は、コンマではなく空白です。第 1 欄から整数のみ抽出するには、unarfij コマンドのパラメータとして、-S¥x20 -n1 を付加してください。





パラメータ	ほぼ同等な入力	1バイト文字	説明
-S¥A	-S¥x3C	<	小なり記号
-S¥^	-S¥x3C	<	小なり記号
-S¥	-S¥x7C		縦線
-S¥s	-S¥x20		空白
-S¥t	-S¥x09		タブ、水平タブ
-S¥v	-S¥x3E	>	大なり記号

### 実体参照

インターネットには、Wiki 形式のウェブサイトがあります。Wiki の表は、区切り記号として縦線を使用します。Wiki では、たとえば、以下の文字列は、円記号( ¥ )に置換されます。こうした文字列を実体参照と言います。

## &yen ;

### Wiki の縦線

添付の equivalents\_UTF-16.txt は、Wiki の表のテキストファイルです。区切り文字として縦線を使用するには、unarjijコマンドのパラメータとして、-S¥| を付加してください。十六進数で表現した-S¥x7C は、-S¥| と同じ結果を生じます。

```
C:\Q>unarjij -S¥| -j2 -F text.tmp equivalents_UTF-16.txt
```

```

1 | パラメータ |
2 | -S&yen;A |
3 | -S&yen;^ |
4 | -S&yen;| |
5 | -S&yen;s |
6 | -S&yen;t |
7 | -S&yen;v |
8 | [EOF]

```

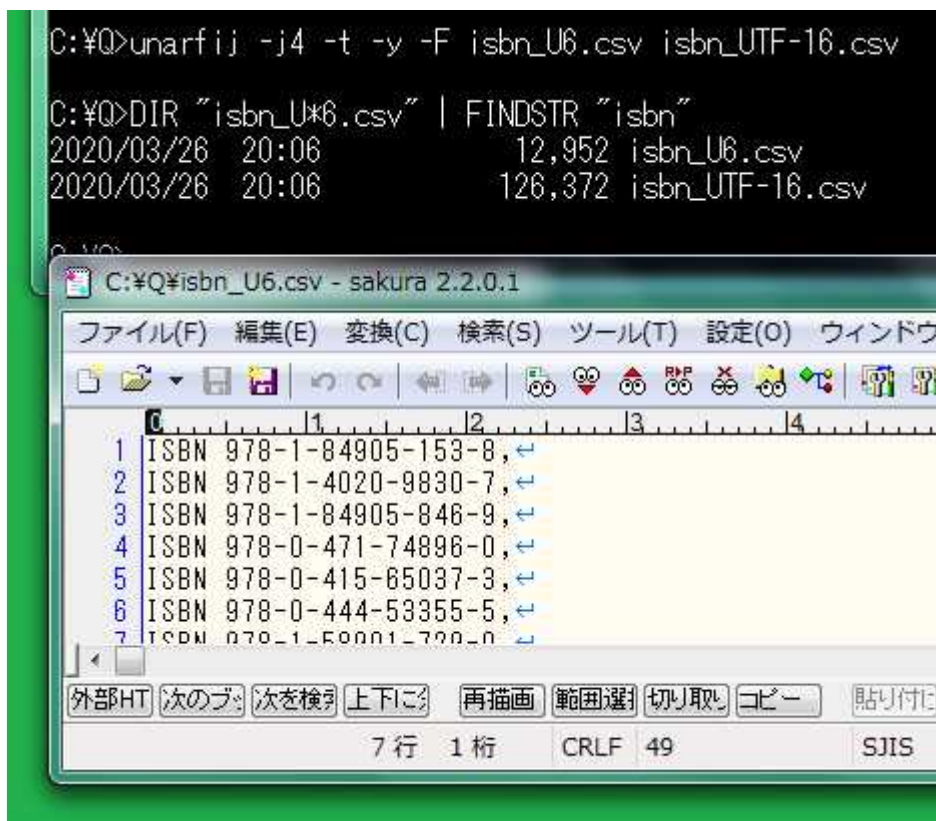
パラメータ	ほぼ同等な入力	1バイト文字	説明
-S¥A	-S¥x3C	<	小なり記号
-S¥^	-S¥x3C	<	小なり記号
-S¥l	-S¥x7C		縦線
-S¥s	-S¥x20		空白
-S¥t	-S¥x09		タブ、水平タブ
-S¥v	-S¥x3E	>	大なり記号

## ファイルの日時

元のファイルが新しいファイルと同じ日時を有するように命令するには、unarfij コマンドのパラメータとして、-t を付加してください。

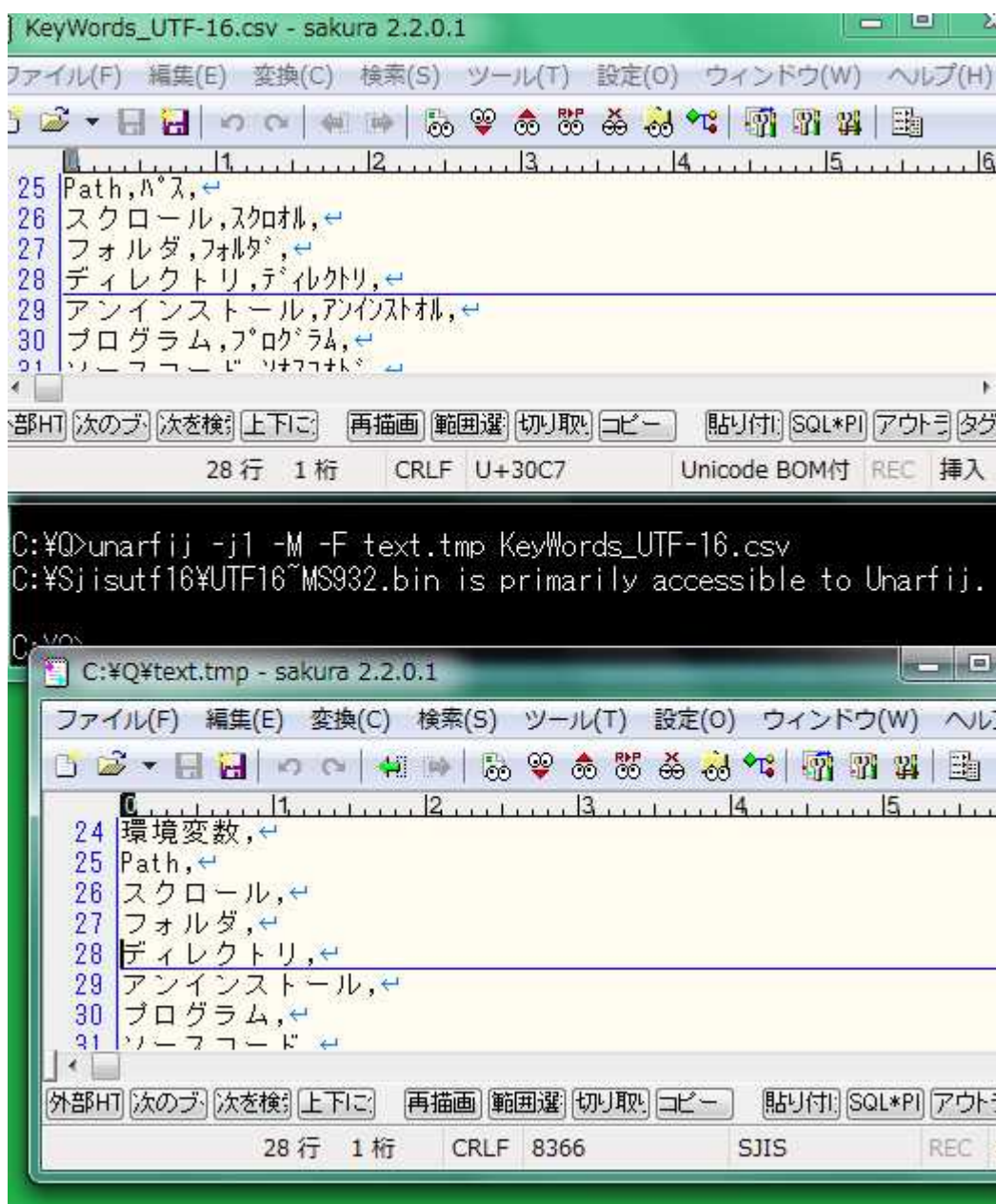
### 一時ファイルを削除しない

-t 付きの unarfij コマンドは、一時ファイルとして binary.tmp を作成します。unarfij コマンドが自動的に一時ファイルを削除しないように命令するには、unarfij コマンドのパラメータとして、-t -y を付加してください。



## どの変換表を使用するか

surro.csv および UTF16~MS932.bin にしたがって、unarfij コマンドは、UTF-16 のファイルをシフト JIS のテキストファイルに変換します。どの変換表を使用するか表示するには、unarfij コマンドのパラメータとして、-M を付加してください。画面に表示された内容は、data.txt に保存されています。data.txt は、シフト JIS のテキストファイルです。



## 変換表を探索する順序

unarfig コマンドは、下記の順序で変換表を探索します。

1. unarfig コマンドがインストールされたディレクトリ
2. カレントディレクトリ
3. 環境変数 Path に記載の各ディレクトリ
4. 設定ファイルの第 1 行に記載のパス名

### 設定ファイルに記載のパス名を優先

インストールディレクトリ、カレントディレクトリ、Path に記載の各ディレクトリに変換表があろうがなかろうが、設定ファイルの第 1 行からパス名を捕捉して、変換表を開くには、unarfig コマンドのパラメータとして、`-K` を付加してください。

### 探索しない

変換表の探索を抑制するには、unarfig コマンドのパラメータとして、`-D` を付加してください。

## 欄の数の上限

欄数の上限は、255 欄です。

### 上限を変更

255 から 3071 に欄数の上限を変更するには、unarfig コマンドのパラメータとして、`-Z` を付加してください。`Z` は、大文字です。

## --out と同等なパラメータ

`--out` は、`-F-` `-z` と同じ結果を生じます。

パラメータ	新しいファイル	長さ	説明
<code>-F text.tmp</code>	text.tmp	8	利用者は、自分で、新しいファイルを text.tmp と命名した。
<code>-F-</code>	unarfig.txt	11	利用者は、自分で新しいファイルを命名しない。
<code>-F- -z</code>	空文字	0	利用者は、新しいファイルを空文字に設定した。utf16sjis コマンドは、標準出力に変換結果を書き出す。

### 標準出力に書き出す例

isbn\_UTF-16.csv の第 4 欄のみ選択して、標準出力にシフト JIS のデータを書き出すには、コマンドプロンプトに下記の j コマンドおよびパラメータを入力して、Enter を押してください。

```
unarfig -j4 --out isbn_UTF-16.csv
```

```

C:\Windows\system32\cmd.exe
ISBN 978-0-19-975456-4,
ISBN 978-1-4419-3734-6,
ISBN 0-19-506442-9,
ISBN 978-0-262-01523-3,
ISBN 978-3-0346-0225-9,
ISBN 978-0-19-979431-7,
ISBN 978-1-4614-0958-8,
ISBN 0-387-36512-5,
ISBN 978-1-118-73282-3,
ISBN 978-1-4488-9552-6,
ISBN 978-3-7091-0722-5,
ISBN 978-0-19-929675-0,
ISBN 978-0-19-974431-2,
ISBN 978-0-307-27208-9,
ISBN 978-1-60021-504-9,
ISBN 0-387-31415-6,
ISBN 978-1-4614-3902-8,
ISBN 978-0-7377-5254-0,
ISBN 978-0-262-51311-1,
ISBN 0-19-853768-9,
ISBN 978-0-444-63425-2,
C:\>

```

**参考**

<b>.zip</b>	<b>コンパイラ</b>	<b>並べ替え</b>	<b>文字コード</b>	<b>説明</b>
Atnd	FPC	sortk	EUC-JP	Delphi 4 版の Narfi を添付しました。
Sjisutf16	FPC		UTF-16	
NarfiDelphi	Delphi 2009	sortk	EUC-JP	
Utf16sjis	Delphi 2009	usortk	UTF-16	

Free Pascal Compiler 3.0.4(FPC)