

# Sortk ユーザーズガイド

sortk コマンドは、テキストファイルの行を並べ替えて、標準出力に書き出します。



## CSV

拡張子が.csv であるテキストファイルは、各行が、コンマで分離された多数の語句からなります。CSV のファイルは、Excel などの表計算ソフトウェアで容易に表示できます。sortk コマンドは、CSV の欄にある語句を比較して、行を並べ替えます。

## 区切り文字

CSV の行を並べ替えるには、コマンドプロンプトに sortk コマンドのパラメータとして、`-t`(マイナステーコンマ)を付加してください。`-t` およびコンマの間に空白を挿入しないでください。

### 空白

区切り文字が指定されていない場合、sortk コマンドは、区切り文字として空白を使用します。

## 標準入力

CSVファイル名が指定されていない場合、sortkコマンドは、標準入力から読み込みます。

## 新しいファイル

下記のとおり、標準出力からファイルにリダイレクトすることで、新しいファイルを作成できます。

```
sortk -t, < csv > text.tmp
```

### 命名

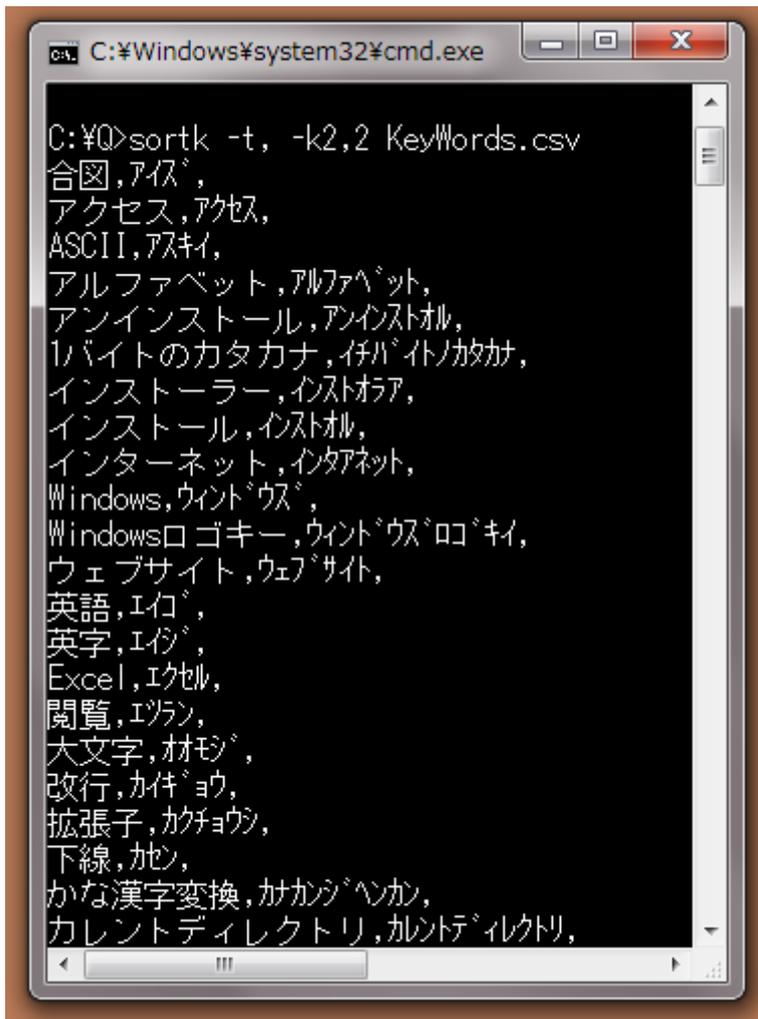
csv を読み込んで、新しいファイルの作成および命名を行うには、コマンドプロンプトに sortk コマンドのパラメータとして、`-o text.tmp` を付加してください。`-o` および `text.tmp` の間に空白を挿入してください。`text.tmp` は、新しいファイルの名前です。

```
sortk -t, -o text.tmp csv
```

## 第 2 欄を比較

sortk コマンドは、たとえば、第 2 欄にある語句を比較して、CSV のテキストファイルの行を並べ替えることができます。第 2 欄にある語句を比較するには、sortk コマンドのパラメータとして、`-k2,2` を付加してください。`-k` および数字の間に空白を挿入しないでください。

# sortk -t, -k2,2 KeyWords.csv



## sortk の由来

sortk (ソーティーケー) という名前は、UNIX の `sort -t, -k` に由来します。FreeBSD, Linux, Solaris などの UNIX は、`sort` コマンドを用意しています。sortk は、UNIX の `sort` コマンドに似ています。

## 日本語

sortk コマンドは、半角カタカナの濁点、半濁点を無視できます。Windows のコマンドプロンプトで、EUC-JP のテキストファイルを並べ替えることができます。0Ah のみで改行できます。

# インストール

sortk 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 Windows 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-CS\\_zip.html](http://tori.tobihiro.jp/PDF-CS_zip.html)) というウェブサイトから sortkCS.zip をダウンロードしてください。
2. sortkCS.zip を展開してください。
3. Cドライブに sortkCS フォルダを移動してください。
4. 利用者は、sortkCS フォルダの移動および改名が可能です。
5. Windows ログキー(田キー)を押しながら Pause を押してください。
6. メニューからシステムの詳細設定を選択してください。
7. 詳細設定タブをクリックしてください。
8. [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**
9. 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。
10. Path をクリックして選択してください。
11. [編集]ボタンを押してください。
12. Windows 10 の場合、[テキストの編集]ボタンを押してください。
13. 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。
14. 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:¥sortkCS を入力してください。
15. 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、sortk フォルダごと削除してください。Path から C:¥sortkCS を削除してください。

## 著作権

Sogaya(そがや)は、sortk の著作権を保有しています。ソースコードを添付しました。

## 免責条項

sortk 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

Visual C# 2008 Express で構築しました。`.NET Framework 3.5` を使用します。

## ユーザーズガイド

2019 年 7 月 27 日 Revision 1.044-part-1

# sortk コマンドのパラメータ

sortk コマンドは、CSV の欄にある語句、整数、浮動小数点数を比較して、行を並べ替えます。利用者は、CSV の欄を選択できます。

## 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(①②③⑩⑰)、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。

### CHCP コマンド

コマンドプロンプトの文字コードがシフト JIS に設定されていることを確認するには、パラメータを付加しないで、コマンドプロンプトに CHCP を入力して、Enter を押してください。932 が表示されます。MS932 が使用されることがわかります。どの Windows も最初から CHCP コマンドを用意しています。MS932 は、CP932 とともに Windows-31J とも言います。



```
C:\>CHCP
現在のコード ページ: 932
C:\>
```

## 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。

## atnd コマンド

添付の atnd は、シフト JIS 用のコンソールアプリケーションです。シフト JIS のファイルの最初から 7 行を表示するには、コマンドプロンプトに、たとえば、下記の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。-h の直後に 7 を入力してください。-h7 は、-b6 -p7 を意味します。

```
C:\¥Q>atnd -h8 Gongitsune.txt
こん狐
新美南吉
—

これは、私《わたし》が小さいときに、村の茂平《もへい》というおじいさんからきいたお話です。
むかしは、私たちの村のちかくの、中山《なかやま》というところに小さなお城があって、中山さまというおとのさまが、おられたそうです。
その中山から、少しはなれた山の中に、「こん狐《ぎつね》」という狐がいました。ごんは、一人《ひとり》ほっちの小狐で、しだの一ぱいしげった森の中に穴をほって住んでいました。そして、夜でも昼でも、あたりの村へ出てきて、いたずらばかりしました。はたけへ入って芋をほりちらしたり、菜種《なたね》がらの、ほしてあるのへ火をつけたり、百姓家《ひやくしょうや》の裏手につるしてあるとんがらしをむしりとり、いったり、いろんなことをしました。
C:\¥Q>
```

## FC コマンド

テキストファイルを比較するために、FC コマンドを試すことができます。どの Windows も最初から FC コマンドを用意しています。

```
C:\¥Q>sortk -t, -k2,2 -o text.tmp KeyWords.csv
C:\¥Q>sortk -t, -k2,2 KeyWords.csv > new.csv
C:\¥Q>FC text.tmp new.csv
ファイル text.tmp と NEW.CSV 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>
```

## バージョン番号

sortk コマンドのバージョン番号を確認するには、-V を付加してください。

### sortk -V

#### 短い説明

パラメータに関する短い説明を表示するには、--help を付加してください。画面からテキストファイルに、リダイレクトするには、-s を前置してください。

### sortk -s --help > help.txt

```
C:\¥Q>sortk -s --help > help.txt
```

```
C:\¥Q>NOTEPAD help.txt
```

help - メモ帳

ファイル(F) 編集(E) 書式(O) 表示(V) ヘルプ(H)

sortk C# Version 1.03

```
sortk -t, -o new csv          # Read csv to produce a new file.
sortk -t, -L -o new csv      # 0Dh is removed from every line.
sortk -t, < csv > new        # Read csv to produce a new file.
sortk -t, -a4 -h1011 csv     # Compare in Field-1, Field-3 and Field-4.
sortk -t, -j2 csv            # Compare phrases in Field-2.
sortk -t, -k2,2 csv          # Compare phrases in Field-2.
sortk -t, -k2,3 csv          # Compare phrases from Field-2 to Field-3.
sortk -t, -k2 csv            # Compare phrases from Field-2 to finish.
sortk -t, -k1 -$ csv         # Check if all lines share the field count.
sortk -t, -k2,2 -# csv       # Add ID to each summary.
sortk -t, -k2,3 -# -! csv    # Report ID if any summaries match each other.
sortk -t, -k2,2 -% csv      # Produce manager.txt
sortk -t, -k2,2 -8 csv      # UTF-8
sortk -t, -k2,2 -A- csv     # Do not replace "&" with &.
sortk -t, -k2,2 -B csv      # Replace ">" with >.
sortk -t, -k2,2 -b csv      # Skip space in the first part of Field-2.
sortk -t, -k2,2 -c- csv     # Skip combining characters.
sortk -t, -k2,2 --cope csv  # --cope means -c- -p -e -o.
sortk -t, -k2,2 --pic csv   # --pic means -c- -p -i.
sortk -t, -k2,2 -e csv      # EUC-JP
sortk -t, -k2,2 -f csv      # EIGHT equals eight.
sortk -t, -k2,2 -g csv      # Compare 0.12 with 0.78 in Field-2.
sortk -t, -k2,2 -i csv      # Skip control characters.
sortk -t, -k2,2 -N csv      # Replace "&#xC;" with a form-feed character.
sortk -t, -k2,2 -n csv      # Compare 3456 with 9012 in Field-2.
sortk -t, -k2,2 -p csv      # H and I are omitted from ASCII characters.
sortk -t, -k2,2 -q csv      # Commas disappear if placed in quotes.
sortk -t, -k2,2 -r csv      # Arrange in descending order.
sortk -t, -k2,2 -S csv      # Strictly check </hetn>.
sortk -t, -k2,2 -T csv      # Count ticks.
sortk -t, -k2,2 -u csv      # Display a line for each unique summary.
sortk -t, -k2,2 -y csv      # Do not remove temporary files.
sortk -t¥1 -k2,2 file        # Wiki table
sortk -t¥1 -k2,2 -x- csv     # Do not check if the file name includes .csv
sortk -t¥x7C -k2,2 file      # Wiki table
sortk -U131072 -k2,2 csv     # The number of lines shall not exceed 131072.
sortk -v¥x1F -k2,3 csv      # 1Fh connects phrases from Field-2 to Field-3.
sortk -w~ -k2,4 csv         # ~ is used as a quote.
sortk -w~ -k2,4 csv         # ~ is used as a quote.
```

パラメータ	ほぼ同等な入力	説明
-!		要約が合致した場合、ID を表示する。
-\$		どの行も同じ欄数を有しているか点検する。
-#		要約に ID を付与する。
-%		設定ファイルとして、manager.txt を作成する。
-8		UTF-8 のテキストファイルを読み込む。
-a4	-h1111000000 -k1,4	第 1 欄から第 4 欄まで選択する。
-b		比較開始欄の前半にある空白を無視する。
-c-		半角カタカナの濁点、半濁点を無視する。
-e		EUC-JP のテキストファイルを読み込む。
-f		大文字と小文字を区別しないで比較する。
-g		浮動小数点数を比較する。
-h1011		第 2 欄を除外する。
--help	-VC	パラメータの短い説明を表示する。
-i		タブ、改ページなど制御文字を無視する。
-j2	-k2,2	第 2 欄のみ選択する。
-k2		第 2 欄から最終欄まで選択する。
-k2,3	-a3 -h011	第 2 欄から第 3 欄まで選択する。
-L	-LF	0Ah のみで改行する。
-n		整数を比較する。
-o new	> new	新しいファイルを命名する。
-p		濁点の代用文字として、H を無視する。
-q		引用符で囲まれた区切り文字を削除する。
-r		降順で並べ替える。
-s		標準エラー出力ではなく標準出力に書き出す。
-t,	.csv の拡張子	区切り文字としてコンマを使用する。
-t¥1	-t¥x7C	Wiki の表の区切り記号として縦線を使用する。
-U131072		行数は、131072 を超えてはならない。
-u		要約ごとに 1 行のみ表示する。
-v¥x1F		選択された語句を 1Fh で連結する。
-V		バージョン番号を表示する。
-w~	-w¥x7E	~ が引用符として利用される。
-x-		.csv がファイル名に含まれるか点検しない。
-y		一時ファイルを削除してはならない。

## 参考 1

C++版 sortk コマンドがございます。ロトサマリーまたはトリシーカーのウェブページとして <http://tori.tobihiro.jp/PDF-zip-7z.html> から sjisEUCjpCxx.7z をダウンロードできます。7-Zip のソフトウェアで展開してください。C 版 sortk コマンドがございます。ロトサマリーまたはトリシーカーのウェブページとして <http://tori.tobihiro.jp/PDF-zip-C.html> から sjisEUC-C.zip をダウンロードできます。

### シフト JIS のテキストファイル

シフト JIS のテキストファイルを読み込んで、制御文字を無視したい場合、sortk コマンドのパラメータとして、`-i` を付加してください。この場合、一時ファイルを出力しないで行を並べ替えます。C#版 sortk コマンドの一時ファイルは、XML です。

### 経過時間

C#の Stopwatch クラスを利用して、パルスをカウントして経過時間を表現するには、sortk コマンドのパラメータとして、`-T` を付加してください。時間の単位は、ミリ秒ではなく、ticks です。`--tick` は、`-T` を同じ結果を生じます。

### XML ファイル

C++版も C 版も sortk コマンドは、バイナリファイルとして要約および詳細を出力します。C++版と異なり、C#版の sortk コマンドは、XML ファイルとして要約および詳細を出力します。ファイルの名前は、`sutai.xml` です。添付の `hetn` コマンドは、`sutai.xml` を再利用できます。

パラメータ	ほぼ同等な入力	説明
<code>-A-</code>		実体参照を&に置換してはならない。
<code>-B</code>		実体参照を<に置換する。実体参照を>に置換する。
<code>-N</code>		<code>&amp;#xHH;</code> および <code>&amp;#xH;</code> を数に変換する。H は、十六進数(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)
<code>-S</code>		<code>&lt;/hetn&gt;</code> を厳密に点検する。
<code>-T</code>	<code>--tick</code>	パルスをカウントして経過時間を表現する。

### ¥x および十六進数

上記のパラメータのうち、`-t`, `-v`, `-w` は、十六進数で 1 バイト文字を指定できます。たとえば、`-w¥x7E` は、`-w~` と同じ結果を生じます。

### ¥v で大なり記号を意味する

AWK, C#などのプログラミング言語では、¥v は、垂直タブ(0Bh)を意味する場合がありますが、sortk コマンドのパラメータでは、大なり記号(>)を意味します。すなわち、`-t¥v` は、`-t¥x3E` と同じ結果を生じます。下表では、`-t` の例を示します。`-v`, `-w` の表も同様になります。

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

```

C:¥Q>TYPE equivalent.txt
|~パラメータ|~ほぼ同等な入力|~1バイト文字|~説明| |
|-t&yen;A|-t&yen;x3C|&lt;|小なり記号|
|-t&yen;^|-t&yen;x3C|&lt;|小なり記号|
|-t&yen;| |-t&yen;x7C|&#124;|縦線|
|-t&yen;s|-t&yen;x20| |空白|
|-t&yen;t|-t&yen;x09| |タブ、水平タブ|
|-t&yen;v|-t&yen;x3E|&gt;|大なり記号|

C:¥Q>sortk -t%| -k2,2 equivalent.txt
|~パラメータ|~ほぼ同等な入力|~1バイト文字|~説明| |
|-t&yen;A|-t&yen;x3C|&lt;|小なり記号|
|-t&yen;| |-t&yen;x7C|&#124;|縦線|
|-t&yen;s|-t&yen;x20| |空白|
|-t&yen;t|-t&yen;x09| |タブ、水平タブ|
|-t&yen;v|-t&yen;x3E|&gt;|大なり記号|
|-t&yen;^|-t&yen;x3C|&lt;|小なり記号|

C:¥Q>sortk -t%| -k2,2 -B equivalent.txt
|~パラメータ|~ほぼ同等な入力|~1バイト文字|~説明| |
|-t&yen;A|-t&yen;x3C|&lt;|小なり記号|
|-t&yen;| |-t&yen;x7C|&#124;|縦線|
|-t&yen;s|-t&yen;x20| |空白|
|-t&yen;t|-t&yen;x09| |タブ、水平タブ|
|-t&yen;v|-t&yen;x3E|&gt;|大なり記号|
|-t&yen;^|-t&yen;x3C|&lt;|小なり記号|

```

**¥| は縦線を意味する**

Wiki の表は、区切り記号として縦線を使用します。上記の `equivalent.txt` は、Wiki の表のテキストファイルです。第 1 欄および第 6 欄は、空です。第 2 欄のみ比較し

て、並べ替えました。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;

実体参照	記号	パラメータ	説明
&amp;	&	-A-	&amp;を&に置換してはならない。
&copy;	©		
&gt;	>	-B	&gt;を>に置換する。
&lt;	<	-B	&lt;を<に置換する。
&quot;	"		
&yen;	¥		

## パラメータからコンマを排除

sortk コマンドのパラメータとして、-t,(マイナスティーコンマ)も-k2,3(マイナスケーニコマサン)もコンマが含まれていますが、コンマの使用を回避できます。

### 範囲を表現するハイフン

コンマではなくハイフンを使用できます。-k2,3ではなく-k2-3が指定されても、sortk コマンドは、第2欄から第3欄までにある語句を比較します。

### 十六進数で入力

コンマの十六進数は、2Chです。-t,ではなく-t¥x2Cが指定されても、sortk コマンドは、区切り文字としてコンマを使用します。

### .csv の拡張子

何も指定されなければ、sortk コマンドは、区切り文字として空白(20h)を利用します。行が並べ替えられるテキストファイルの拡張子が.csvである場合、sortk コマンドは、、自動的に、区切り文字としてコンマを使用します。拡張子の点検を抑制するには、sortk コマンドのパラメータとして、-x-を付加してください。

```
C:\¥Q>sortk -t, -k10,20 -o text.tmp Field-20.csv
C:\¥Q>sortk -t¥x2C -k10-20 -o yen_x2C.csv Field-20.csv
C:\¥Q>sortk -k10-20 -o extension.csv Field-20.csv
C:\¥Q>FC text.tmp yen_x2C.csv
ファイル text.tmp と YEN_X2C.CSV 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 text.tmp extension.csv
ファイル text.tmp と EXTENSION.CSV 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>NOTEPAD extension.csv
C:\¥Q>
```

## 要約および詳細のブロック

利用者は、多数の欄を選択できます。sortk コマンドは、CSV の表を読み込んで、利用者選択の欄で要約を構成して、要約に詳細を結合します。要約のみ比較しながら要約および詳細のブロックを並べ替えます。詳細のみ標準出力に書き出します。

### 要約を構成

たとえば、21 欄もある CSV で、まず、第 1 欄から第 5 欄までを選択するには、sortk コマンドのパラメータとして、-a5 を付加してください。次に、第 2 欄を選択から除外する

には、sortk コマンドのパラメータとして、-h10111 を付加してください。欄を選択することの真偽を示すために、1 または 0 を入力してください。10111 のうち 0 は、第 2 欄を選択しないことを意味します。結果として、4 欄で要約を構成します。ただし、-h10111 が指定されない場合、sortk コマンドは、-h11111 が指定されたとみなします。

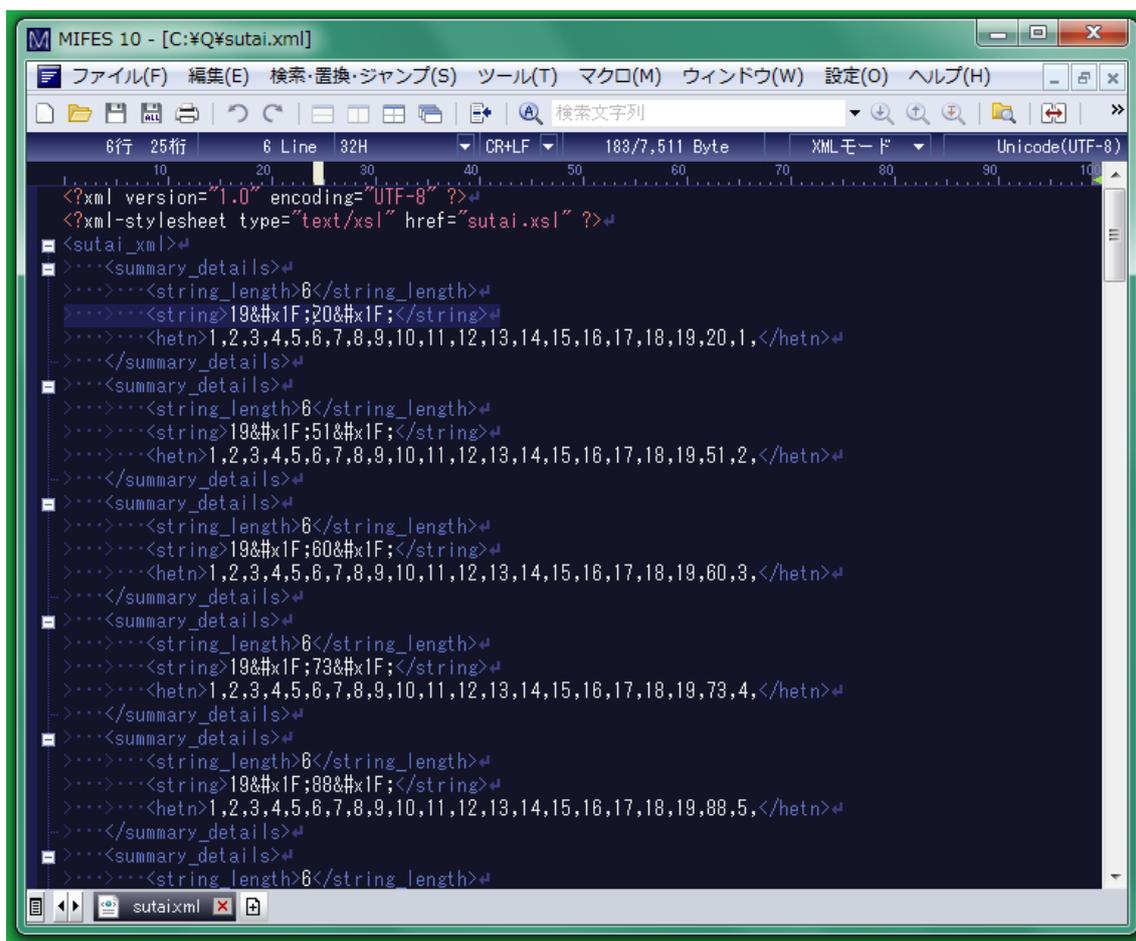
<b>-h10111</b>					
-h	1	0	1	1	1
<b>真偽</b>	第 1 欄選択	第 2 欄を除外する	第 3 欄選択	第 4 欄選択	第 5 欄選択

### 語句の連結

利用者は、多数の欄を選択できます。利用者選択の欄で要約を構成するとき、語句を連結する文字として、感嘆符(21h)が使用されます。語句連結文字は、sortk の内部で要約に使用されますが、詳細として書き出されません。二桁の十六進数で語句連結文字を設定できます。

### 十六進数で入力

二桁の十六進数で語句連結文字を設定できます。選択された語句を 1Fh で連結するには、sortk コマンドのパラメータとして、-v#x1F を付加してください。MIFES で sutai.xml を表示した例



## 2 桁の十六進数は 1 バイトの数に置換される

2 桁の十六進数が sutai.xml に含まれている場合、sortk コマンドも hetn コマンドも数に置換できます。たとえば、-v¥x1F を付加した sortk コマンドは、自動的に 2 桁の十六進数を 1 バイトの数に置換します。

1. 利用者が -v¥x1F を付加した sortk コマンドを入力した。
2. sortk コマンドは、利用者選択の欄を 1Fh で連結して要約を構成した。
3. C# の XmlDocument クラスが 1Fh を &#x1F; に置換した。
4. sortk コマンドは、要約および詳細の多数のブロックとして、sutai.xml を書き出した。
5. sortk コマンドまたは hetn コマンドが、sutai.xml を読み込んだ。
6. sortk コマンドも hetn コマンドも &#x1F; を 1Fh に置換した。
7. sortk コマンドも hetn コマンドも最終的には、1Fh を空白 (20h) に置換した。
8. 本来の空白は、感嘆符 (!) に置換された。

### 1Fh に置換したことを確認

感嘆符 (21h) ではなく、たとえば 1Fh を使用するには、sortk コマンドのパラメータとして、-v¥x1F を付加してください。下記の例では、-v¥x1F のパラメータを付加した sortk コマンドを実行しました。intermediate-3.tmp という一時ファイルを削除しないように、-y も付加しました。上記の第 6 の工程の結果として、intermediate-3.tmp を表示してサクラエディタのカーソルを連結文字に移動すれば、ステータスバーに U+001F UTF-8 が表示されます。すなわち、sortk コマンドが &#x1F; を 1Fh に置換したことを確認できます。図を[参照](#)してください。

```
</string_length><string>Kyoto□Kyoto
```

### obot コマンド

添付の obot コマンドで上記の付近を表示してみました。コマンドプロンプトをドラッグして反転表示した部分を MIFES でも反転表示しました。この方法でも、sortk コマンドが &#x1F; を 1Fh に置換したことを確認できます。このユーザーズガイドにおける □ は、連結文字を意味します。MIFES では、□ は、^\_ と表示されています。

```
>Kyoto□Kyoto□Japan
```

十六進数	文字		十六進数	文字		十六進数	文字
K	4B		J	4A		N	4E
y	79		a	61		r	72
o	6F		p	70		>	3E
t	74		n	6E		<	3C

```

C:¥Q>sortk -t, -k1,3 -v¥x1F -y cities.csv
Kyoto,Kyoto,Japan
Kyoto City,Kyoto Prefecture,Japan
Nara,Nara,Japan
Nara City,Nara Prefecture,Japan
Osaka,Osaka,Japan
Osaka City,Osaka Prefecture,Japan

C:¥Q>obot intermediate-3.tmp
EF BB BF 3C 3F 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22 31 2E
30 22 20 65 6E 63 6F 64 69 6E 67 3D 22 55 54 46 2D 38 22 20
3F 3E 3C 3F 78 6D 6C 2D 73 74 79 6C 65 73 68 65 65 74 20 74
79 70 65 3D 22 74 65 78 74 2F 78 73 6C 22 20 68 72 65 66 3D
22 73 75 74 61 69 2E 78 73 6C 22 20 3F 3E 3C 73 75 74 61 69
5F 78 6D 6C 3E 3C 73 75 6D 6D 61 72 79 5F 64 65 74 61 69 6C
73 3E 3C 73 74 72 69 6E 67 5F 6C 65 6E 67 74 68 3E 31 37 3C
2F 73 74 72 69 6E 67 5F 6C 65 6E 67 74 68 3E 3C 73 74 72 69
6E 67 3E 4B 79 6F 74 6F 1F 4B 79 6F 74 6F 1F 4A 61 70 61 6E
3C 2F 73 74 72 69 6E 67 3E 3C 68 65 74 6E 3E 4B 79 6F 74 6F
2C 4B 79 6F 74 6F 2C 4A 61 70 61 6E 0A 3C 2F 68 65 74 6E 3E
3C 2F 73 75 6D 6D 61 72 79 5F 64 65 74 61 69 6C 73 3E 3C 73 74 72 69
75 6D 6D 61 72 79 5F 64 65 74 61 69 6C 73 3E 3C 73 74 72 69
6E 67 5F 6C 65 6E 67 74 68 3E 33 33 3C 2F 73 74 72 69 6E 67
5F 6C 65 6E 67 74 68 3E 3C 73 74 72 69 6E 67 3E 4B 79 6F 74
6F 20 43 69 74 79 1F 4B 79 6F 74 6F 20 50 72 65 66 65 63 74
75 72 65 1F 4A 61 70 61 6E 3C 2F 73 74 72 69 6E 67 3E 3C 68
65 74 6E 3E 4B 79 6F 74 6F 20 43 69 74 79 2C 4B 79 6F 74 6F
20 50 72 65 66 65 63 74 75 72 65 2C 4A 61 70 61 6E 0A 3C 2F
68 65 74 6E 3E 3C 2F 73 75 6D 6D 61 72 79 5F 64 65 74 61 69
6C 73 3E 3C 73 75 6D 6D 61 72 79 5F 64 65 74 61 69 6C 73 3E
3C 73 74 72 69 6E 67 5F 6C 65 6E 67 74 68 3E 31 35 3C 2F 73
74 72 69 6E 67 5F 6C 65 6E 67 74 68 3E 3C 73 74 72 69 6E 67
3E 4E 61 72 61 1F 4E 61 72 61 1F 4A 61 70 61 6E 3C 2F 73 74
72 69 6E 67 3E 3C 68 65 74 6E 3E 4E 61 72 61 2C 4E 61 72 61
2C 4A 61 70 61 6E 0A

```

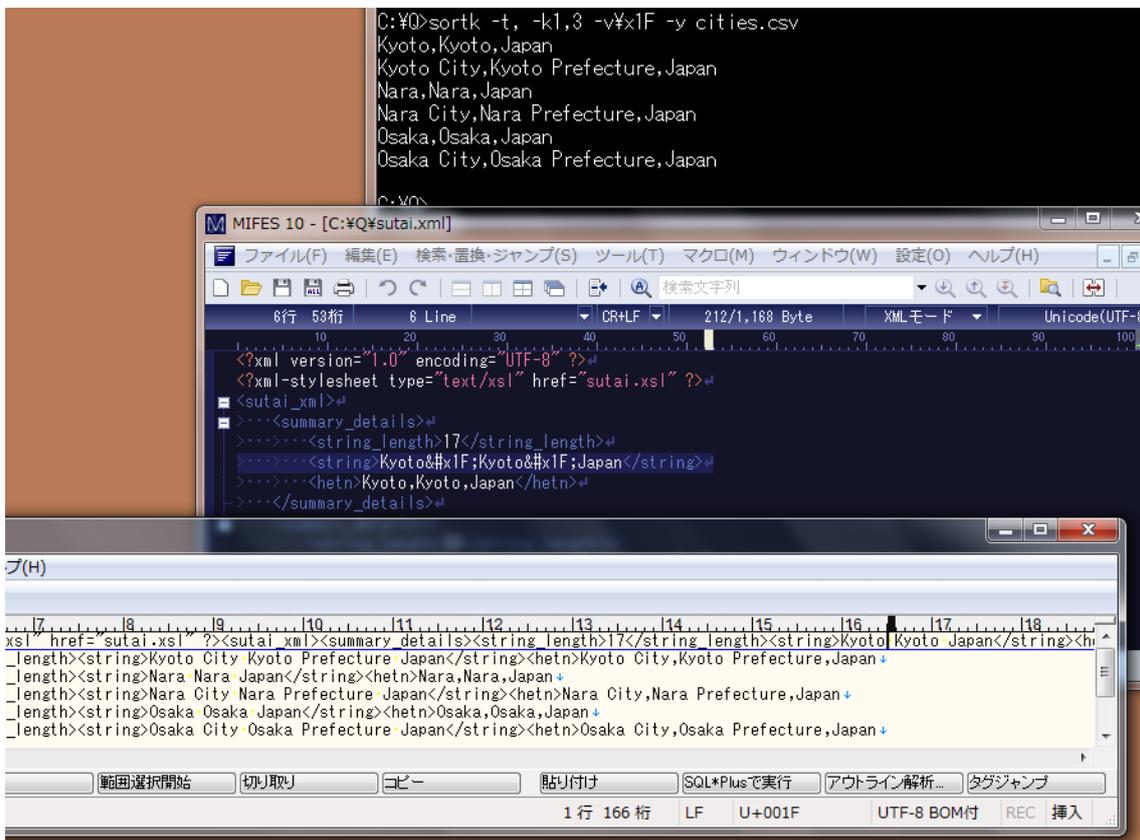
### 語句が短いほど上に来る

語句が短いほど上に来るように、本来の空白を感嘆符に置換して、さらに連結文字(1Fh)を空白に置換します。21h(これは、文書に由来する空白に対応しています)は、20h(連結文字)より大きいいため、Nara City,の上にNara,が来ます。(sortk コマンドは、要約を比較して、要約および詳細のブロックを並べ替えて、詳細のみ書き出します。連結文字は、要約に含まれているため、書き出されません。)

主語	表現	ファイル	工程
利用者	-v¥x1F		パラメータとして入力した。
sortk コマンド	1Fh		数に変換した。
C#の XmlDocument	&#x1F;	sutai.xml	書き出した。
sortk コマンド	1Fh		読み込んで、数に変換した。
sortk コマンド	20h		さらに、空白に置換した。
	21h	cities.csv	文書に由来する空白を感嘆符に置換した。

### サクラエディタのカーソルを移動

上記の第 6 の工程の結果として、**intermediate-3.tmp** を表示してサクラエディタのカーソルを移動すれば、ステータスバーに U+001F UTF-8 が表示されます。すなわち、sortk コマンドが&#x1F;を 1Fh に置換したことを確認できます。



### 課題

C#版の sortk コマンドは、感嘆符を空白に置換するため、感嘆符が文書に含まれる場合、空白の上に感嘆符が来ます。C++版も C 版もこの問題は、発生しません。

```
C:¥Q>TYPE confuse.txt
Nara City,Nara Prefecture,Japan
Nara!City,Nara Prefecture,Japan

C:¥Q>sortk -t, -k1,3 confuse.txt
Nara!City,Nara Prefecture,Japan
Nara City,Nara Prefecture,Japan

C:¥Q>
```

## 引用符を変更したい

引用符が出現したら、sortk 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( ")が欄内に表示される場合、**sortk 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、sortk コマンドのパラメータとして、-w~を付加してください。sortk 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。

```
C:¥Q>sortk -t, -k2,2 Mark-14.txt
35,~And going a little farther, he fell on the ground and prayed that, if it were possible, the hour might pass him.~
36,~And he said, "Abba, Father, all things are possible for you. Remove this cup from me. Yet not what I will, but what you will."~~~
34,~And he said to them, "My soul is very sorrowful, even to death. Remain here and watch."~~~

C:¥Q>sortk -t, -k2,2 -w~ Mark-14.txt
35,~And going a little farther, he fell on the ground and prayed that, if it were possible, the hour might pass him.~
34,~And he said to them, "My soul is very sorrowful, even to death. Remain here and watch."~~~
36,~And he said, "Abba, Father, all things are possible for you. Remove this cup from me. Yet not what I will, but what you will."~~~

C:¥Q>
```

### 語句が短いとみなされた

上記の例では、まず、-wA(A は、ASCII の文字です)のパラメータを付加しない sortk コマンドを実行しました。次に-w~を付加してみました。-w~が付加されていない場合、sortk コマンドは、And he said の次も区切り文字としてコンマを使用して、語句が短いほど上に来るように行を並べ替えます。このため、マルコ 14 章 36 節がマルコ 14 章 34 節の上に来ました。

### いわゆる引用符もコンマも無視するべきだ

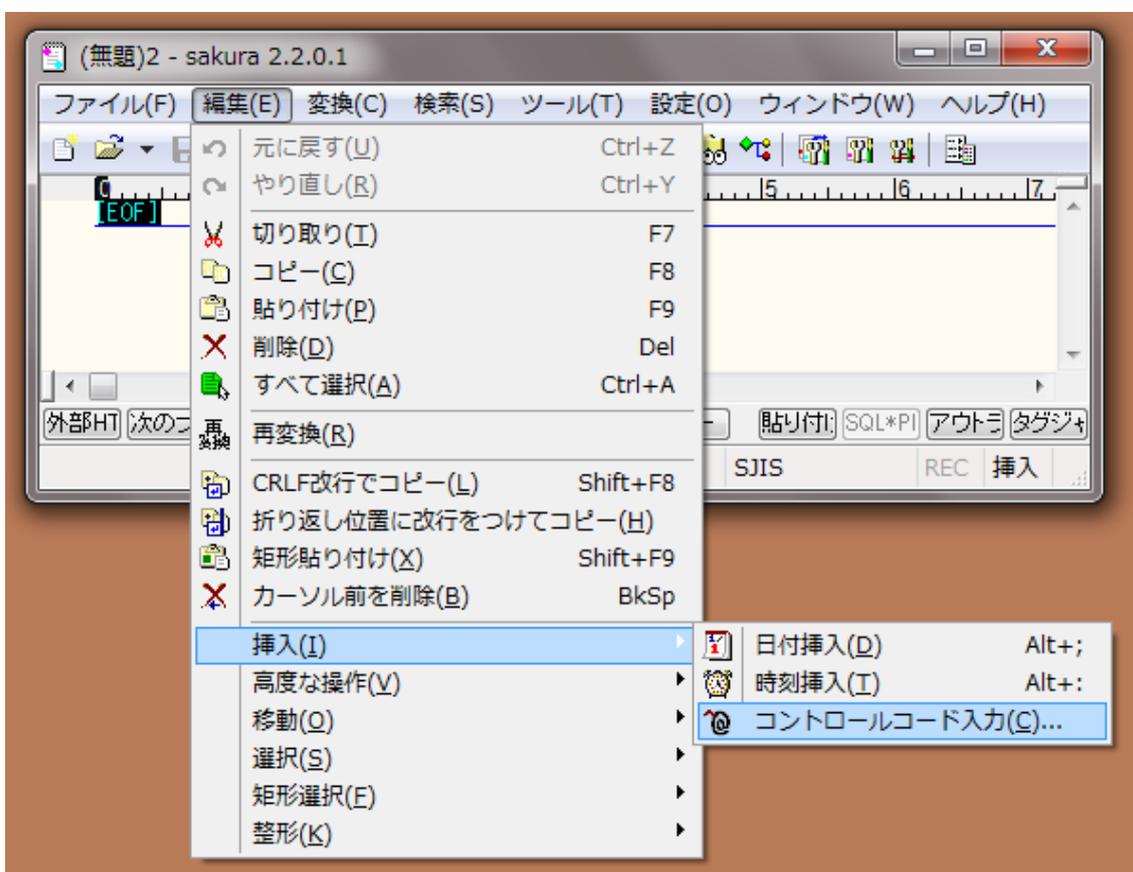
自然な表現として、引用符もコンマも sortk コマンドに無視されるべきである場合、-w~ を付加してください。CSV であるにもかかわらずコンマを無視したい範囲を~で囲んでください。

## 制御文字

多機能のテキストエディタは、制御文字を入力する機能があると思われます。

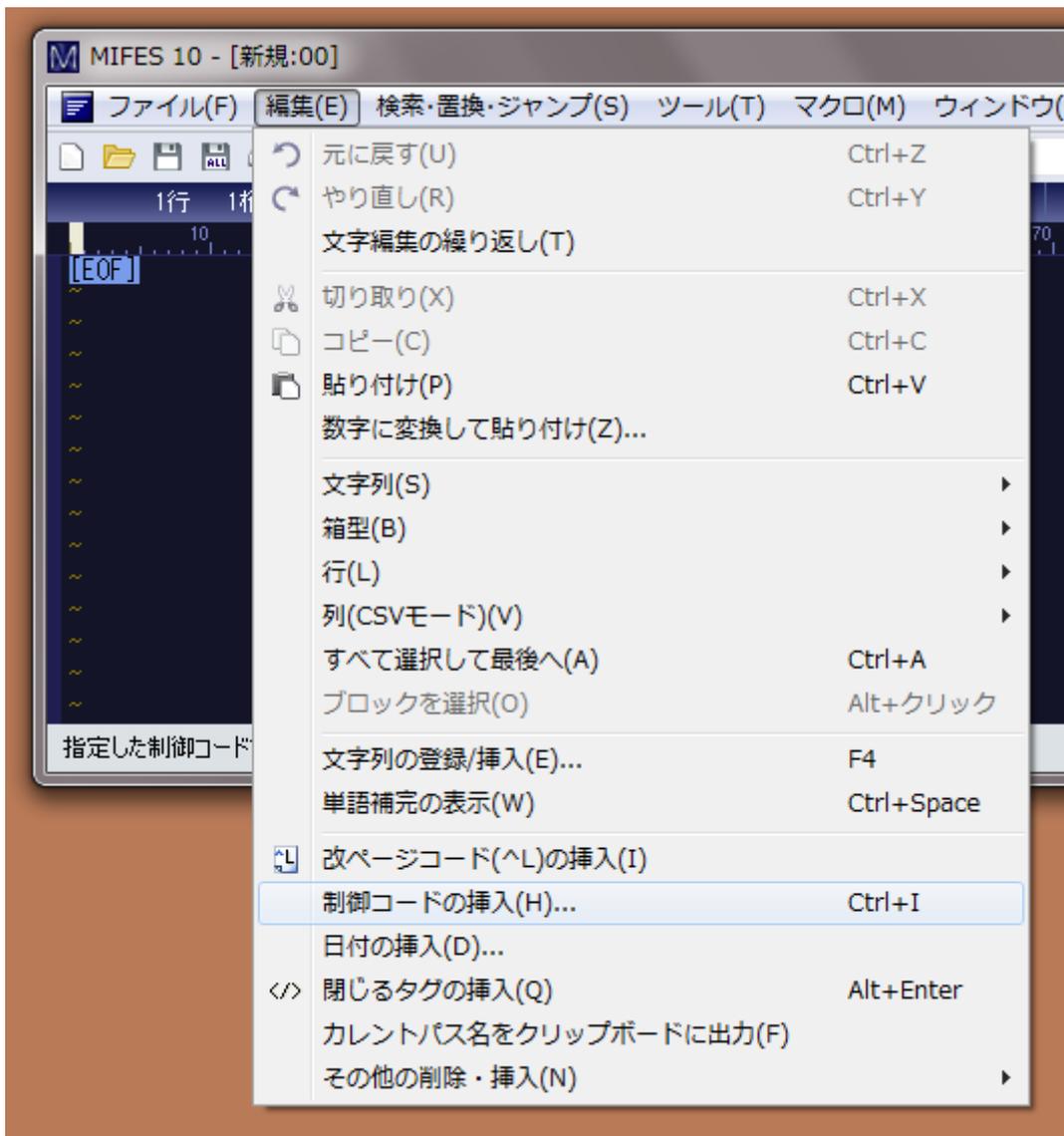
### サクラエディタ

サクラエディタの場合、編集メニューから挿入を選択して、サブメニューからコントロールコード入力を選択できます。



### MIFES

MIFES の場合、編集メニューから[制御コードの挿入]を選択できます。



### タブおよび改ページ

シフト JIS のテキストファイルである場合、タブ、改ページ、エスケープシーケンスなどの制御文字を無視して比較するには、sortk コマンドのパラメータとして、-i を付加してください。制御文字の例

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	substitute	SUB	MS-DOS テキストファイルの終わり
1Bh	27	escape	ESC	エスケープシーケンスの開始

### 制御文字が含まれている

下記の例では、`mini.txt` という CSV にタブ、改ページ、エスケープシーケンスが含まれています。`-i` 無しの `sortk` コマンドは、文字または数字をタブと比較できます。

### 改ページが XML の文字列になった

改ページを `sutai.xml` に書き出したとき、数値ではなく、XML の文字列(`&#xC;`)になりました。`-i` 無しの `sortk` コマンドは、文字または数字を改ページ(0Ch)と比較できません。ただし、通常は、文字または数字を改ページと比較しないで、改ページを無視するのが好適です。

### 数値に変換

`sutai.xml` に書き出したとき、数値ではなく、XML の文字列になりましたが、`sortk` コマンドが [XML の文字列を数値に変換する](#) ように命令するには、`sortk` コマンドがのパラメータとして、`-N` を付加してください。N は、大文字です。たとえば、改ページは、`&#xC;` から 0Ch に変換されます。

### シフト JIS のテキスト

テキストの文字コードがシフト JIS であることが明白であれば、制御文字を無視するために、`sortk` コマンドのパラメータとして、`-i` を付加してください。この場合、`sutai.xml` は、書き出されません。

### 整数を比較

`mini.txt` の第 2 欄は整数であるため、制御文字を無視する目的で、欄から整数のみ抽出できます。欄にある整数を比較するには、`sortk` コマンドのパラメータとして、`-n` を付加してください。n は、小文字です。この場合、`sutai.xml` は、書き出されます。

パラメータ	ファイル	説明
<code>-i</code>		シフト JIS テキストの制御文字を無視する。
<code>-N</code>	<code>sutai.xml</code>	XML 文字列を数値に変換する。
<code>-n</code>	<code>sutai.xml</code>	XML を書き出す前に、欄から整数のみ抽出する。

```
C:¥Q>TYPE mini.txt
4429回,24,
4647回,64,
4759回,28,
4840回,27,
4851回,38,
4881回,86,
4902回,59,
4908回,14,
4944回,19,
4954回,93,
4956回,01,
4960回,25,
4967回,51,
4972回,20,
4973回,41,
4976回,18,
4983回,90,
5000回,17,
5006回,39,
5009回,74,
5014回,40,
5025回,68,
5030回,70,
5032回,83,
5035回,12,
5037回,94,
5043回,15,
5049回,32,

C:¥Q>sortk -t, -k2,2 mini.txt | atnd -h8
4881回,86,
5037回,&#x1B;94,
5014回,&#xC;40,
4956回,01,
5035回,12,
4908回,14,
5043回,15,
5000回,17,

C:¥Q>
```

#### 制御文字は無視されるべきだ

-h8 を付加した atnd コマンドは、ファイルの第 1 行から第 8 行を標準出力に書き出します。atnd コマンドは、標準入力から読み込むことができます。sortk コマンドおよび atnd コマンドのパイプを構成することで、8 行を標準出力に書き出します。86 は 01 より大きいにもかかわらず、上図は、タブが比較されて 86 が 01 より上に来ました。タブ

は、無視されるべきであることがわかります。

### 期待された結果

下記の例では、`mini.txt` というシフト JIS テキストにタブ、改ページ、エスケープシーケンスが含まれていますが、`-i` 付きの `sortk` コマンドは、こうした制御文字を無視して多数の二桁数字を並べ替えることができました。

```
C:¥Q>sortk -t, -k2,2 -i mini.txt
4956回,01,
5035回,12,
4908回,14,
5043回,15,
5000回,17,
4976回,18,
4944回,19,
4972回,20,
4429回,24,
4960回,25,
4840回,27,
4759回,28,
5049回,32,
4851回,38,
5006回,39,
5014回,40,
4973回,41,
4967回,51,
4902回,59,
4647回,64,
5025回,68,
5030回,70,
5009回,74,
5032回,83,
4881回,86,
4983回,90,
4954回,93,
5037回,94,
```

### XML を書き出して並べ替えたい

添付の `hetn` コマンドは、XML を再利用できます。上記のとおり、`-i` 付きの `sortk` コマンドは、期待された結果を生じましたが、XML を書き出しません。`mini.txt` の第 2 欄は整数であるため、制御文字を無視する目的で、欄から整数のみ抽出できますが、XML を書き出したとき、XML の文字列(`&#xc;`)が出力されます。`-i` 付きの `sortk` コマンドと同様な結果を生じるには、XML の文字列を数値に変換するために、さらに`-N` を付加してください。まず、`-n` 付きの `sortk` コマンドを実行して、次に、`-n -N` を付加して `sortk` コマンドを実行してみました。

```
C:\%Q>sortk -t, -k2,2 -n mini.txt | atnd -h19
4956回,01,
5035回,12,
4908回,14,
5043回,15,
5000回,17,
4976回,18,
4944回,19,
4972回,20,
4429回,24,
4960回,25,
4840回,27,
4759回,28,
5049回,32,
4851回,38,
5006回,39,
5014回,&#xC;40,
4973回,41,
4967回,51,
4902回,59,

C:\%Q>sortk -t, -k2,2 -n -N mini.txt | atnd -h19
4956回,01,
5035回,12,
4908回,14,
5043回,15,
5000回,17,
4976回,18,
4944回,19,
4972回,20,
4429回,24,
4960回,25,
4840回,27,
4759回,28,
5049回,32,
4851回,38,
5006回,39,
5014回, 40,
4973回,41,
4967回,51,
4902回,59,

C:\%Q>
```

## 浮動小数点数

浮動小数点数を比較するには、`-g` を付加してください。C++版 `sortk` コマンドと異なり、

C#版 sortk コマンドは、20 欄まで選択できます。

### 桁数が同一ではない

桁数が同一ではない整数を比較するには、`-n` を付加してください。C++版 sortk コマンドと同様に、C#版 sortk コマンドは、20 欄まで選択できます。`-n` が指定された場合、sortk コマンドは、もはや語句を比較できません。

## 欄の前半にある空白

空白は、制御文字ではありません。比較開始欄の前半にある空白を無視するには、sortk コマンドのパラメータとして、`-b` を付加してください。下記の例では、添付の `atnd` コマンドで標準出力の 3 行のみコマンドプロンプトに表示しました。空白が無視されたため、並べ替えた結果が異なります。空白を無視するために `-b` を付加した sortk コマンドは、整数でも浮動小数点数でもなく語句を比較します。

```
C:\>sortk -t, -k1,3 isbn-r.csv | atnd -h3
71,2014,21.00mm, ISBN 978-0-444-63425-2,Elsevier,Dopamine,,By,Marco Diana,
73,2000,38.58mm, ISBN 0-19-853768-9,Oxford,Temporal Logic,,By,Dov M. Gabbay,
73,2010,21.77mm, ISBN 978-0-262-51311-1,MIT,What Is Addiction?,,By,Don Ross,

C:\>sortk -t, -k1,3 -b isbn-r.csv | atnd -h3
100,1993,28.08mm, ISBN 0-8493-4441-7,CRC,Limbic Motor Circuits and Neuropsychiatr
y,,By,Peter W. Kalivas,
100,2010,12.30mm, ISBN 978-0-7618-5041-0,Hamilton,Come Along,We Are Truth-Bound,B
y,Jacinta Respondowska,
100,2010,33.85mm, ISBN 978-1-84882-127-9,Springer,Atlas of Epilepsies,Volume 1,By
,C. P. Panayiotopoulos,

C:\>
```

### 小数点が発見されない

浮動小数点数を比較するために `-g` が付加された sortk コマンドは、小数点を発見しません。小数点が発見されない場合、0.0 になります。

## 0.0

### 数字に到達するまで無視される

浮動小数点数を比較するために `-g` が付加された sortk コマンドは、比較開始欄の前半にある空白を無視します。負号、小数点、数字に到達するまで、空白だけではなく文字を無視します。整数を比較するために `-n` が付加された sortk コマンドは、比較開始欄の前半にある空白を無視します。負号、数字に到達するまで、空白だけではなく文字を無視します。

## 制限

32ビット版 Windows において、C#で sortk コマンドを構築しました。ソースコードを添付

しました。用途によっては、sortk コマンドの利用者は、限界、制限を感じるかもしれません。

### 欄数の上限

利用者は、3072 欄まで選択できます。sortk コマンドは、利用者選択の欄で要約を構成します。

ファイル(.cs)	ソースコード(const int)	説明
FieldsList	FIELD_COUNT = 3072	欄数の上限
CheckID	LIMIT_OF_MY_MEMORY = 128	合致を記憶
IntegerBlock DoubleBlock DescendingIntegerBlock DescendingDoubleBlock	LINE_ID = 20	20 個まで数を選択できる。21 個目として ID を付与できる。

### 要約のサイズ

要約に含める浮動小数点数は、最高 20 個です。要約に含める整数も、最高 20 個です。語句の長さは、無制限です。

### ID を比較できる

浮動小数点数であれ整数であれ、利用者は、20 個まで数を選択できます。21 個目として、ID を比較できます。ある要約が他の要約と合致した場合も順序を確定するために、要約に ID を付与するには、sortk コマンドのパラメータとして、-#を付加してください。要約に ID を付与した効果については、後述します。

## 参考 2

C#版の sortk コマンドは、語句の長さが無制限ですが、C++版および C 版の sortk コマンドは、最大 79 バイトまで比較すれば順序が確定することを想定しています。

### 浮動小数点数の個数

C#版の sortk コマンドは、20 個の浮動小数点数を比較できますが、C++版および C 版の sortk コマンドは、10 個の浮動小数点数を比較できます。

### 一時ファイル

C#版の sortk コマンドは、一時ファイルとして、sutai.xml を作成しますが、C++版および C 版の sortk コマンドは、binary.tmp というバイナリファイルを作成します。テキストエディタでは、binary.tmp を読むことができません。

sortk	ライブラリ	パラメータ	説明
C		-y	binary.tmp を削除しない。
C++		-y	binary.tmp を削除しない。
C#	.NET Framework 3.5	-i	そもそも sutai.xml を作成しない。
		-y	一時ファイルを削除しない。一時ファイルは、バイナリファイルではない。
DMC	STLport 4.5.3	-y	binary.tmp を削除しない。

#### 一時ファイルを再利用できる

C#版の sortk コマンドは、一時ファイルとして、sutai.xml を作成します。hetn コマンドは、sutai.xml を再利用できます。C++版および C 版の sortk コマンドは、一時ファイルとして、binary.tmp を作成します。keyun コマンドは、binary.tmp を再利用できます。

## 大文字と小文字を区別しない

C#版の sortk コマンドは、全体として大文字と小文字をほとんど区別しないと思われます。大文字であれ小文字であれ、他の文字と区別されて順序が確定しますが、大文字と小文字は、同じような位置に来ます。正確には、小文字は、大文字より上に来ます。

#### 明確に命令したい

大文字と小文字を区別しないで比較するように明確に命令するには、sortk コマンドのパラメータとして、-f を付加してください。-f が付加されたかどうかによって結果が異なる例を示します。

```
C:¥Q>TYPE cookies.txt
chocolate cookie,
chocolate chip cookie,
gingerbread cookie,
oatmeal cookie,
sugar cookie,
home-made cookie,
fortune cookie,
plate of cookies,
bag of cookies,
batch of cookies,
box of cookies,
Girl Scouts went around the neighborhood selling cookies.,
cookie crumb,
cookie dough,
cookie jar,
cookie cutter,
Cookie Cutter,
cookie cutter,
cookie sheet,
cookie recipe,
cookies and milk,
milk and cookies,
```

```
C:¥Q>sortk -t, -k1,1 cookies.txt | atnd -h9
bag of cookies,
batch of cookies,
box of cookies,
chocolate chip cookie,
chocolate cookie,
cookie crumb,
cookie cutter,
cookie cutter,
Cookie Cutter,
```

```
C:¥Q>sortk -t, -k1,1 -f cookies.txt | atnd -h9
bag of cookies,
batch of cookies,
box of cookies,
chocolate chip cookie,
chocolate cookie,
cookie crumb,
cookie cutter,
Cookie Cutter,
cookie cutter,
```

## 行数

ファイルの行数は、16384 を超えてはなりません。行数の上限を変更するには、sortk コマンドのパラメータとして、-U 数を付加してください。たとえば、16384 ではなく、131072 を上限とするには、-U131072 を付加してください。下記の例では、-U20 を付加してみました。atnd コマンドで 5156 行もあることがわかりましたが、第 1 行から第 20 行までのみ並べ替えることができました。

```
C:¥Q>atnd -LF -p20 numbers7.csv
numbers7.csv: 5156 lines
maximum LF-to-LF span of 13 bytes (line-1000)
20,502,2264

C:¥Q>sortk -t, -k2,2 -U20 numbers7.csv
15,019,6419
4,105,0171
16,179,0524
1,191,1149
3,194,0097
13,229,4673
8,234,7507
9,243,2664
12,247,9722
18,291,3062
11,326,0254
10,340,2436
17,475,8697
20,502,2264
5,592,0265
7,708,3950
14,743,1409
6,792,5482
19,852,1013
2,988,7921

C:¥Q>
```

### 欄数の上限

利用者は、3072 欄まで選択できます。sortk コマンドは、利用者選択の欄で要約を構成します。

### 欄が少ない

欄が少ないことが判明しているならば、-a 欄数を付加してみてください。下記の例では、-a3 を付加したことで経過時間が短縮したことがわかります。

```
C:¥Q>atnd -h10 numbers7.csv
1,191,1149
2,988,7921
3,194,0097
4,105,0171
5,592,0265
6,792,5482
7,708,3950
8,234,7507
9,243,2664
10,340,2436

C:¥Q>atnd -LF -p10 numbers7.csv
numbers7.csv: 5156 lines
maximum LF-to-LF span of 13 bytes (line-1000)
10,340,2436

C:¥Q>sortk -t, -k2,2 -T -o text.tmp numbers7.csv
493662 ticks

C:¥Q>sortk -t, -k2,2 -T -o text.tmp numbers7.csv
500217 ticks

C:¥Q>sortk -t, -k2,2 -T -o text.tmp numbers7.csv
490086 ticks

C:¥Q>sortk -t, -k2,2 -T -a3 -o text.tmp numbers7.csv
375458 ticks

C:¥Q>sortk -t, -k2,2 -T -a3 -o text.tmp numbers7.csv
363205 ticks

C:¥Q>sortk -t, -k2,2 -T -a3 -o text.tmp numbers7.csv
366958 ticks

C:¥Q>_
```

## dayu コマンドと連携する

添付の dayu コマンドは、MS932~Shift\_JIS.bin という変換表を編集できます。変換表は、バイナリファイルです。もうひとつの機能として、dayu コマンドは、シフト JIS のテキストファイルを規格化できます。

パラメータ	設定ファイル	説明
-r-		シフト JIS のテキストファイルを規格化しない。
-e-		MS932~Shift_JIS.bin(変換表)を編集しない。
	twenty.txt	変換表の編集およびテキストの規格化を行う。

## すべて下線になる

dayu コマンドが MS932 から Shift\_JIS にテキストファイルを規格化するとき、丸付きの数字(①②③⑩⑳)、ローマ数字(ⅠⅡⅢⅣⅩ)などの機種依存文字は、すべて下線(   )に変更されます。1 バイトのカタカナ(半角カタカナ)は、Shift\_JIS に含まれていますが、dayu コマンドは、半角カタカナを廃止して、ASCII の下線(   )に変更します。

## 1 文字ずつ編集

①を下線ではなく1で代用するには、dayu コマンドのパラメータとして **-m① -s1** を付加してください。通常の文字および代用文字のシフト JIS 文字コードが十六進数で表示されます。

### 標準入力モードから脱出

規格化したいファイルが設定されていない場合、dayu コマンドは、自動的に標準入力から読み込んで標準出力に書き出します。標準入力のモードから脱出するには、たとえば、Ctrl を押しながら Z を押してください。^Z がコマンドプロンプトに表示されます。^Z は、ファイルの終わり(EOF)を意味します。

### 規格化しない

変換表を編集するが、シフト JIS のテキストファイルを規格化しないように命令するには、dayu コマンドのパラメータとして、**-r-** を付加してください。dayu コマンドは、標準入力から読み込もうとしません。

### どの変換表を編集するか

dayu コマンドがアクセスできる変換表のパス名を表示するには、コマンドのパラメータとして、**-w** を付加してください。パス名ではなく単なるファイル名が表示された場合、カレントディレクトリにある変換表です。

**dayu -m① -s1 -r- -w**

### 編集しない

シフト JIS のテキストファイルを規格化するが、変換表を編集しないように命令するには、dayu コマンドのパラメータとして、**-e-** を付加してください。

```
C:¥Q>COPY CON maruni.txt
①は、下線ではない。
②は、下線だ。
③も下線だ。
^Z
    1 個のファイルをコピーしました。

C:¥Q>dayu -m① -s1 -r- -w
C:¥sortkCS¥MS932~Shift_JIS.bin is primarily accessible to Dayu.
8740 ⇒ 8250

C:¥Q>DIR | FINDSTR "MS932~Shift_JIS.bin"
2019/07/24  00:35                31,528 MS932~Shift_JIS.bin

C:¥Q>COPY MS932~Shift_JIS.bin C:¥sortk /Y
    1 個のファイルをコピーしました。

C:¥Q>dayu -e- maruni.txt text.tmp

C:¥Q>TYPE text.tmp
1は、下線ではない。
_は、下線だ。
__も下線だ。

C:¥Q>
```

## バージョン番号

dayu コマンドのバージョン番号を表示するには、-V を付加してください。

### 短い説明

短い説明を表示するには、-VC を付加してください。

```

C:¥Q>dayu -VC
Dayu C# Version 1.02

dayu MS932 # Read MS932 to produce dayu.txt
dayu MS932 SJIS # Read MS932 to produce a new file.
dayu < MS932 # Every line includes 0Dh.
dayu -b MS932 # Produce binary.tmp
dayu -b MS932 MS932 # Produce and rename binary.tmp
dayu -b -y MS932 # Do not remove binary.tmp
dayu -d MS932 # Do not search a table.
dayu -e- MS932 # Do not edit MS932~Shift_JIS.bin
dayu -f bars.txt MS932 # Edit MS932~Shift_JIS.bin according to bars.txt
dayu -i MS932 # Skip control characters except 0Ah.
dayu -kナ -aN # ナ is substituted with N.
dayu -L MS932 # 0Dh is removed from every line.
dayu -m① -s1 -r- # Edit MS932~Shift_JIS.bin without reading MS932
dayu -m① -s1 MS932 # ① is substituted with 1.
dayu -q MS932 # Acquire a table name from twenty.txt
dayu -t MS932 SJIS # MS932 has the same time as SJIS.
dayu -w MS932 # -w shows which table dayu uses.

```

パラメータ	密接に関係するファイル	説明
-b	シフト JIS テキスト	binary.tmp を作成します。
-CRLF	シフト JIS テキスト	0Dh および 0Ah で改行します。
-d	MS932~Shift_JIS.bin	変換表を探索してはならない。
-e-	MS932~Shift_JIS.bin	変換表を編集しません。
-f	代用文字のリスト。 MS932~Shift_JIS.bin	多数の代用文字にわたって変換表を編集します。
-i	シフト JIS テキスト	0Ah 以外の制御文字を削除します。
-kナ	MS932~Shift_JIS.bin	ある半角カタカナを廃止します。
-aN	MS932~Shift_JIS.bin	ASCII の 1 文字で代用します。
-LF	シフト JIS テキスト	0Ah のみで改行します。
-m①	MS932~Shift_JIS.bin	ある機種依存文字を廃止します。
-s1	MS932~Shift_JIS.bin	Shift_JIS の 1 文字で代用します。
-r-	シフト JIS テキスト	シフト JIS テキストを規格化しません。
-w	MS932~Shift_JIS.bin	どの変換表を使用するか表示します。
-y	シフト JIS テキスト	binary.tmp を削除してはならない。