

SjisEuc ユーザーズガイド

sjiseuc コマンドは、シフト JIS のテキストファイルを変換して EUC-JP のテキストファイルに変換します。



下線で代用

EUC-JP は、Windows ではなく UNIX の日本語です。sjiseuc コマンドがシフト JIS から EUC-JP にテキストファイルを変換するとき、たとえば、丸付きの数字 (①②③⑩⑳)、ローマ数字 (I II III IV X)、1 バイトのカタカナは、EUC-JP で表現しにくいいため、下線に変更されます。

変換表を編集

たとえば、①、②、③が、それぞれ 1、2、3 で代用されるように、添付の diyo コマンドは、変換表を編集できます。

改行

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。sjiseuc コマンドは、各行から 0Dh を削除できます。

標準入力

ファイル名が指定されていない場合、sjiseuc コマンドは、標準入力から行を読み込むことができます。

インストール

sjiseuc コマンドは、32 ビット版 Windows のコンソールアプリケーションです。

パスを通す手順の例

1. トリシーカー (http://tori.tobihiro.jp/PDF-CS_zip.html) というウェブサイトから sjisEUCjpCS をダウンロードしてください。CS は、C#を意味します (C Sharp)。
2. 7-Zip の圧縮ソフトウェアで sjisEUCjpCS.7z を展開してください。
3. C ドライブに sjisEUCjpCS フォルダを移動してください。
4. sjisEUCjpCS フォルダの名前を変更しないでください。
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:¥sjisEUCjpCS 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、sjisEUCjpCS フォルダごと削除してください。Path から sjisEUCjpCS を削除してください。

著作権

Sogaya(そがや)は、sjiseuc コマンド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

sjiseuc 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 sjiseuc コマンドを構築しました。

ユーザーズガイド

2019 年 7 月 28 日 Revision 1.016

sjiseuc コマンドのパラメータ

MS932~EUC-JP.bin というバイナリファイルにしたがって、sjiseuc コマンドは、シフト JIS のテキストファイルを EUC-JP のテキスト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(I II

ⅢⅣⅩ)、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 とも言います。

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開覧する方法として、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。

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 8 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 8 を入力してください。-h8 は、-b7 -p8 を意味します。

atnd -h8 Gongitsune.txt

```
C:\>atnd -h8 Gongitsune.txt
ごん狐
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、「ごん狐《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《ひゃくしょうや》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:\>
```

テキストエディタ

EUC-JP をサポートしているテキストエディタとしては、EmEditor, K2Editor, MIFES,

TeraPad, WZ, サクラエディタなどがあります。

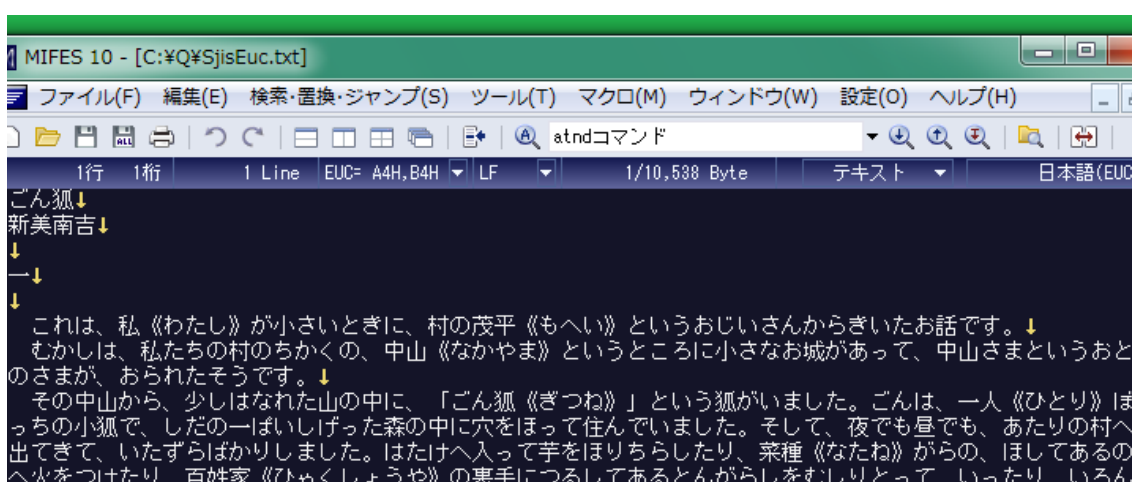
XML ファイルを表示

MIFES は、XML を適切に表示できます。

変換したいファイル

sjiseuc コマンドのパラメータとして、シフト JIS のファイルを指定してください。sjiseuc コマンドは、シフト JIS のファイルを読み込んで、SjisEuc.txt という EUC-JP のファイルを書き出します。

sjiseuc Gongitsune.txt



2 個のファイル

シフト JIS のファイル、EUC-JP のファイルがこの順に指定された場合、sjiseuc コマンドは、シフト JIS のファイルから行を読み込んで、EUC-JP のファイルに行を書き出します。

sjiseuc Shift_JIS.txt EUC-JP.txt

標準入力

sjiseuc コマンドのパラメータとして、ファイル名が指定されなかった場合、sjiseuc コマンドは、標準入力から行を読み込んで、標準出力に書き出します。下記の例では、英語を入力した場合、英語が表示されますが、日本語を入力した場合、文字化けが発生することがわかります。コマンドプロンプトは、シフト JIS を正しく表示します。EUC-JP は、文字化けが発生します。Ctrl を押しながら Z を押すことでファイルの終わり (EOF) を入力しました。^Z が表示されました。

```
C:¥Q>sjiseuc
How do you do?
How do you do?
はじめまして
マ、久、兎、キ、ニ
^Z
C:¥Q>
```

リダイレクト

標準入力から行を読み込む場合、リダイレクトでファイルを作成できます。ただし、0Dh および 0Ah で改行します。

0Dh を削除

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。標準入力から読み込んで標準出力へ書き出す場合、0Dh が含まれます。さもないと、sjiseuc コマンドは、各行から 0Dh を削除します。

-CRLF

0Dh および 0Ah で改行するには、パラメータとして、-CRLF を付加してください。

-LF

リダイレクトではなく sjiseuc コマンドでファイルを作成する場合、0Ah のみで改行するために、パラメータとして、-LF を付加できます。

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。

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。iconv コマンドの結果と、sjiseuc コマンドの結果をテキストファイルとして比較しました。各行が一致しました。さもないと、一致しない付近が表示されます。

```
C:¥Q>iconv -f Shift_JIS -t EUC-JP Gongitsune.txt > text.tmp
C:¥Q>sjiseuc Gongitsune.txt
C:¥Q>FC text.tmp SjisEuc.txt
ファイル text.tmp と SJISEUC.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>
```

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。iconv コマンドの結果と、-CRLF 付きの sjiseuc コマンドの結果をバイナリファイルとして比較しました。0Dh および 0Ah による改行も一致しました。

```
C:¥Q>iconv -f Shift_JIS -t EUC-JP Gongitsune.txt > binary.tmp
C:¥Q>sjiseuc -CRLF Gongitsune.txt
C:¥Q>COMP binary.tmp SjisEuc.txt
binary.tmp と SjisEuc.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>
```

画面を消去

コマンドプロンプトの画面を消去するには、コマンドプロンプトに下記のコマンドを入力して、Enter を押してください。

CLS

バージョン番号

sjiseuc コマンドのバージョン番号を確認するには、コマンドプロンプトに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。コマンドおよびパラメータの例

も表示されます。

sjiseuc -V

```
C:¥Q>sjiseuc -V
SjisEuc C# Version 1.03

sjiseuc SJIS
sjiseuc SJIS EUC
sjiseuc < SJIS > EUC
sjiseuc -a SJIS
sjiseuc -b SJIS
sjiseuc -b SJIS SJIS
sjiseuc -b -y SJIS
sjiseuc -CRLF SJIS
sjiseuc -CRLF SJIS EUC
sjiseuc -d SJIS
sjiseuc -LF SJIS
sjiseuc -s SJIS
sjiseuc -t SJIS EUC
sjiseuc -w SJIS
sjiseuc -V
sjiseuc -VC

C:¥Q>
```

短い説明

コマンドおよびパラメータの各行に短い説明を付与して表示するには、コマンドプロンプトに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。

sjiseuc -VC

パラメータ	密接に関係するファイル	説明
-a	MS932~EUC-JP.bin および sete.txt	sete.txt から変換表のパス名を取得します。
-b	binary.tmp	いったん binary.tmp を作成して、改名します。
-CRLF	EUC-JP テキスト	0Dh および 0Ah で改行します。
-d	MS932~EUC-JP.bin	変換表を探索してはならない
-LF	EUC-JP テキスト	0Ah のみで改行します。
-s	シフト JIS テキスト	0Ah 以外の制御文字を削除します。
-t	シフト JIS テキスト、EUC-JP テキスト	シフト JIS のテキストは、EUC-JP のテキストと同じ日時を有します。
-w	MS932~EUC-JP.bin	どの変換表を使用するか表示します。
-V		バージョン番号および用法を表示します。
-VC		用法、短い説明を表示します。
-y	binary.tmp	binary.tmp を削除してはならない。

実行ファイル

変換表の名前は、MS932~EUC-JP.bin です。sjiseuc コマンドの特徴として、実行ファイル(SjisEuc.exe)の外部から変換表を読み込むことができます。実行ファイルは、自分がインストールされたディレクトリに存在している変換表を読み込みます。

カレントディレクトリ

変換表および実行ファイルが同じディレクトリに存在していない場合、カレントディレクトリから変換表を読み込みます。

Path に記載の各ディレクトリ

変換表を開くことができるまで、sjiseuc コマンドは、下記の順に変換表を探索します。

1. 過去に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. いま SjisEuc.exe を実行しているディレクトリ
5. 利用者が設定ファイル(sete.txt)に記入したパス名
探索してはならない

変換表の探索を抑制するには、sjiseuc コマンドに-d を付加してください。

設定ファイルを優先

設定ファイル(sete.txt)に記入の変換表を優先して使用するには、sjiseuc コマンドのパラメータとして、-a を付加してください。

どの変換表を使用するか

過去に実行ファイルがインストールされたディレクトリか、Path にあるディレクトリか、カレントディレクトリのいずれかに存在している変換表を開くことができたとき、どの変換表を使用するか表示するには、sjiseuc 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の変換表を使用します。

```
sjiseuc Gongitsune.txt -w
```

新しいファイル

COPY コマンドで、シフト JIS の新しいテキストファイルを作成するには、コピーされるファイルとして CON を指定してください。この場合、CON は、パソコンのキーボードです。コマンドプロンプトは、CON がファイルであるかのように COPY コマンドを実行します。

ファイルの終わり

コマンドプロンプトに、ファイルの終わり(EOF)を入力するには、Ctrl を押しながら、Z を押ししてください。Ctrl + Z は、この操作を意味します。^Z が画面に表示される場合があります。

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。

かな漢字変換

コマンドプロンプトで、かな漢字変換を行うには、Alt を押しながら[半角/全角]キーを押ししてください。Windows 7, Windows 10 の場合、Alt を押す必要がありません。かな漢字変換モードから脱出するには、[半角/全角]キーを押ししてください。

F8 で半角カタカナに変換

コマンドプロンプトをかな漢字変換モードに移行して、ローマ字入力法で半角カタカナに変換する手順の例を示します。

ECHO を入力してください。スペースキーを押して空白を入力してください。まだ Enter キーを押さないでください。

Alt を押しながら[半角／全角]キーを押してください。かな漢字変換モードに移行します。

H キーを押してください。

E キーを押してください。へが表示されます。

F8 を押してください。へからへに変化します。

Enter キーを押してください。かな漢字変換が確定します。

もう一回 Enter キーを押してください。ECHO コマンドおよびパラメータが入力されません。

コマンドプロンプトに半角カタカナのへが表示されます。

かな漢字変換モードから脱出するには、[半角／全角]キーを押してください。

変換表を編集する

変換表の名前は、MS932~EUC-JP.bin です。通常、変換表および実行ファイルは、同じディレクトリに存在しています。

diyo コマンド

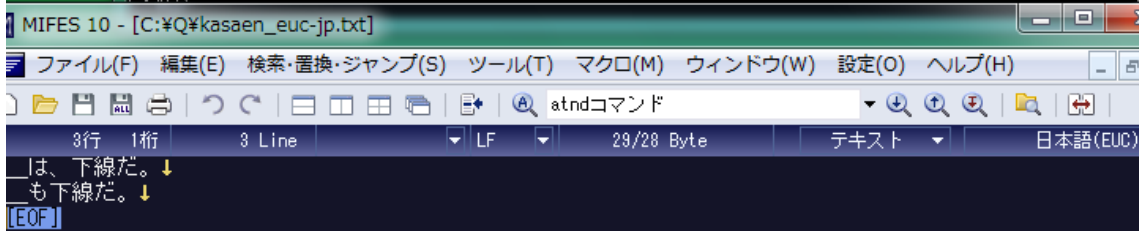
添付の diyo コマンドは、MS932~EUC-JP.bin というバイナリファイルを編集できます。

すべて下線になる

EUC-JP は、Windows ではなく UNIX の日本語です。sjiseuc コマンドがシフト JIS から EUC-JP にテキストファイルを変換するとき、丸付きの数字(①②③⑩⑳)、ローマ数字(I II III IV X)などの機種依存文字は、すべて下線()に変更されます。1 バイトのカタカナは、ASCII の下線()に変更されます。

```
C:\>COPY CON kasen.txt
①は、下線だ。
②も下線だ。
^Z
      1 個のファイルをコピーしました。

C:\>sjiseuc kasen.txt kasaen_euc-jp.txt
C:\>
```



バッチファイルで上書き

diyo コマンドは、変換表を編集します。編集する前の変換表が test_files フォルダに保存されています。添付の ctm.bat は、test_files フォルダから編集前の変換表をコピーします。diyo コマンドがインストールされたディレクトリにある変換表が [上書き](#) されます。下記のコマンドを実行してください。

ctm 0

```
C:\>CD %sjisEUCjpCS
C:\%sjisEUCjpCS>ctm 0
      1 個のファイルをコピーしました。
2018/12/07 13:50          31,528 MS932~EUC-JP.bin
C:\%sjisEUCjpCS>
```

1 文字ずつ編集

たとえば、①を下線ではなく1で代用するには、コマンドプロンプトに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。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。

diyo -s① -e 1

新しいファイル

コマンドプロンプトで、COPY CON を利用して [新しいファイル](#) を作成できます。

どの変換表を編集するか

diyo 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。

diyo -s① -e1 -w

```
C:\>diyo -s① -e1 -w
C:\$jisEUCjpCS¥MS932~EUC-JP.bin is primarily accessible to Diyo.
8740 => 8250

C:\>DIR | FINDSTR "MS932~EUC-JP.bin"
2019/07/11 23:30          31,528 MS932~EUC-JP.bin

C:\>COPY MS932~EUC-JP.bin C:\$jisEUCjpCS /Y
1 個のファイルをコピーしました。

C:\>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:\>sjiseuc maruni.txt maruni_euc-jp.txt
```

1 バイトの文字

1 バイトの文字としては、ASCII および半角カタカナがあります。

```
! " # $ % & ' ( ) * + , - . /
0 1 2 3 4 5 6 7 8 9 : ; < = > ?
@ A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z [ ¥ ] ^ _
` a b c d e f g h i j k l m n o
p q r s t u v w x y z { | } ~ `
ー アイウエオカキクケコサシスセソ
タチツテトナニヌネノハヒフヘホ
ミムメモヤユヨラリルレロワ
```

1 バイトのカタカナ

1 バイトのカタカナは、シフト JIS に含まれています。コマンドプロンプトの表示では、1 バイトのカタカナは、幅が漢字の半分になります。

半角カタカナ

1 バイトのカタカナを半角カタカナとも言います。半角カタカナの代用文字として、ASCII の下線 (_) が sjiseuc コマンドに使用されます。sjiseuc コマンドが、たとえば、ナ の代用として ASCII の下線ではなく N を使用するように変換表を編集するには、コマンドプロンプトに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。k およびナ の間に空白を挿入しないでください。

diyo -k ナ -aN

設定ファイル

diyo コマンドの設定ファイルは、setting.txt です。設定ファイルは、いわゆるシフト JIS (MS932) のファイルです。setting.txt は、必須ではありません。setting.txt を開くことができる場合も、diyo コマンドのパラメータで設定を変更できます。setting.txt による設定を diyo コマンドのパラメータで変更できないと思われる場合、setting.txt を削除してください。

パラメータ	用語	目的	説明
-a	ASCII	半角カタカナを廃止	1 バイトの英字、数字、記号
-e	EUC-JP	機種依存文字を廃止	UNIX の日本語
-k	katakana		半角カタカナ
-s	Shift_JIS		Windows の日本語、MS932

記入した設定を否定するパラメータ

たとえば、-w が記入されていても、diyo コマンドのパラメータとして、-w- を付加すれば、変換表のパス名が表示されません。

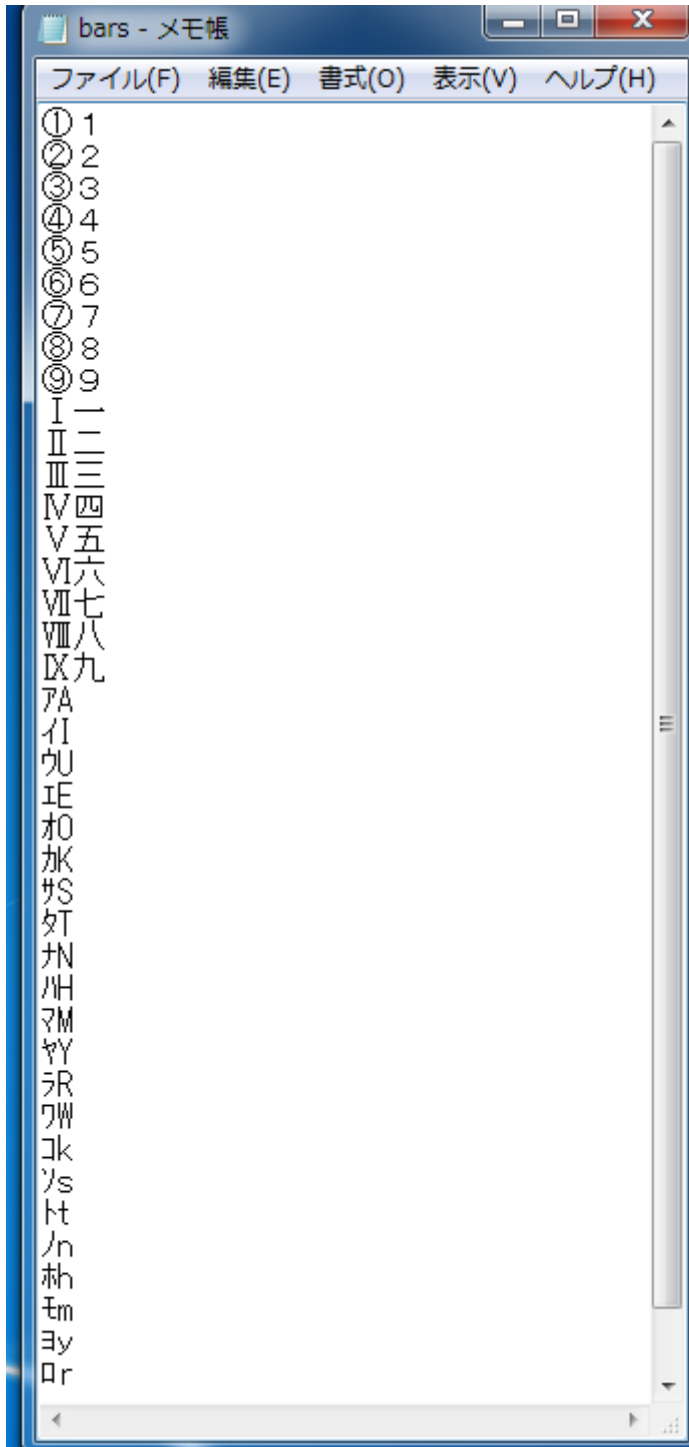
記入した設定	否定する入力	説明
-d	-d-	変換表を探索する / 探索しない
-w	-w-	変換表のパス名を表示する / 表示しない

代用文字のリスト

事前に利用者が代用文字のリストを用意した場合、diyo コマンドは、代用文字のリストにしたがって、一度に多数の文字を編集できます。

通常文字および代用文字

diyo コマンドは、代用文字リストのファイルから 906 行まで読み込むことができます。代用文字リストのファイルの各行は、通常文字と、代用文字とからなります。代用文字もシフト JIS で記入してください。



ファイル名を指定

利用者が代用文字リストのファイルを用意した場合、diyo コマンドのパラメータとして、`-f` を付加して、空白で区切って、さらにファイル名を付加してください。

```
diyo -f bars.txt
```

どの変換表を使用するか

過去に実行ファイルがインストールされたディレクトリか、Path にあるディレクトリか、カレントディレクトリのいずれかに存在している変換表を開くことができたとき、どの変換表を使用するか表示するには、diyo 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の変換表を使用します。

diyo -f bars.txt -w

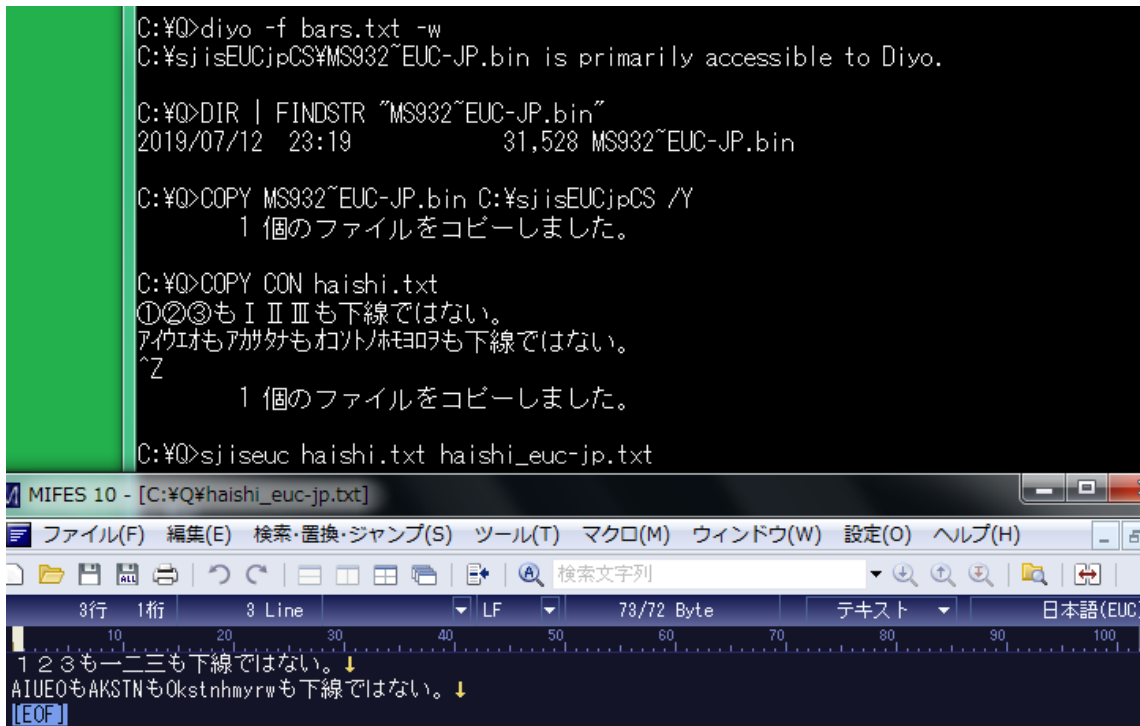
```
C:\>diyo -f bars.txt -w
C:\$jisEUCjpCS$MS932~EUC-JP.bin is primarily accessible to Diyo.

C:\>DIR | FINDSTR "MS932~EUC-JP.bin"
2019/07/12  23:19                31,528 MS932~EUC-JP.bin

C:\>COPY MS932~EUC-JP.bin C:\$jisEUCjpCS /Y
      1 個のファイルをコピーしました。

C:\>COPY CON haishi.txt
①②③も I II III も下線ではない。
アイエオも アカサも カケテヨロも下線ではない。
^Z
      1 個のファイルをコピーしました。

C:\>sjiseuc haishi.txt haishi_euc-jp.txt
```



MIFES 10 - [C:\\$Q\$haishi_euc-jp.txt]

ファイル(F) 編集(E) 検索・置換・ジャンプ(S) ツール(T) マクロ(M) ウィンドウ(W) 設定(O) ヘルプ(H)

検索文字列

3行 1行 3 Line LF 73/72 Byte テキスト 日本語(EUC)

1 2 3 も 一 二 三 も 下線 ではない。 ↓
AIUEO も AKSTN も Okstnhmyrw も 下線 ではない。 ↓
[EOF]

並べ替えに適した代用文字リスト

上記の bars.txt では、ソフトウェアで並べ替えた結果が昇順になりません。代用文字リストのファイルを作成するとき、並べ替えも考慮したい。

sortk コマンド

添付の 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

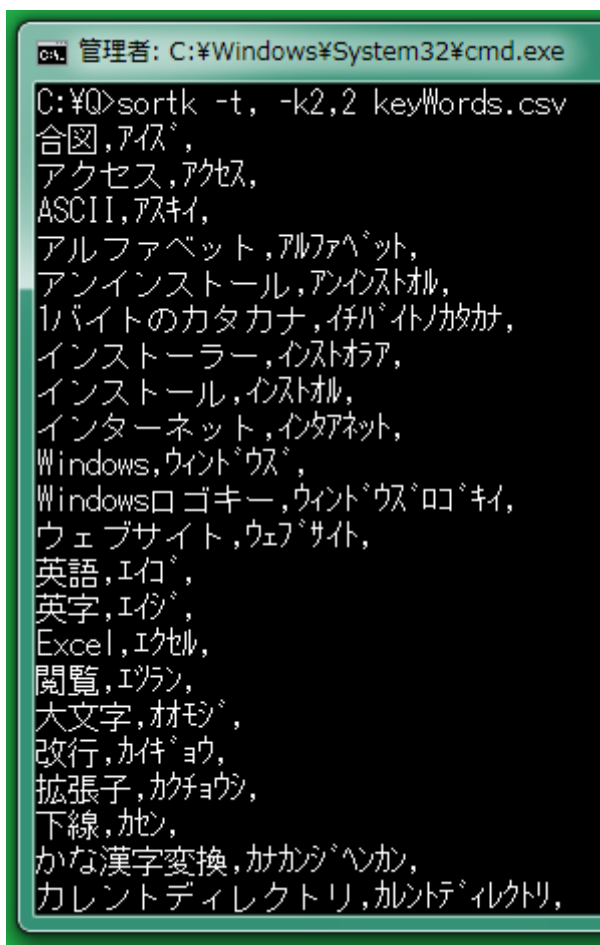
csv を読み込んで、新しいファイルの作成および命名を行うには、コマンドプロンプトに sortk コマンドのパラメータとして、`-o text.tmp` を付加してください。`-o` および `text.tmp` の間に空白を挿入してください。`text.tmp` は、新しいファイルの名前です。

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



```
管理者: C:\Windows\System32\cmd.exe
C:\>sortk -t, -k2,2 keyWords.csv
合図,アズツ,
アクセス,アクセス,
ASCII,アスキー,
アルファベット,アルファベット,
アンインストール,アンイストル,
1バイトのカタカナ,イチッバイトノカタカナ,
インストーラー,イストオラア,
インストール,イストル,
インターネット,イタアネツト,
Windows,ウィンドウズ,
Windows ロゴキー,ウィンドウズ ロゴキー,
ウェブサイト,ウェブサイト,
英語,エイゴ,
英字,エイジ,
Excel,エクセル,
閲覧,エツラン,
大文字,オオモジ,
改行,カクョウ,
拡張子,カクョウジ,
下線,カソン,
かな漢字変換,カナカジヘンカン,
カレントディレクトリ,カレントディレクトリ,
```

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。

sjiseuc コマンド

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。

zip を展開

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 sjisEUCjpCS.zip をダウンロードしてください。CS は、C#を意味します(C Sharp)。sjisEUCjpCS.zip を展開してください。[diyo, sjiseuc, sortk, narfi, rufeol を使用した例](#)

コマンド	文字コード	説明
atnd	シフト JIS	周辺の行を表示する。最長の行の長さを記録する。
dayu	Shift_JIS	変換表の編集およびシフト JIS テキストの規格化を行う。
diyo	EUC-JP	変換表を編集する。
hanzen	シフト JIS	ある欄で、半角カタカナを全角カタカナに置換する。
hetn	UTF-8	sutai.xml を読み込んで、行を並べ替える
narfi	シフト JIS	広い表から細長い表を作成する。ある欄を編集する。
obot	ASCII	ファイルの開始部分のみ 2 桁の十六進数で表示する。
rufeol	シフト JIS	テキストの各行の末尾から不要な文字を除去する。
sortk	シフト JIS	選択された欄を比較して、テキストの行を並べ替える。
sjiseuc	EUC-JP	シフト JIS テキストを EUC-JP テキストに変換する。

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

Sortk を実行するには、.NET Framework 3.5 かそれ以上のバージョンの .NET Framework が必要です。

sortk コマンドのパラメータ

パラメータに関する短い説明を表示するには、sortk コマンドのパラメータとして、-VC または--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。s は、小文字です。

```
sortk -s --help > help.txt
```

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

```
C:\>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¥! -k2,2 file # Wiki table
sortk -t¥! -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.
```

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。
-%		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 の圧縮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>&#xHH;</code> および <code>&#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></hetn></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;^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|

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;^A|-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;^A|-t&yen;x3C|&lt;|小なり記号|

C:¥Q>

```

¥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。

¥

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

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`(マイナスケーニコムマサン)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。

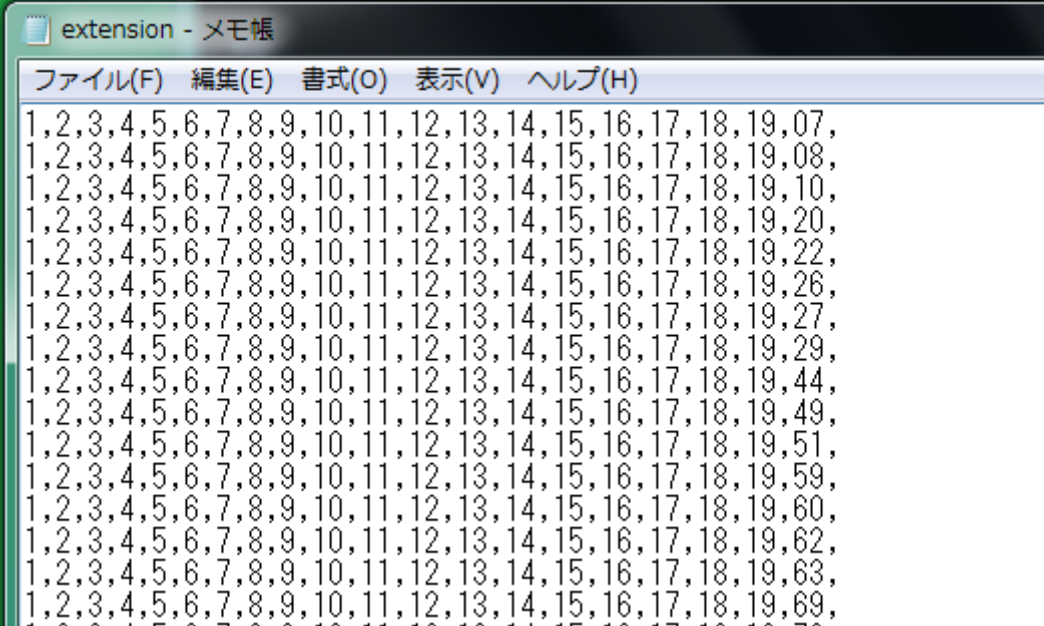
十六進数で入力

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:\>sortk -t, -k10,20 -o text.tmp Field-20.csv
C:\>sortk -t¥x2C -k10-20 -o yen_x2C.csv Field-20.csv
C:\>sortk -k10-20 -o extension.csv Field-20.csv
C:\>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:\>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:\>NOTEPAD extension.csv
```



```
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,07,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,08,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,10,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,26,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,27,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,29,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,44,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,49,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,51,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,60,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,62,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,63,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,69,
```

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。

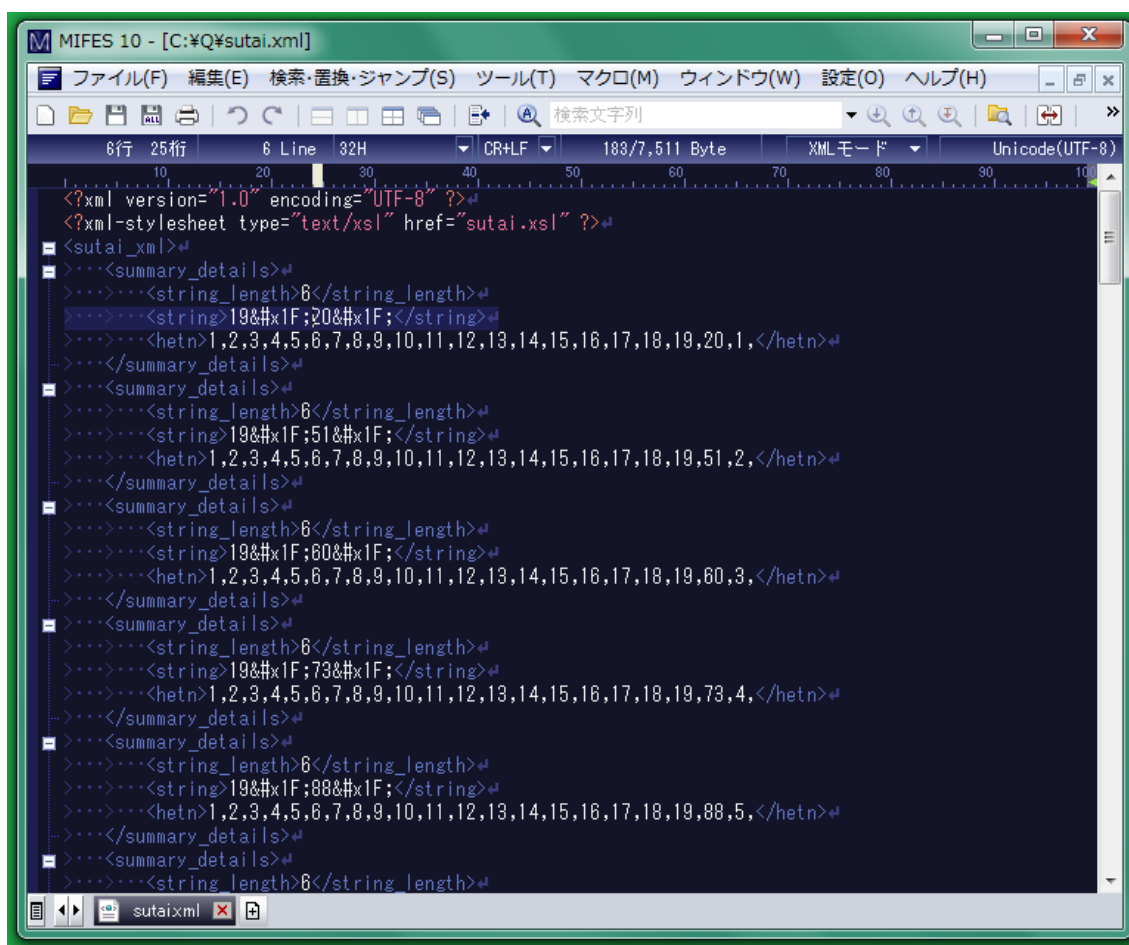
-h10111					
-h	1	0	1	1	1
真偽	第 1 欄選択	第 2 欄を除外する	第 3 欄選択	第 4 欄選択	第 5 欄選択

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 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 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 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>Nara□Nara
```

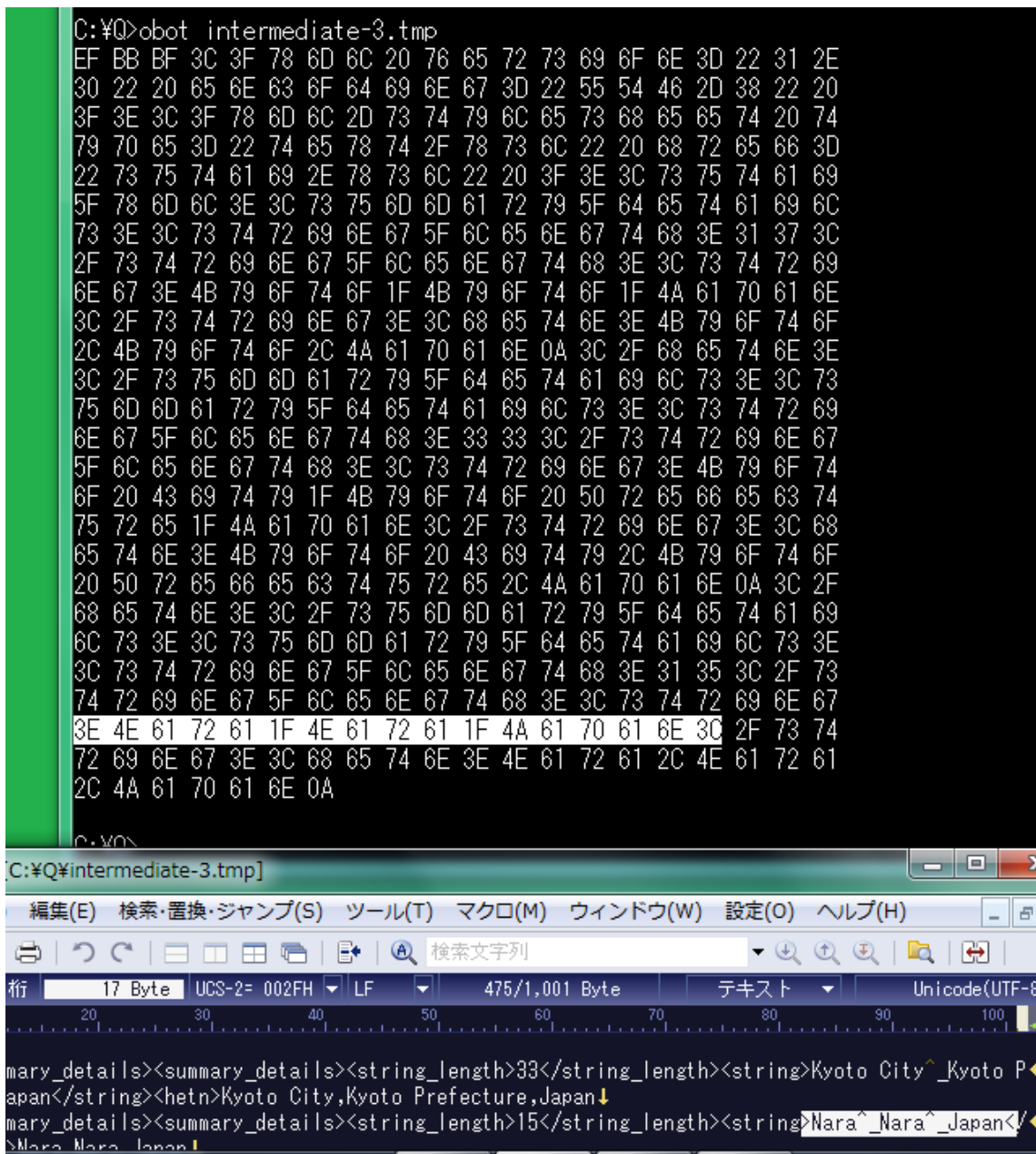
obot コマンド

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 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。

```
>Nara□Nara□Japan<
```

十六進数	文字		十六進数	文字
4E	N		70	p
61	a		6E	n
72	r		3E	>
4A	J		3C	<

```
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
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
```



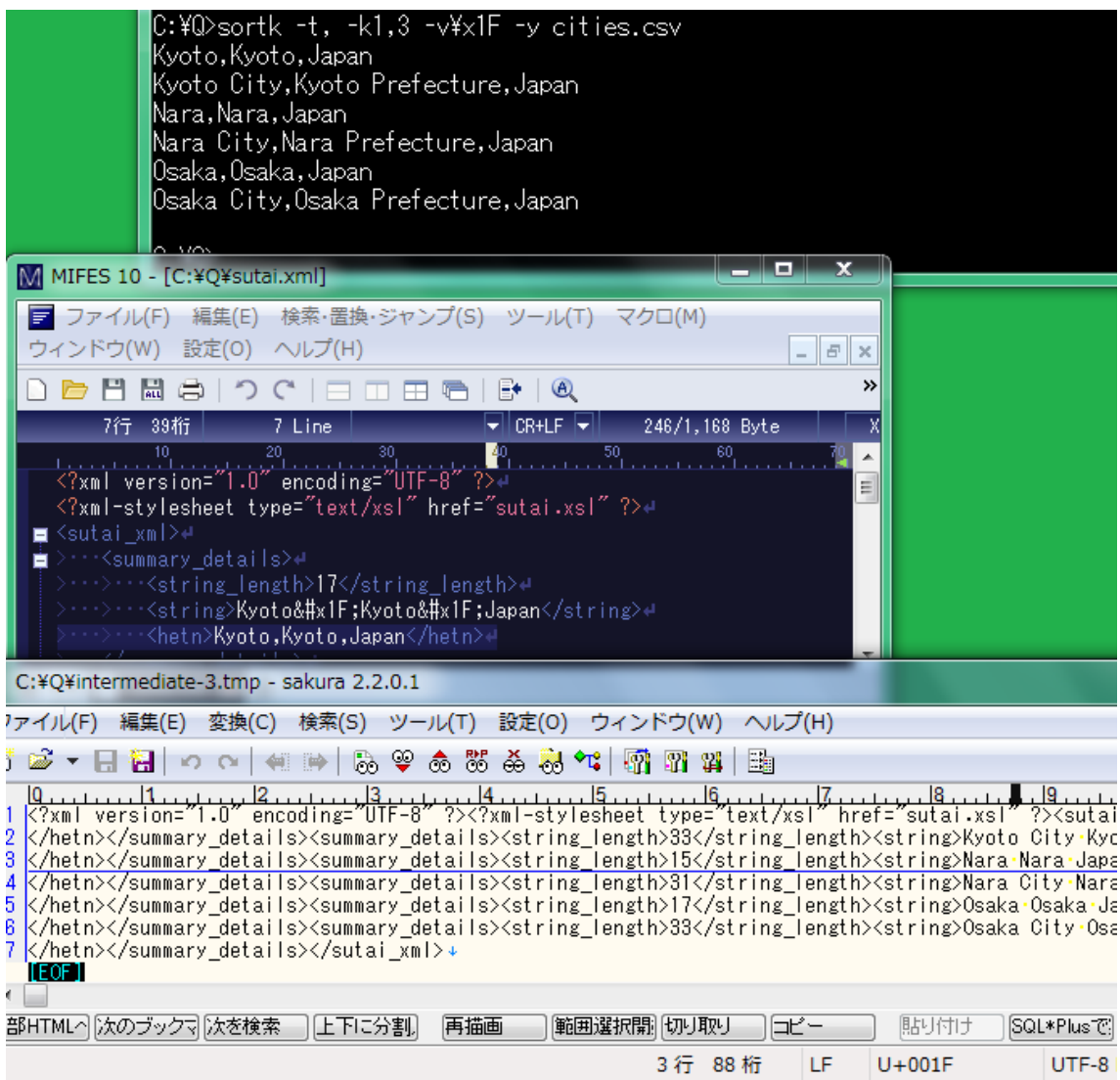
語句が短いほど上に来る

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		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(")が欄内に出現する場合、**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 wer
e 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 wer
e 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.,"~
```

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

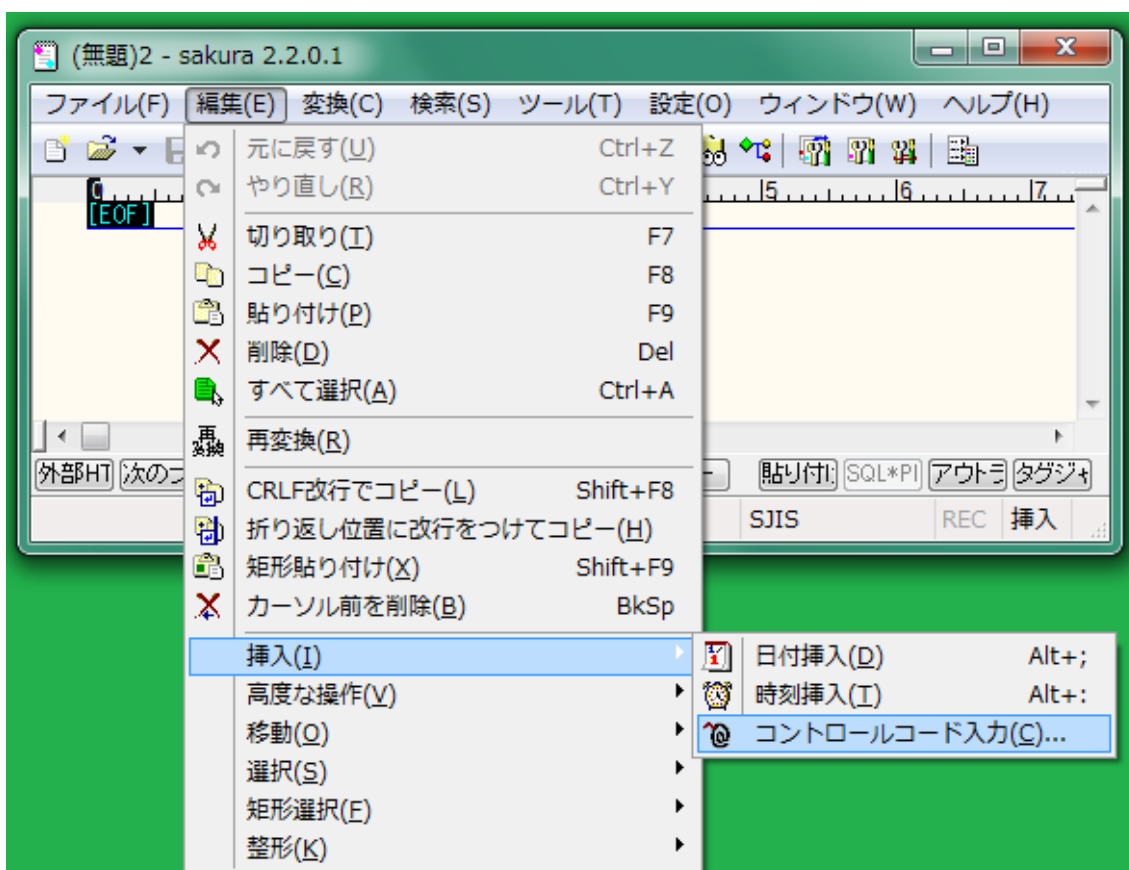
自然な表現として、引用符もコンマも sortk コマンドに無視されるべきである場合、-w~ を付加してください。CSV であるにもかかわらずコンマを無視したい範囲を~で囲んでください。

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。

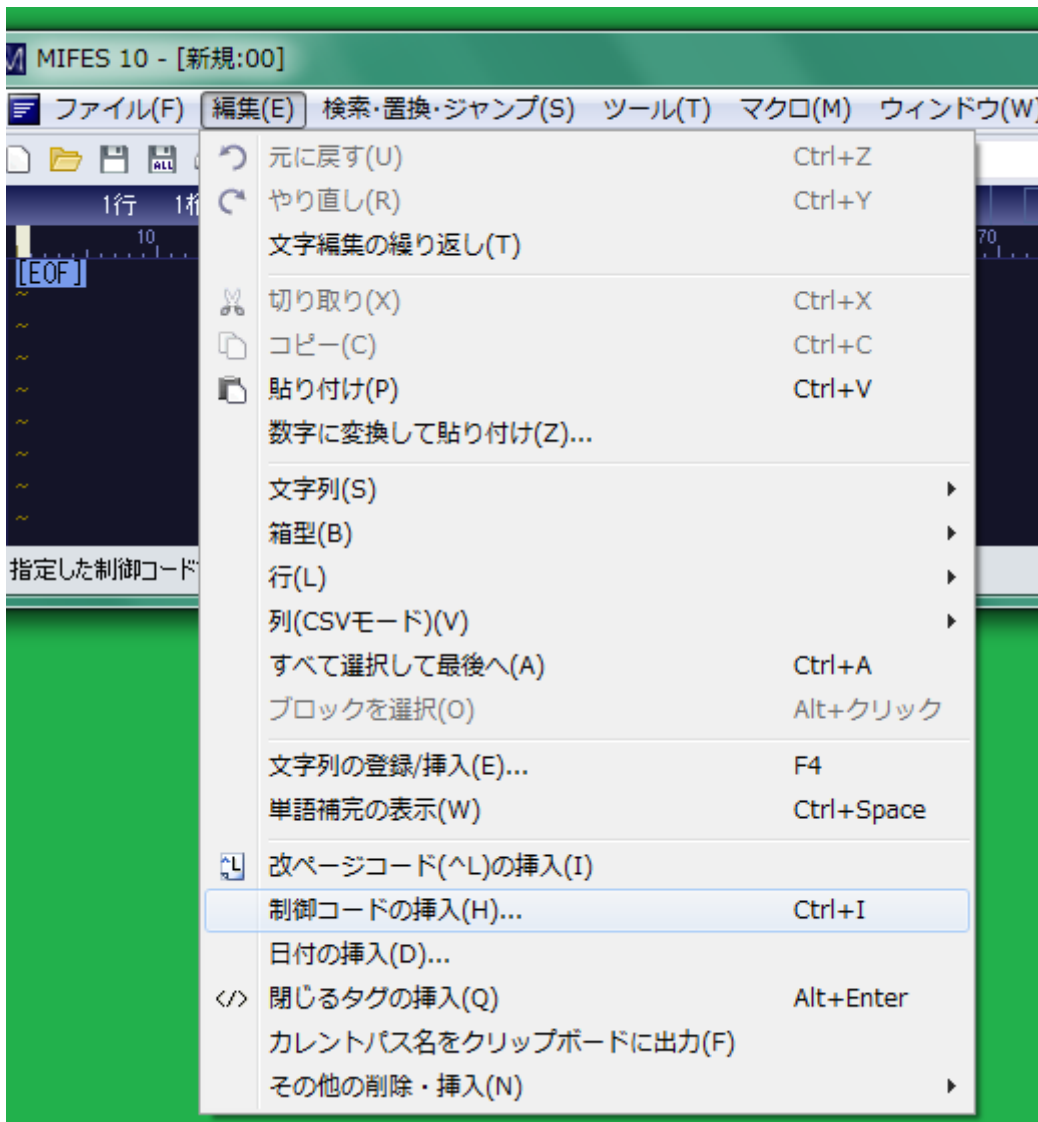
サクラエディタ

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(``)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、`` 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,
C:¥Q>
```

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(``)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 -h20
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,&#x2D;40,
4973回,41,
4967回,51,
4902回,59,
4647回,64,

C:¥Q>sortk -t, -k2,2 -n -N mini.txt | atnd -h20
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,

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,E
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
す。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>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,

C:¥Q>
```

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 -h9 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

C:¥Q>atnd -LF -p9 numbers7.csv
numbers7.csv: 5156 lines
maximum LF-to-LF span of 13 bytes (line-1000)
9,243,2664

C:¥Q>sortk -t, -k2,2 -T -o text.tmp numbers7.csv
566302 ticks

C:¥Q>sortk -t, -k2,2 -T -o text.tmp numbers7.csv
546827 ticks

C:¥Q>sortk -t, -k2,2 -T -o text.tmp numbers7.csv
522759 ticks

C:¥Q>sortk -t, -k2,2 -T -a3 -o text.tmp numbers7.csv
385532 ticks

C:¥Q>sortk -t, -k2,2 -T -a3 -o text.tmp numbers7.csv
380246 ticks

C:¥Q>sortk -t, -k2,2 -T -a3 -o text.tmp numbers7.csv
381620 ticks
```

diyo コマンドと連携する

添付の diyo コマンドは、MS932~EUC-JP.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。

バージョン番号

短い説明を表示するには、コマンドプロンプトに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。

diyo -VC

```

C:¥Q>diyo -VC
Diyo C# Version 1.03

diyo -f bars.txt      # Edit according to bars.txt
diyo -kナ -aN        # ナ is subsituted with N.
diyo -kナ -aN file    # ナ is subsituted with N to produce a binary file.
diyo -s① -e1         # ① is subsituted with 1 and represented in EUC-JP.
diyo -s① -e1 file    # ① is subsituted with 1 to produce a binary file.
diyo -s① -e1 -d      # Do not search a table.
diyo -s① -e1 -q      # Acquire a table name from setting.txt
diyo -s① -e1 -w      # -w shows which table Diyo uses.
diyo -V              # version
diyo -VC             # version with comments

C:¥Q>

```

パラメータ	密接に関係するファイル	説明
-d	MS932~EUC-JP.bin	変換表を探索してはならない。
-f	代用文字のリスト。 MS932~EUC-JP.bin	多数の代用文字にわたって変換表を編集します。
-k ナ	MS932~EUC-JP.bin	ある半角カタカナを廃止します。
-aN	MS932~EUC-JP.bin	ASCII の 1 文字で代用します。
-s①	MS932~EUC-JP.bin	ある機種依存文字を廃止します。
-e1	MS932~EUC-JP.bin	EUC-JP の 1 文字で代用します。
-w	MS932~EUC-JP.bin	どの変換表を使用するか表示します。

代用文字のリスト

事前に利用者が代用文字のリストを用意した場合、diyo コマンドは、代用文字のリストにしたがって、一度に多数の文字を編集できます。

通常文字および代用文字

diyo コマンドは、代用文字リストのファイルから 906 行まで読み込むことができます。代用文字リストのファイルの各行は、通常文字と、代用文字とからなります。利用者は、代用文字もシフト JIS で記入できます。変換表にしたがって、diyo コマンドは、シフト JIS から EUC-JP に利用者記入の文字を変換します。

ファイル名を指定

利用者が代用文字リストのファイルを用意した場合、diyo コマンドのパラメータとして、-f を付加して、空白で区切って、さらにファイル名を付加してください。

diyo -f bars.txt -w

CD コマンド

どの Windows も最初から CD コマンドを用意しています。ディレクトリを移動するには、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。Q ディレクトリから sjisEUCjpCS ディレクトリに移動する例

CD ¥sjisEUCjpCS

ctm を実行する

sjisEUCjpCS ディレクトリには、ctm.bat というバッチファイルがあります。パラメータを付加しないで ctm を実行した場合、用法が表示されます。0 を付加して ctm を実行した場合、test_files ディレクトリに保存されている古い変換表 (MS932~EUC-JP.bin) が現在のディレクトリ (sjisEUCjpCS ディレクトリ) にコピーされます。変換表は、古い変換表で上書きされます。

```
C:¥Q>CD ¥sjisEUCjpCS
C:¥sjisEUCjpCS>ctm
Enter ctm 2 to copy MS932~Shift_JIS.bin
Enter ctm 0 to copy MS932~EUC-JP.bin

C:¥sjisEUCjpCS>ctm 0
    1 個のファイルをコピーしました。
2018/12/07 13:50          31,528 MS932~EUC-JP.bin

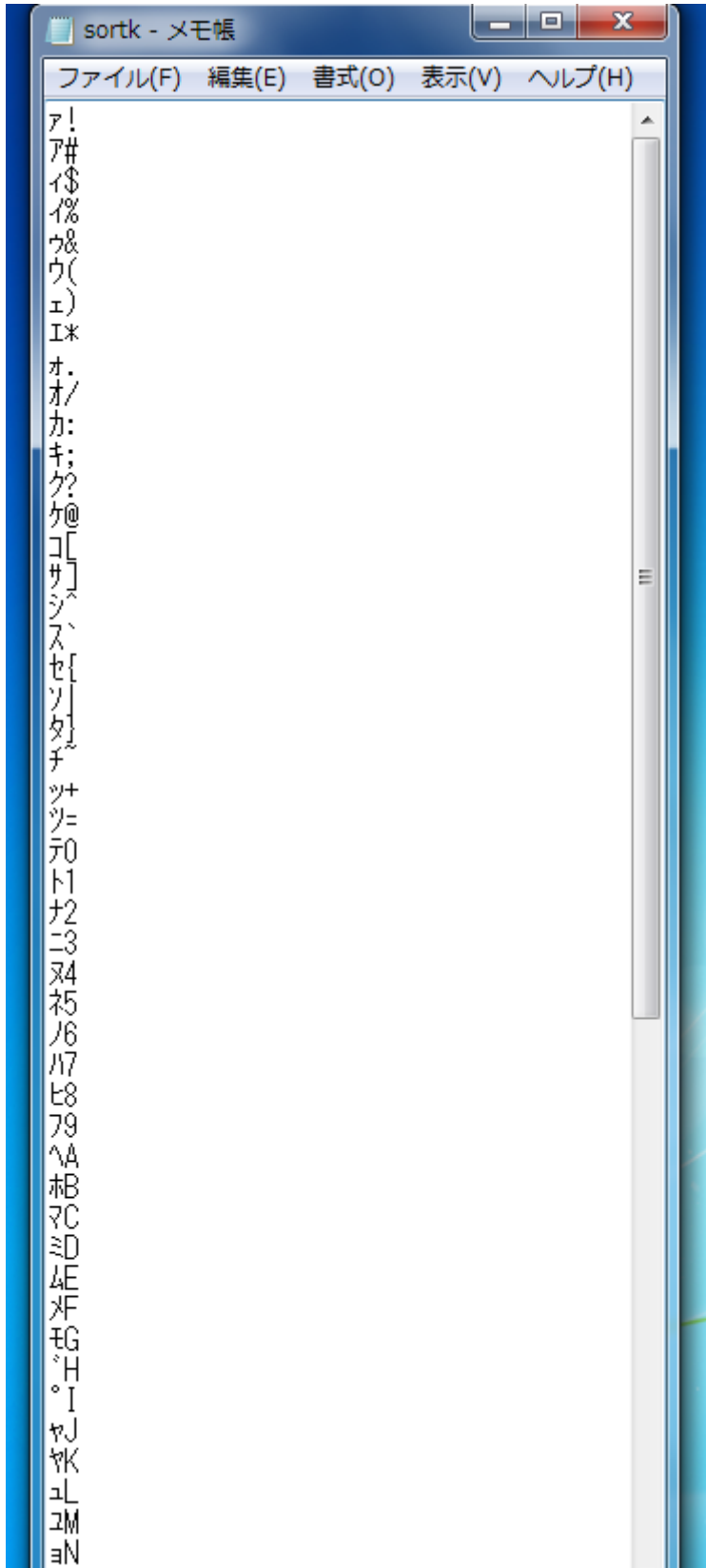
C:¥sjisEUCjpCS>CD ¥Q
C:¥Q>
```

並べ替えに適した代用文字リスト

上記の bars.txt では、ソフトウェアで並べ替えた結果が昇順になりません。代用文字リストのファイルを作成するとき、並べ替えも考慮したい。

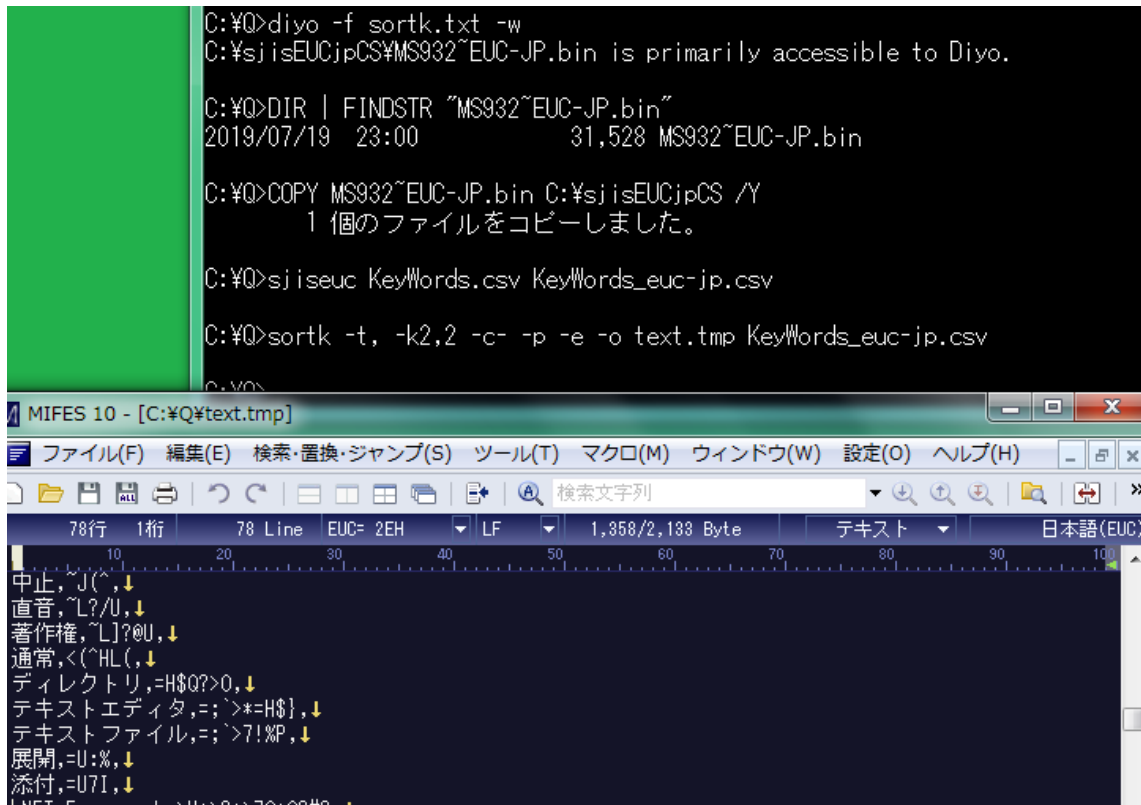
sortk コマンドに適した代用文字リスト

機種 (キシユ)、著作権 (チョサクケン) などの拗音を直音の付近に並べ、実行可能 (ジッコウカノウ)、接頭辞 (セツウジ) などの促音を直音の付近に並べ、合図 (アイズ)、英語 (エイゴ) などの濁音を清音の付近に並べ、添付 (テンプ) などの半濁音を清音の付近に並べるのに適した代用文字リストが必要です。C#版の sortk コマンドに適した代用文字リストは、sortk.txt です。半角カタカナの濁点を廃止して、H で代用します。



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。半角カタカナの代用として ASCII を使用している例を示します。sortk.txt にしたがって、diyo コマンドが変換表を編集した例(-p については、後述します)



```
C:\%Q>diyo -f sortk.txt -w
C:\%Q>DIR | FINDSTR "MS932~EUC-JP.bin"
2019/07/19 23:00          31,528 MS932~EUC-JP.bin

C:\%Q>COPY MS932~EUC-JP.bin C:\%Q\sjiseuc\ /Y
1 個のファイルをコピーしました。

C:\%Q>sjiseuc KeyWords.csv KeyWords_euc-jp.csv

C:\%Q>sortk -t, -k2,2 -c- -p -e -o text.tmp KeyWords_euc-jp.csv
```

MIFES 10 - [C:\%Q\text.tmp]

ファイル(F) 編集(E) 検索・置換・ジャンプ(S) ツール(T) マクロ(M) ウィンドウ(W) 設定(O) ヘルプ(H)

検索文字列

78行 1行 78 Line EUC= 2EH LF 1,358/2,133 Byte テキスト 日本語(EUC)

中止, J(↑, ↓)
直音, L?/U, ↓
著作権, L]?@U, ↓
通常, (^HL(, ↓
ディレクトリ, =H\$Q?>0, ↓
テキストエディタ, =;>*=H\$}, ↓
テキストファイル, =;>?!%P, ↓
展開, =U:%, ↓
添付, =U7I, ↓
LIST [Example] >U>2>79*CS#2

濁点を無視

半角カタカナは、清音のカナおよび濁点を組み合わせることで濁音を表現しますが、清音および濁音は、ほぼ同等の位置になるように並べ替えたい。上記の例のとおり、sortk コマンドが、濁点、半濁点を無視するように、sortk コマンドのパラメータとして、-c- を付加してください。diyo コマンドで、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。

新しいファイルを命名

新しいファイルを作成するには、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。text.tmp は、新しいファイルの名前です。

五十音順

まず、diyo コマンドは、sortk.txt にしたがって変換表を編集しました。次に、sjiseuc コマンドは、シフト JIS から EUC-JP にテキストファイルを変換しました。半角カ

タカナの読み方は、ASCII に置換されました。さらに、sortk コマンドは、-e が指定されたため、EUC-JP のテキストファイルを読み込んで、第 2 欄で語句を比較して行を並べ替えました。たしかに五十音順に並べ替えることができました。たとえば、ディレクトリは、テキストエディタの上に来ました。

0Ah のみで改行

0Ah のみで改行したため、MIFES の改行は、左向きではなく下向きの矢印で表現されています。

略記法

上記の例では、濁点、半濁点を無視するように、sortk コマンドのパラメータとして、-c- を付加して、半角カタカナの代用として ASCII を使用している場合、-p を付加して、EUC-JP のテキストファイルを読み込むために、-e を付加して、標準出力ではなく新しいファイルを作成して 0Ah のみで改行するために、-o を付加していましたが、略記法として、--cope は、同じ結果を生じます。すなわち、--cope は、下記のパラメータを意味します。--cope を入力して、空白で区切って、text.tmp など新しいファイルを命名してください。--cope は、EUC-JP 専用のパラメータです。ただし、--cope が指定された場合、sortk コマンドは、一時ファイルとして sutai.xml を作成しないで、行を並べ替えます。

-c- -p -e -o

パラメータ	ほぼ同等な入力	文字コード	改行	説明
--cope	-c- -p -e -o	EUC-JP	0Ah のみ	濁点、半濁点を無視
--pic	-c- -p -i	シフト JIS	0Dh, 0Ah	濁点、半濁点を無視

dayu コマンドと連携する

添付の dayu コマンドは、シフト JIS のコマンドです。dayu コマンドと sortk コマンドを連携する例については、Sortk.pdf を参照してください。

設定ファイル

diyo の設定ファイルは、setting.txt です。設定ファイルは、いわゆるシフト JIS (MS932) のファイルです。利用者がカレントディレクトリから setting.txt を削除しない限り、diyo は、setting.txt にしたがって設定します。setting.txt は、7 行からなります。第 1 行にパス名ではなくファイル名が記入されており、第 4 行の設定内容が空である例

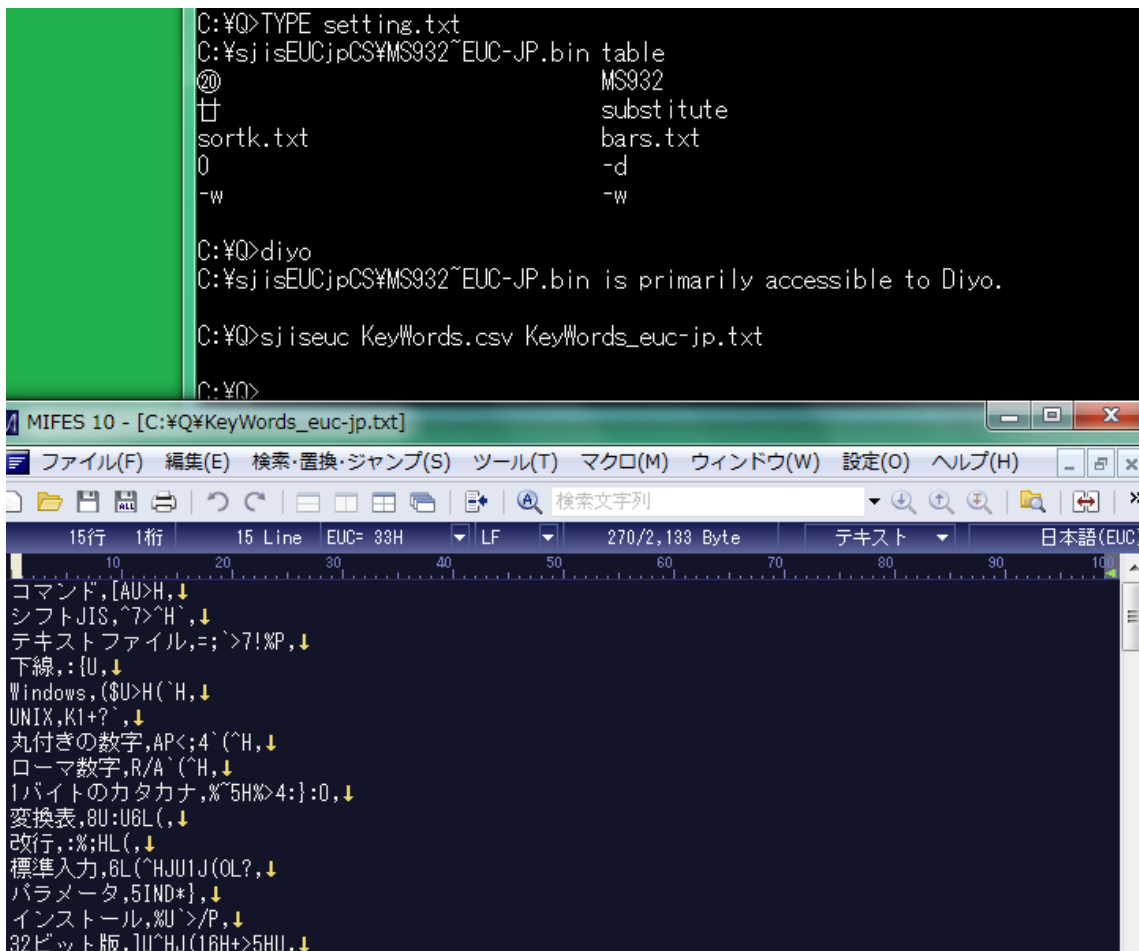
行番号	設定内容	コメント	説明
1	MS932~EUC-JP.bin	table	新しい変換表のファイル名またはパス名
2	㊾	normal	機種依存文字。通常文字
3	廿	substitute	代用文字
4		bars.txt	代用文字のリスト
5	0	-d	変換表を探索しない
6	0	-w	どの変換表を使用するか表示する
7	C:¥MS932~EUC-JP.bin		読み込む変換表のパス名

新しい変換表のパス名

diyoが setting.txt を開くことができた場合も、設定を diyo コマンドのパラメータで変更できます。コマンドがインストールされたディレクトリにある変換表を新しい変換表のパス名として記入した例


```
C:\>TYPE setting.txt
C:\>sjisEUCjpCS¥MS932~EUC-JP.bin table
@MS932
 substitute
sortk.txt bars.txt
0 -d
-w -w

C:\>diyo
C:\>sjiseuc KeyWords.csv KeyWords_euc-jp.txt
C:\>
```



変換表が上書きされる

コマンドがインストールされたディレクトリにある変換表が新しい変換表のパス名として記入されている場合、diyo コマンドは、利用者に警告しないで変換表を上書きします。

利用者が上書きする

setting.txt の第 1 行にパス名ではなくファイル名が記入されている場合、diyo コマンドは、カレントディレクトリに新しい変換表を作成します。diyo がアクセスできるディレクトリに MS932~EUC-JP.bin をコピーしてください(上書きしてください)。

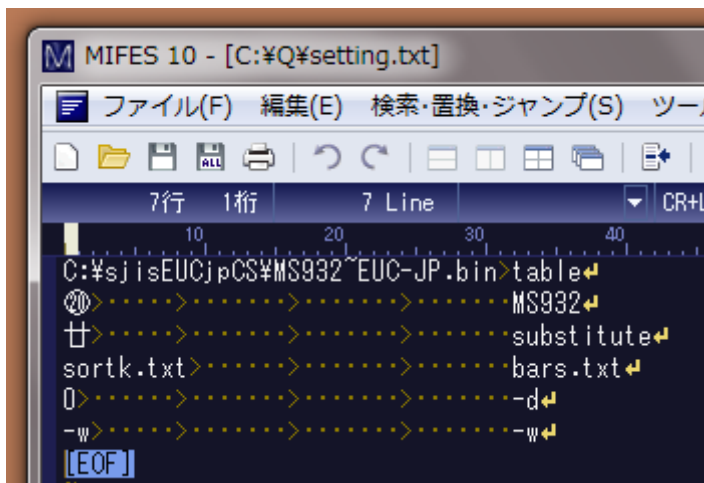
設定ファイルは必須ではない

setting.txt の第 1 行に新しい変換表のパス名またはファイル名を記入できますが、設定ファイルを開くことができない場合、diyo コマンドは、利用者に警告しないで、カレントディレクトリに新しい変換表を作成します。diyo がアクセスできるディレクトリに MS932~EUC-JP.bin をコピーしてください。

内容、タブ、コメントの順に記入

設定ファイルは、いわゆるシフト JIS (MS932) のファイルです。設定内容を左に詰めて記入してください。設定内容のみで改行できますが、シングルクォーツ、縦線、タブの

次にコメントを記入できます。Perl, PHP, Python, Ruby などのスクリプト言語とは異なり、diyo は、シャープ(#)の次にコメントを記入できません。空白(20h)の次にコメントを記入できません。空白ではなくタブを利用してください。Windows のメモ帳では、タブが機能しますが空白に見えます。MIFES で、タブが大なり記号(>)で表示されている例



0 ではないならば有効

第 5 行、第 6 行に 0 を記入すると無効になります。どの変換表を使用するか表示したくないなら第 6 行に 0 を記入してください。

第 4 行の設定内容が空である

代用文字リストのファイル名が空である場合、多数の代用文字にわたって変換表を編集する機能は、無効になるため、第 2 行および第 3 行にしたがって、1 文字を編集します。

第 4 行に sortk.txt を記入

代用文字リストのファイル名が空ではない場合、第 4 行にしたがって、多数の代用文字にわたって変換表を編集します。

五十音順

下記の例では、第 4 行に sortk.txt を記入しました。まず、diyo コマンドは、sortk.txt にしたがって変換表を編集しました。次に、sjiseuc コマンドは、シフト JIS から EUC-JP にテキストファイルを変換しました。半角カタカナの読み方は、ASCII に置換されました。さらに、sortk コマンドは、第 2 欄で語句を比較して行を並べ替えました。たしかに五十音順に並べ替えることができました。たとえば、ディレクトリは、テキストエディタの上に来ました。添付の narfi コマンドは、[第 2 欄を削除](#)できます。

```
C:\%Q>TYPE setting.txt
C:\%sjisEUCjpcS¥MS932~EUC-JP.bin table
② MS932
廿 substitute
sortk.txt bars.txt
0 -d
-w -w

C:\%Q>diyo
C:\%sjisEUCjpcS¥MS932~EUC-JP.bin is primarily accessible to Diyo.

C:\%Q>sjiseuc KeyWords.csv KeyWords_euc-jp.txt

C:\%Q>sortk -t, -k2,2 --cope text.tmp KeyWords_euc-jp.txt
C:\%Q>
```

```
MIFES 10 - [C:\%Q\text.tmp]
ファイル(F) 編集(E) 検索・置換・ジャンプ(S) ツール(T) マクロ(M) ウィンドウ(W) 設定(O) ヘルプ(H)
検索文字列
78行 1桁 78 Line EUC= 2EH LF 1,358/2,133 Byte テキスト 日本語(EUC)
濁点, }H?=U, ↓
中止, ~J(^, ↓
直音, ~L?/U, ↓
著作権, ~L]?@U, ↓
通常, <(^HL(, ↓
ディレクトリ, =H$Q?>0, ↓
テキストエディタ, =; >*=H$}, ↓
テキストファイル, =; >?!%P, ↓
展開, =U:%, ↓
添付, =U7I, ↓
.NET Framework, >H+>3+>70*CS#?, ↓
並べ替え, ON8H:*, ↓
```

ID として行番号を使用する

2 行以上が同じ要約を生じても順序が確定するように、要約に ID を付与できます。sortk コマンドは、ID として行番号を使用します。要約も ID も標準出力に書き出されません。

どの行も唯一だ

利用者が、比較対象として欄をどう選択しても、行番号が異なるため、比較対象は、どれも唯一です。下記の例では、まず、-#を付加しない sortk コマンドを実行しました。次に、要約に ID を付与するために-#を付加して sortk コマンドを実行しました。第 10 欄から第 20 欄まで比較して順序が確定しない場合も、行番号で順序が確定します。

```

C:¥Q>sortk -t, -k10,20 Field-22.csv | atnd -h23
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,07,12,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,08,8,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,10,13,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,1,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,15,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,26,17,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,27,10,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,29,18,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,44,11,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,49,7,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,51,2,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,32,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,26,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,28,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,31,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,29,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,27,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,30,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,35,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,38,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,36,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,39,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,33,

C:¥Q>sortk -t, -k10,20 -# Field-22.csv | atnd -h23
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,07,12,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,08,8,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,10,13,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,1,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,22,15,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,26,17,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,27,10,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,29,18,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,44,11,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,49,7,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,51,2,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,19,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,20,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,21,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,22,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,23,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,24,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,25,
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,59,26,

```

要約ごとに1行を表示

ある要約が他の要約と合致した場合、その要約は、1行のみ表示するには、sortk コ

マンドのパラメータとして-u を付加してください。

sutai.xml を再利用する

sortk コマンドは、いったん sutai.xml という XML ファイルを作成して、sutai.xml を読み込んで、要約および詳細のブロックを並べ替えて、標準出力に詳細のみ書き出します。2 行以上が同じ要約を生じても順序が確定するように、要約に ID を付与できます。sortk コマンドは、ID として行番号を使用します。要約も ID も標準出力に書き出されません。

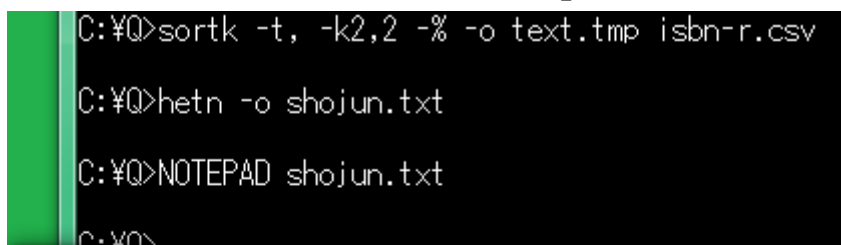
sutai.xml が作成されない例

シフト JIS のテキストであることが明白であり、制御文字を削除するために、-i が付加された sortk コマンドは、sutai.xml を作成しません。EUC-JP のテキストを読み込んで、半角カタカナの濁点を無視して並べ替えるために、--cope を付加した sortk コマンドは、sutai.xml を作成しません。

設定ファイル

添付の hetn コマンドは、sutai.xml を再利用できます。hetn コマンドの設定ファイルを書き出すには、sortk コマンドのパラメータとして-% を付加してください。

```
sortk -t, -k2,2 -% -o text.tmp isbn-r.csv
```

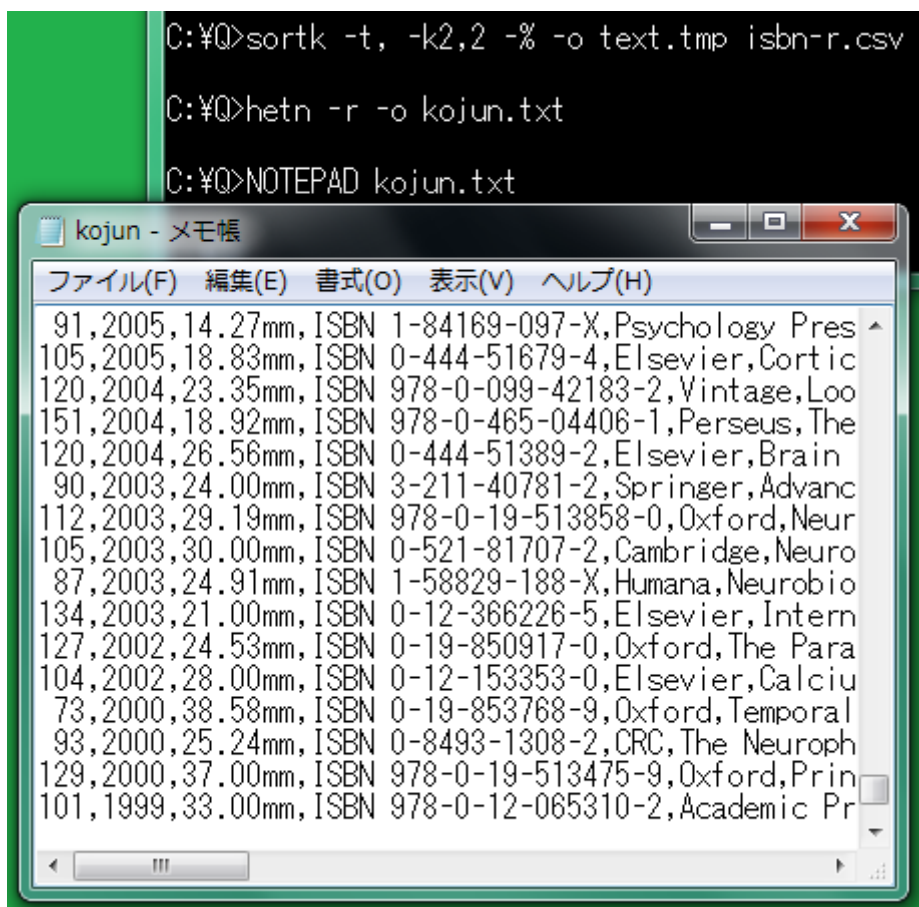


```
C:¥Q>sortk -t, -k2,2 -% -o text.tmp isbn-r.csv
C:¥Q>hetn -o shojun.txt
C:¥Q>NOTEPAD shojun.txt
C:¥Q>
```



降順

hetn コマンドの設定ファイルは、manager.txt です。manager.txt にしたがって、hetn コマンドは、sutai.xml を読み込んで、行を並べ替えます。下記の例では、降順に並べ替えるために、hetn コマンドのパラメータとして-r を付加しました。



細長い表を作成

添付の narfi コマンドは、CSV のテキストファイルを読み込んで、選択された欄を新しいファイルに書き出します。narfi コマンドは、幅が広い表のテキストファイルを細長い表のファイルに縮小できます。

読み方の欄を削除

もう五十音順に並べ替えることができたので読み方の欄を削除したいならば、narfi コマンドのパラメータとして、たとえば、-b101 を付加してください。添付の rufeol コマンドは、[行末からコンマを削除](#)できます。

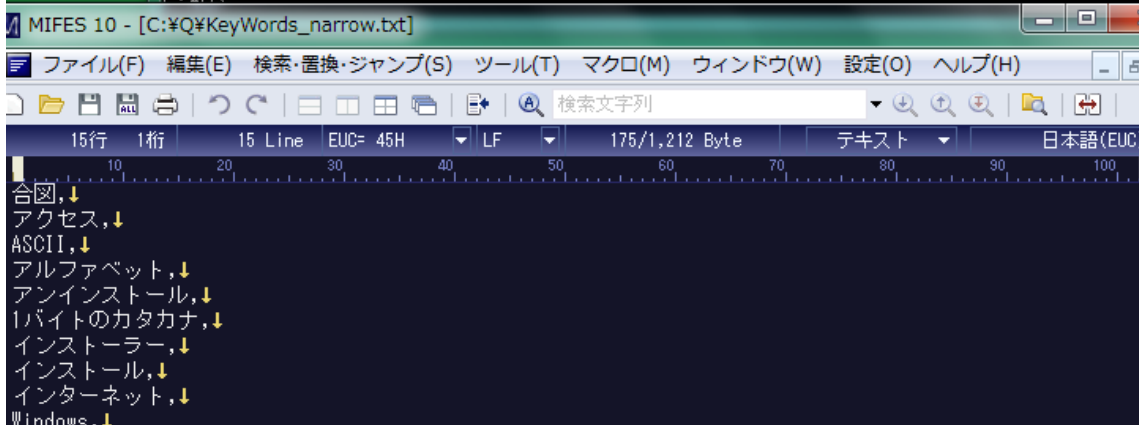
```
C:\Q>TYPE setting.txt
C:\$jsjisEUCjpCS$MS932~EUC-JP.bin table
@ MS932
廿 substitute
sortk.txt bars.txt
0 -d
-w -w

C:\Q>diyo
C:\$jsjisEUCjpCS$MS932~EUC-JP.bin is primarily accessible to Diyo.

C:\Q>sjiseuc KeyWords.csv KeyWords_euc-jp.txt

C:\Q>sortk -t, -k2,2 --cope text.tmp KeyWords_euc-jp.txt

C:\Q>narfi -c3 -b101 -u text.tmp KeyWords_narrow.txt
C:\Q>
```



短い表

シフト JIS のファイルの最初から 5 行を表示するには、atnd コマンドのパラメータとして、-h5 を付加してください。-h5 は、-b4 -p5 を意味します。リダイレクトすることで、短い CSV ファイルを作成できます。

細長い表

narfi コマンドは、幅が広いテキストファイルを細長いファイルに縮小できます。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、narfi コマンドのパラメータとして、-c5 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、narfi コマンドのパラメータとして、-b10111 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 欄を書き出します。すなわち、4 列の表になります。ただし、-b10111 が指定されない場合、narfi コマンドは、-b11111 が指定されたとみなして、第 2 欄も書き出します。

```

C:¥Q>atnd -h6 Field-21.csv
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
85,52,86,48,68,22,59,44,63,61,92,37,09,83,55,58,51,31,60,02,
18,53,10,36,22,69,60,98,02,20,39,25,29,44,99,60,60,26,03,73,
93,15,38,47,92,70,49,02,38,69,29,09,21,27,86,89,73,25,31,47,
32,43,69,31,69,38,13,47,18,60,16,13,30,66,02,80,88,30,92,30,
99,29,69,42,17,87,81,79,28,21,72,26,60,97,49,60,99,14,52,66,

C:¥Q>narfi -c5 -b10111 Field-21.csv Field-5.csv

C:¥Q>atnd -h6 Field-5.csv
1,3,4,5,
85,86,48,68,
18,10,36,22,
93,38,47,92,
32,69,31,69,
99,69,42,17,

C:¥Q>

```

-b10111					
-b	1	0	1	1	1
真偽	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

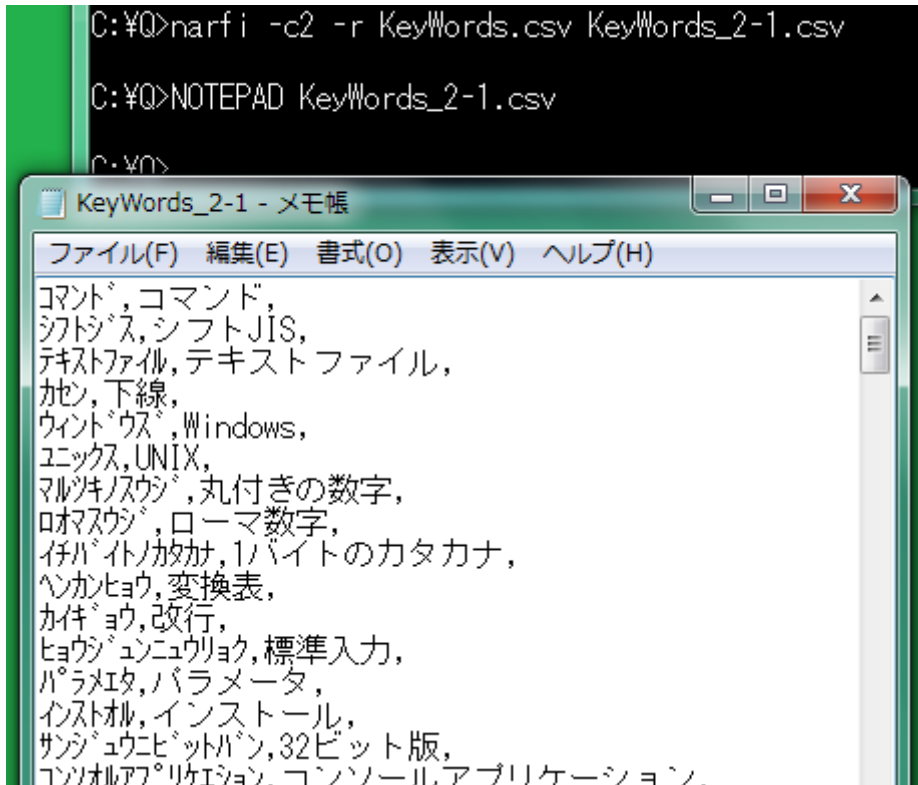
新しいファイルを命名する

広い表のテキストファイル、細長い表のテキストファイルがこの順に指定された場合、narfi コマンドは、広い表のテキストファイルから行を読み込んで、細長い表のテキストファイルに行を書き出します。細長い表のファイル名が指定されていない場合、narfi コマンドは、narfi.txt に行を書き出します。

第 1 欄と第 2 欄を入れ替える

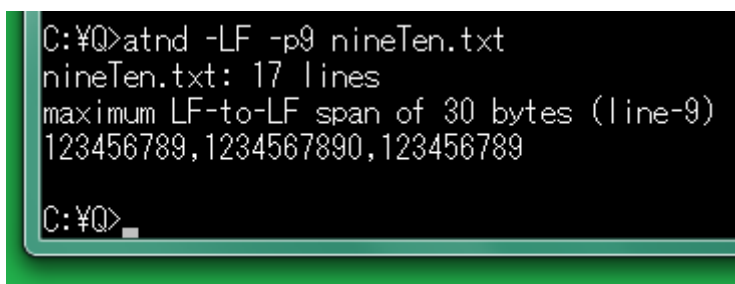
キーワードおよび読み方の CSV は、2 欄からなります(第 3 欄は、空であるため、実質的に 2 欄しかありません)。各行で、欄を逆順に並べ替えるには、narfi コマンドのパラメータとして -r を付加してください。表が 2 欄からなる場合、第 1 欄と第 2 欄を入れ替えることができます。

narfi -c2 -r Keywords.csv



ファイルの行数

添付の `atnd` コマンドは、ファイルの行数をカウントできます。ファイルの行数を表示するには、`atnd` コマンドのパラメータとして、`-LF` を付加してください。ファイル名および行数が表示されます。また、最長の行の長さが記録されており、バイト数および行番号が表示されます。さらに、第1行が表示されます。第1行ではなく第9行を表示するには、`atnd` コマンドのパラメータとして、`-p9` を付加してください。



UNIX の `tail` コマンドに似ている

ファイルの終了部分にある5行を表示するには、`atnd` コマンドのパラメータとして、`-t5` を付加してください。`atnd` コマンドは、自動的にファイルの行数をカウントします。たとえば、1024行である場合、`-t5` は、`-b4 -p1024` を意味します。`atnd -t` は、UNIX の `tail` コマンドに似ています。

`atnd -t5 LineNo.txt`

```

C:¥Q>atnd -LF LineNo.txt
LineNo.txt: 1024 line
maximum LF-to-LF span of 34 bytes (line-1000)
Line-1: 639 lines to Line-640

C:¥Q>atnd -b4 -p1024 LineNo.txt
Line-1020: 380 lines from Line-640
Line-1021: 381 lines from Line-640
Line-1022: 382 lines from Line-640
Line-1023: 383 lines from Line-640
Line-1024: 384 lines from Line-640

C:¥Q>atnd -t5 LineNo.txt
Line-1020: 380 lines from Line-640
Line-1021: 381 lines from Line-640
Line-1022: 382 lines from Line-640
Line-1023: 383 lines from Line-640
Line-1024: 384 lines from Line-640

C:¥Q>

```

短い説明

atnd コマンドに関する短い説明を表示するには、atnd コマンドのパラメータとして、`-VC` を付加してください。

```

C:¥Q>atnd -VC
Atnd C# Version 1.03

atnd -a2 -p567 file           # 2 lines after line-567
atnd -b3 -p567 file           # 3 lines before line-567
atnd -c4 -p567 file           # 4 lines before line-567 and 4 lines after line-567
atnd -c4 -p567 -n file        # A number appears at the beginning of each line.
atnd -h8 file                 # Line-1, Line-2, Line-3, ..., Line-8
atnd -h8 < file               # Line-1, Line-2, Line-3, ..., Line-8
atnd -i90-123 file           # Line-90, Line-91, Line-92, ..., Line-123
atnd -LF -p567 file           # Count lines. Record the max span from LF to LF.
atnd -LF -s¥x2C -p567 file    # Record the max span from separator to separator.
atnd -LF -s, -p567 file       # Record the max span from separator to separator.
atnd -LF -s, -f file          # Check if all lines share the field count.
atnd -p567 file               # line-567
atnd -p567 -8 file            # UTF-8
atnd -p567 -e file            # EUC-JP
atnd -t9 file                 # 9 lines before EOF
atnd -V                       # version
atnd -VC                      # version with comments
DIR | atnd -LF --DIR -p9      # Display a label (DIR) instead of files.

C:¥Q>

```

パラメータ	ほぼ同等な入力	説明
-8		UTF-8 のテキストを読み込む。
-a2		後続の 2 行を表示する。
-b3		先行する 3 行を表示する。
-c4		先行する 4 行および後続の 4 行を表示する。
-e		EUC-JP のテキストを読み込む。
-f		どの行も同じ欄数を有しているか点検する。
-h8	-b7 -p8	第 1 行から第 8 行まで表示する。
-i90-123	-b33 -p123 -p90 -a33	第 90 行から第 123 行まで表示する。
-LF		行数をカウントする。最長の行のバイト数を記録する。
-LF ==DIR		ラベルとして DIR を表示する。
-LF -s,		コマンドうしの間にある欄の長さの最大値を記録する。
-LF -s¥x7C		2 桁の十六進数で区切り文字を指定する。
-n		行の先頭に番号を表示する。
-p567		第 567 行を表示する。
-t5	-b4 -p 行数	ファイルの終わりにある 5 行を表示する。
-V		バージョン番号を表示する。
-VC		バージョン番号、短い説明を表示する。

標準入力から読み込む

ファイル名が指定されていない場合、atnd コマンドは、標準入力(stdin)から読み込みます。利用者は、情報源(標準入力)をラベルとして命名できます。下記の例では、DIR というラベルを入力しました。

ファイルにコマンドの出力をリダイレクト

下記の例では、まず、DIR コマンドの出力を画面から一時ファイル(text.tmp)に、リダイレクトしました。次に、ファイルの行数をカウントするために-LF を付加して atnd コマンドを実行しました。ファイル名として、text.tmp が表示されました。

ファイル名ではなく stdin を表示

一時ファイルに、リダイレクトするのではなく、DIR コマンドの出力を標準入力として利用できます。すなわち、DIR コマンドの出力を atnd コマンドに入力するパイプを構成しました。情報源として、ファイル名ではなく stdin が表示されました。

stdin ではなく DIR を表示

最後に、stdin ではなく DIR が表示されるように、==DIR を付加した atnd コマンドに

よるパイプを構成しました。

```
C:¥s jisEUCjpCS>DIR *.txt > text.tmp

C:¥s jisEUCjpCS>atnd -LF -p11 text.tmp
text.tmp: 16 lines
maximum LF-to-LF span of 67 bytes (line-16)
2019/07/19 23:36          111 setting.txt

C:¥s jisEUCjpCS>DIR *.txt | atnd -LF -p11
stdin: 16 lines
maximum LF-to-LF span of 67 bytes (line-16)
2019/07/19 23:36          111 setting.txt

C:¥s jisEUCjpCS>DIR *.txt | atnd -LF -p11 -=DIR
DIR: 16 lines
maximum LF-to-LF span of 67 bytes (line-16)
2019/07/19 23:36          111 setting.txt

C:¥s jisEUCjpCS>_
```

欄を編集する

narfi コマンドは、ある欄から負号、数字、小数点を書き出すことができます。

浮動小数点数を抽出

たとえば、第 2 欄で、小数点を発見して、その欄から浮動小数点数のみ書き出すには、narfi コマンドのパラメータとして、`-e2` を付加してください。

```
narfi -e2 isbn-t.csv isbn-t_decimal.csv
```

```
C:¥Q>atnd -t5 isbn-t.csv
2013年,厚さ42.49mm,ISBN 978-1-4614-3902-8,Springer,Adenosine,,By,Susan Masino,
2011年,厚さ12.74mm,ISBN 978-0-7377-5254-0,GALE,Migraines,,By,Mary E. Williams,
2010年,厚さ21.77mm,ISBN 978-0-262-51311-1,MIT,What Is Addiction?,,By,Don Ross,
2000年,厚さ38.58mm,ISBN 0-19-853768-9,Oxford,Temporal Logic,,By,Dov M. Gabbay,
2014年,厚さ21.00mm,ISBN 978-0-444-63425-2,Elsevier,Dopamine,,By,Marco Diana,

C:¥Q>narfi -e2 isbn-t.csv isbn-t_decimal.csv

C:¥Q>atnd -t5 isbn-t_decimal.csv
2013年,42.49,ISBN 978-1-4614-3902-8,Springer,Adenosine,,By,Susan Masino,
2011年,12.74,ISBN 978-0-7377-5254-0,GALE,Migraines,,By,Mary E. Williams,
2010年,21.77,ISBN 978-0-262-51311-1,MIT,What Is Addiction?,,By,Don Ross,
2000年,38.58,ISBN 0-19-853768-9,Oxford,Temporal Logic,,By,Dov M. Gabbay,
2014年,21.00,ISBN 978-0-444-63425-2,Elsevier,Dopamine,,By,Marco Diana,

C:¥Q>_
```

小数点が発見されない

浮動小数点数を抽出するために-e2が付加されたnarfi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。整数は、小数点がありません。整数を抽出するには、-e2ではなく-n2を付加してください。第1欄から整数を抽出するには、-n1を付加してください。

```
C:¥Q>atnd -t5 isbn-t.csv
2013年,厚さ42.49mm,ISBN 978-1-4614-3902-8,Springer,Adenosine,,By,Susan Masino,
2011年,厚さ12.74mm,ISBN 978-0-7377-5254-0,GALE,Migraines,,By,Mary E. Williams,
2010年,厚さ21.77mm,ISBN 978-0-262-51311-1,MIT,What Is Addiction?,,By,Don Ross,
2000年,厚さ38.58mm,ISBN 0-19-853768-9,Oxford,Temporal Logic,,By,Dov M. Gabbay,
2014年,厚さ21.00mm,ISBN 978-0-444-63425-2,Elsevier,Dopamine,,By,Marco Diana,

C:¥Q>narfi -n1 isbn-t.csv isbn-t_years.csv

C:¥Q>atnd -t5 isbn-t_years.csv
2013,厚さ42.49mm,ISBN 978-1-4614-3902-8,Springer,Adenosine,,By,Susan Masino,
2011,厚さ12.74mm,ISBN 978-0-7377-5254-0,GALE,Migraines,,By,Mary E. Williams,
2010,厚さ21.77mm,ISBN 978-0-262-51311-1,MIT,What Is Addiction?,,By,Don Ross,
2000,厚さ38.58mm,ISBN 0-19-853768-9,Oxford,Temporal Logic,,By,Dov M. Gabbay,
2014,厚さ21.00mm,ISBN 978-0-444-63425-2,Elsevier,Dopamine,,By,Marco Diana,

C:¥Q>
```

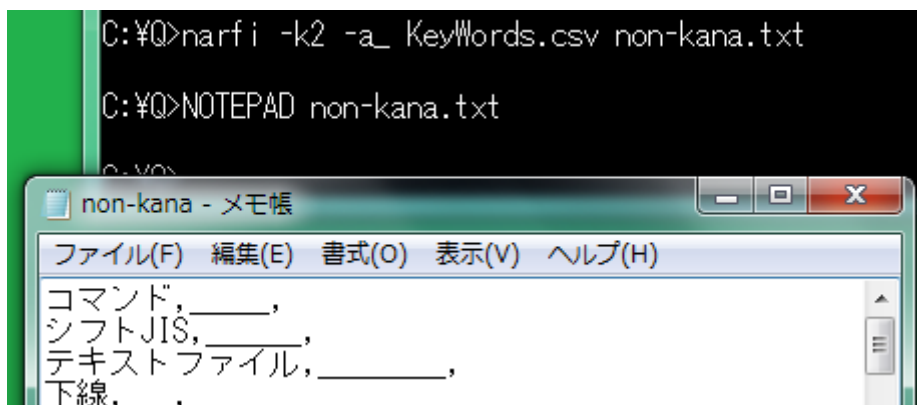
数字に到達するまで無視される

浮動小数点数を抽出するために、たとえば、-e2が付加されたnarfiコマンドは、第2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、たとえば、-n1が付加されたnarfiコマンドは、第1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。

半角カタカナを廃止

第2欄で、半角カタカナを廃止して、ASCIIの下線で代用するには、narfiコマンドのパラメータとして、-k2 -a_を付加してください。

```
narfi -k2 -a_ Keywords.csv non-kana.txt
```

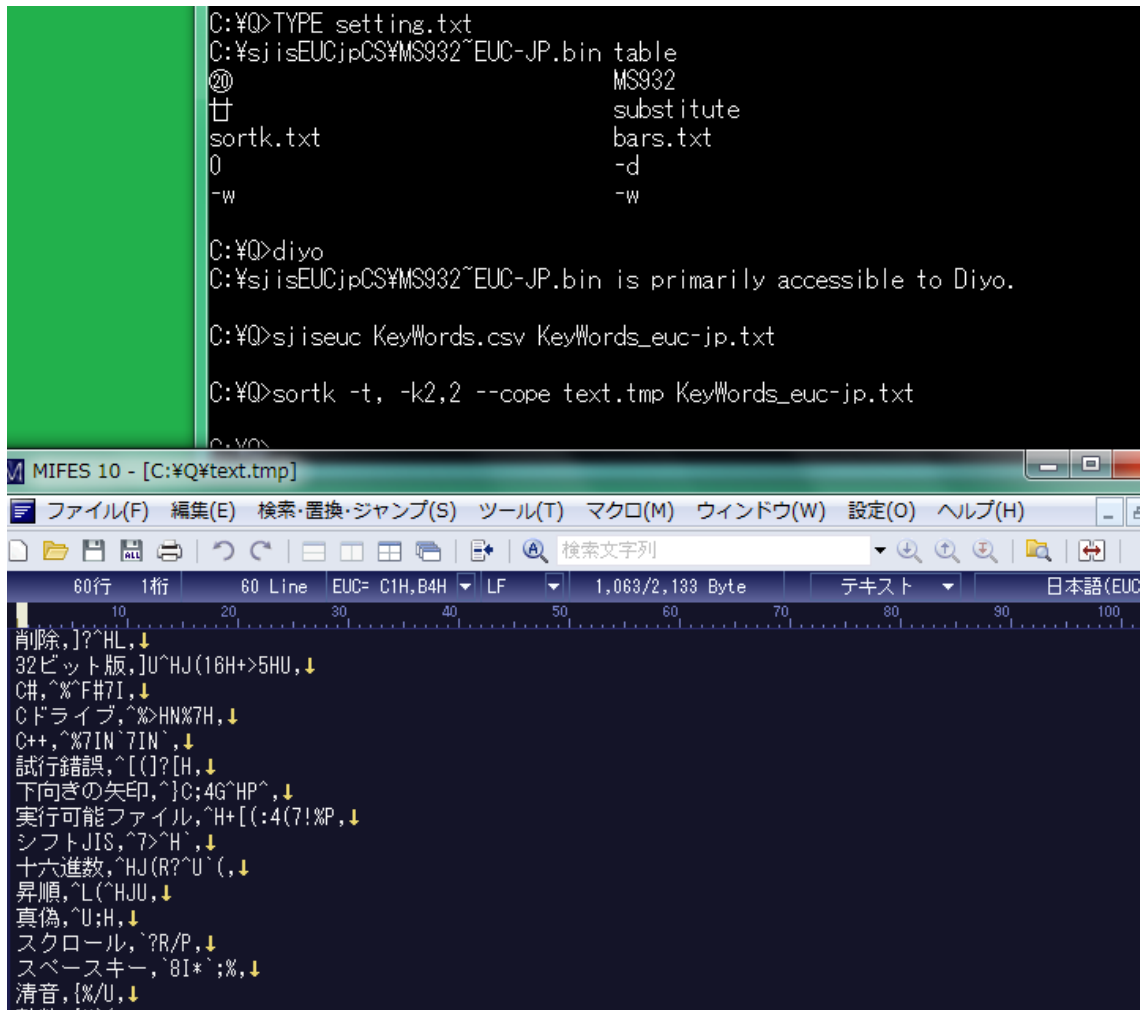


ディレクトリが上に来る

sortk.txt にしたがって、変換表を編集した場合、ディレクトリは、テキストエディタの上に来ます。

実行可能ファイルが上に来る

sortk.txt にしたがって、変換表を編集した場合、もうひとつの例として、実行可能ファイルは、シフト JIS の上に来ます。



```
C:\%Q>TYPE setting.txt
C:\%sjisEUCjpCS%MS932~EUC-JP.bin table
㊦ MS932
㊦ substitute
sortk.txt bars.txt
0 -d
-w -w

C:\%Q>diyo
C:\%sjisEUCjpCS%MS932~EUC-JP.bin is primarily accessible to Diyo.

C:\%Q>sjiseuc KeyWords.csv KeyWords_euc-jp.txt

C:\%Q>sortk -t, -k2,2 --cope text.tmp KeyWords_euc-jp.txt
C:\%Q>
```

MIFES 10 - [C:\%Q%\text.tmp]

ファイル(F) 編集(E) 検索・置換・ジャンプ(S) ツール(T) マクロ(M) ウィンドウ(W) 設定(O) ヘルプ(H)

検索文字列

60行 1桁 60 Line EUC= C1H,B4H LF 1,063/2,133 Byte テキスト 日本語(EUC)

```
削除, ]?^HL, ↓
32ビット版, ]U^HJ(16H+>5HU, ↓
C#, ^%F#7I, ↓
Cドライブ, ^%HN%7H, ↓
C++, ^%7IN`7IN`, ↓
試行錯誤, ^([]?[H], ↓
下向きの矢印, ^}C;4G^HP^, ↓
実行可能ファイル, ^H+[(;4(7!%P, ↓
シフトJIS, ^7>^H, ↓
十六進数, ^HJ(R?^U`(), ↓
昇順, ^L(^HJU, ↓
真偽, ^U;H, ↓
スクロール, `?R/P, ↓
スペースキー, `8I*`;%, ↓
清音, [%/U, ↓
```

制御文字を削除

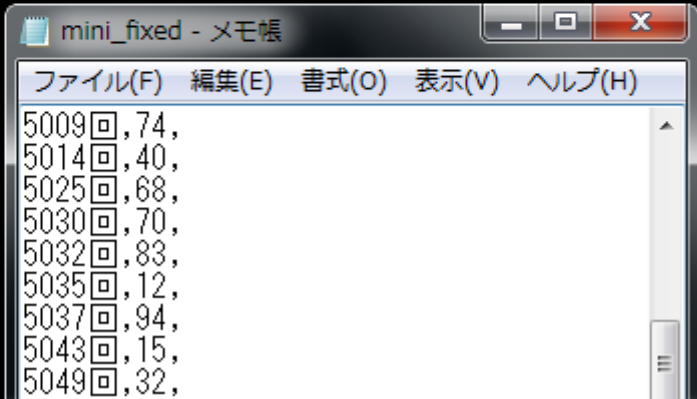
下記の例では、mini.txt という CSV にタブ、改ページ、エスケープシーケンスが含まれています。第 2 欄から制御文字を削除するには、narfi コマンドのパラメータとして、-o2 を付加してください。

```
narfi -o2 mini.txt mini_fixed.txt
```

```
C:\Windows\system32\cmd.exe
C:\>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:\>narfi -o2 mini.txt mini_fixed.txt

C:\>TYPE mini_fixed.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,
```



FC コマンドで確認

上記の `mini_fixed.txt` と元の `mini.txt` を FC コマンドで比較してみました。

```
C:\YQ>FC mini.txt mini_fixed.txt
ファイル mini.txt と MINI_FIXED.TXT を比較しています
***** mini.txt
4851回,38,
4881回,86,
4902回,59,
***** MINI_FIXED.TXT
4851回,38,
4881回,86,
4902回,59,
*****

***** mini.txt
5009回,74,
5014回,40,
5025回,68,
***** MINI_FIXED.TXT
5009回,74,
5014回,40,
5025回,68,
*****

***** mini.txt
5035回,12,
5037回,94,
5043回,15,
***** MINI_FIXED.TXT
5035回,12,
5037回,94,
5043回,15,
*****

C:\YQ>
```

半角カタカナの濁点を削除

シフト JIS であることが明白である場合、たとえば、第 2 欄から半角カタカナを削除するには、`narfi` コマンドのパラメータとして、`-d2` を付加してください。

濁点の代用文字を削除

`sortk.txt` にしたがって、半角カタカナの代用文字を使用している場合、濁点の代用文字を削除するには、`narfi` コマンドのパラメータとして、さらに `-p` を付加してください。下記の例では、添付の `dayu` コマンドで半角カタカナを ASCII で代用しています。`dayu` コ

マンドについては、Sortk.pdf を参照してください。narfi コマンドは、第 2 欄のみ編集できます。-d2 -p の効果として、H も I も削除されたことがわかります。なお、-d が付加された narfi コマンドは、濁点、半濁点、それらの代用文字を削除しますし、制御文字も削除します。

```
C:\>dayu -f sortk.txt -r- -w
C:\>sortkCS\MS932~Shift_JIS.bin is primarily accessible to Dayu.

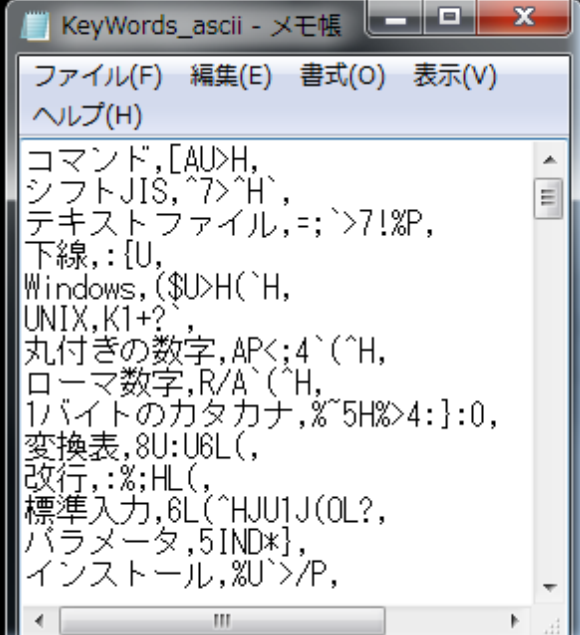
C:\>DIR | FINDSTR "MS932~Shift_JIS.bin"
2019/07/27 15:18          31,528 MS932~Shift_JIS.bin

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

C:\>dayu -e- KeyWords.csv KeyWords_ascii.csv

C:\>narfi -c3 -d2 -p KeyWords_ascii.csv KeyWords_omitted.csv

C:\>NOTEPAD KeyWords_ascii.csv
C:\>MORE KeyWords_omitted.csv
コマンド,[AU>H,
シフトJIS,^7>^H`,
テキストファイル,=;`>7!%P,
下線,:[U,
Windows,($U>H(`H,
UNIX,K1+?`,
丸付きの数字,AP<;4`(^H,
ローマ数字,R/A`(^H,
1バイトのカタカナ,%~5H%>4:}:0,
変換表,8U:U6L(,
改行,:%;HL(,
標準入力,6L(^HJU1J(OL?,
パラメータ,5IND*},
インストール,%U`>/P,
```



バージョン番号

バージョン番号を表示するには、narfi コマンドのパラメータとして、-V を付加してください。

実行可能ファイルが上に来る

narfi コマンドに関する短い説明を表示するには、narfi コマンドのパラメータとして、-VC を付加してください。

パラメータ	ほぼ同等な入力	説明
-8		UTF-8 のテキストを読み込む
-a_	-a¥x5F	ASCII の 1 バイト文字として、下線を使用する。
-b1011		4 欄のうち、第 1 欄、第 3 欄、第 4 欄を選択する。
-c4	-b11111000000	第 1 欄から第 4 欄まで選択する。
-d2		濁点、半濁点、制御文字を第 2 欄から削除する。
-e2		第 2 欄から浮動小数点数を抽出する。
-f		どの行も同じ欄数を有しているか点検する。
-i2-3	-b01110000000	第 2 欄から第 3 欄まで選択する。
-k2		第 2 欄では、半角カタカナをすべて ASCII の 1 バイト文字に置換して、制御文字を削除する。
-L	-LF	0Ah のみで改行する。
-n2		第 2 欄から整数を抽出する。
-o2		第 2 欄から制御文字を削除する。
-p		濁点の代用文字として、H が削除される。
-q		コンマが引用符どうしの間にある場合、コンマを削除する。
-r		各行で、欄を逆順に並べ替える。
-s,	.csv の拡張子	区切り記号としてコンマを使用する。
-s¥l	-s¥x7C	区切り記号として縦線を使用する。
-u		EUC-JP のテキストを読み込む。
-V		バージョン番号およびコマンドの例を表示する。
-VC		コマンドの例、短い説明を表示する。
-x-		.csv がファイル名に含まれるか点検してはならない。

¥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、narfi 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、-s¥v は、-s¥x3E と同じ結果を生じます。下表では、-s の例を示します。-a 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
-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

表計算ソフトウェア

テストファイルとして添付のField-21.csvは、第1行のみ第21欄が記入されています。

Field-21.csvをExcel 2007で読み込んだ例

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
2	85	52	86	48	68	22	59	44	63	61	92	37	9	83	55	58	51	31	60	2	
3	18	53	10	36	22	69	60	98	2	20	39	25	29	44	99	60	60	26	3	73	
4	93	15	38	47	92	70	49	2	38	69	29	9	21	27	86	89	73	25	31	47	
5	32	43	69	31	69	38	13	47	18	60	16	13	30	66	2	80	88	30	92	30	
6	99	29	69	42	17	87	81	79	28	21	72	26	60	97	49	60	99	14	52	66	
7	25	4	79	44	93	20	63	94	43	93	44	93	29	64	86	97	49	80	70	24	
8	13	1	76	59	17	21	45	22	96	88	68	46	88	92	25	94	8	50	64	40	
9	67	93	45	92	76	73	41	52	46	83	39	76	70	76	30	85	77	18	92	43	
10	33	21	28	62	4	51	53	24	39	44	59	2	72	7	14	47	27	30	74	0	
11	84	67	32	53	65	25	84	47	21	80	29	93	21	63	73	24	44	75	79	68	
12	9	38	66	3	14	28	54	67	74	37	54	34	27	82	35	70	7	43	2	43	
13	97	58	86	22	59	80	53	71	90	91	58	37	33	41	59	53	10	6	99	27	
14	24	23	24	46	34	8	74	81	77	51	24	92	89	29	17	30	63	29	3	21	
15	87	56	5	3	86	61	83	59	81	40	87	98	24	90	27	7	22	38	74	4	
16	41	81	72	43	93	53	45	54	95	9	40	70	88	39	28	99	69	85	80	66	

表の最初の部分

atnd コマンドで、表の最初の部分を表示してみました。

```
C:\¥Q>atnd -h16 Field-21.csv
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21
85,52,86,48,68,22,59,44,63,61,92,37,09,83,55,58,51,31,60,02,
18,53,10,36,22,69,60,98,02,20,39,25,29,44,99,60,60,26,03,73,
93,15,38,47,92,70,49,02,38,69,29,09,21,27,86,89,73,25,31,47,
32,43,69,31,69,38,13,47,18,60,16,13,30,66,02,80,88,30,92,30,
99,29,69,42,17,87,81,79,28,21,72,26,60,97,49,60,99,14,52,66,
25,04,79,44,93,20,63,94,43,93,44,93,29,64,86,97,49,80,70,24,
13,01,76,59,17,21,45,22,96,88,68,46,88,92,25,94,08,50,64,40,
67,93,45,92,76,73,41,52,46,83,39,76,70,76,30,85,77,18,92,43,
33,21,28,62,04,51,53,24,39,44,59,02,72,07,14,47,27,30,74,00,
84,67,32,53,65,25,84,47,21,80,29,93,21,63,73,24,44,75,79,68,
09,38,66,03,14,28,54,67,74,37,54,34,27,82,35,70,07,43,02,43,
97,58,86,22,59,80,53,71,90,91,58,37,33,41,59,53,10,06,99,27,
24,23,24,46,34,08,74,81,77,51,24,92,89,29,17,30,63,29,03,21,
87,56,05,03,86,61,83,59,81,40,87,98,24,90,27,07,22,38,74,04,
41,81,72,43,93,53,45,54,95,09,40,70,88,39,28,99,69,85,80,66,
C:\¥Q>
```

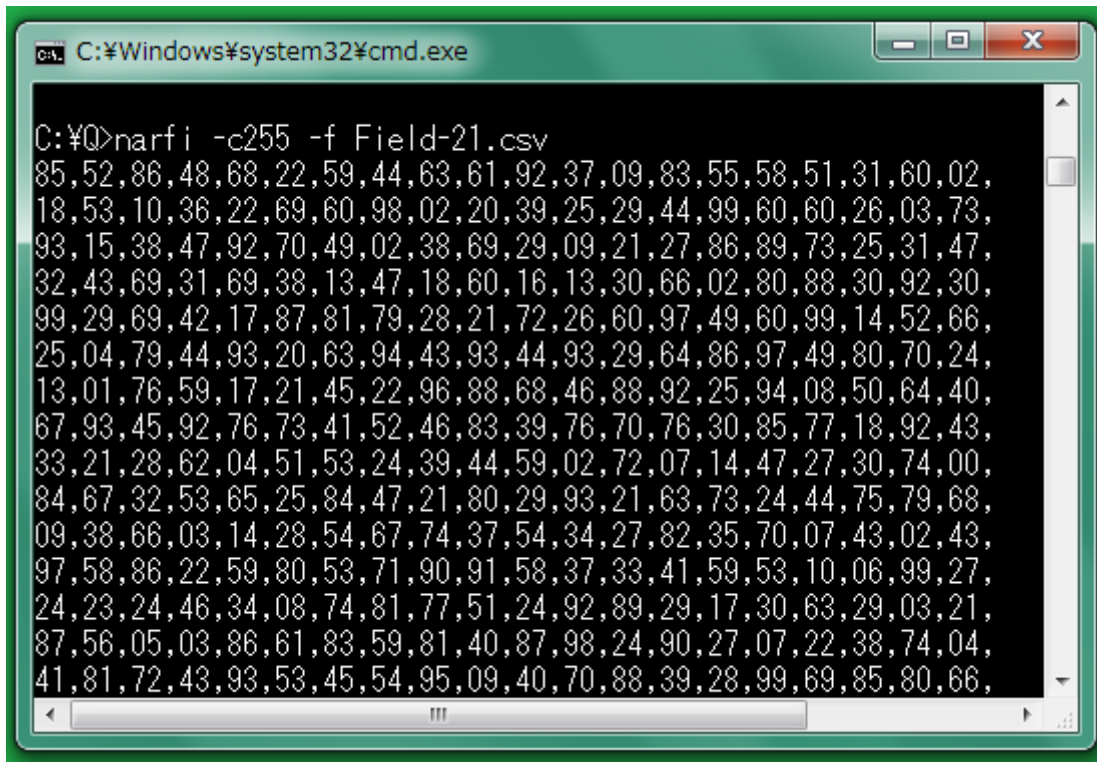
最終の欄が空である

上記のとおり、Excel 2007 で読み込んだ場合、最終の欄が空であるか判明します。最終の欄が空である場合、narfi コマンドは、最終の欄をカウントしません。

欄をカウント

利用者は、第 1 行から第 255 行まで選択できますが、narfi コマンドは、ファイルから行を読み込んで、各行で欄をカウントします。どの行も同じ欄数を有しているか点検するには、narfi コマンドのパラメータとして、-f を付加してください。まず、narfi コマンドは、第 1 行をカウントします。次に、各行が第 1 行と同じ欄数を有しているか点検します。さらに、第 1 行とは欄数が異なる場合、標準出力に行を書き出します。第 1 行は 21 欄あるにもかかわらず、他の行は、実質的に 20 欄しかない例

```
narfi -c255 -f Field-21.csv
```

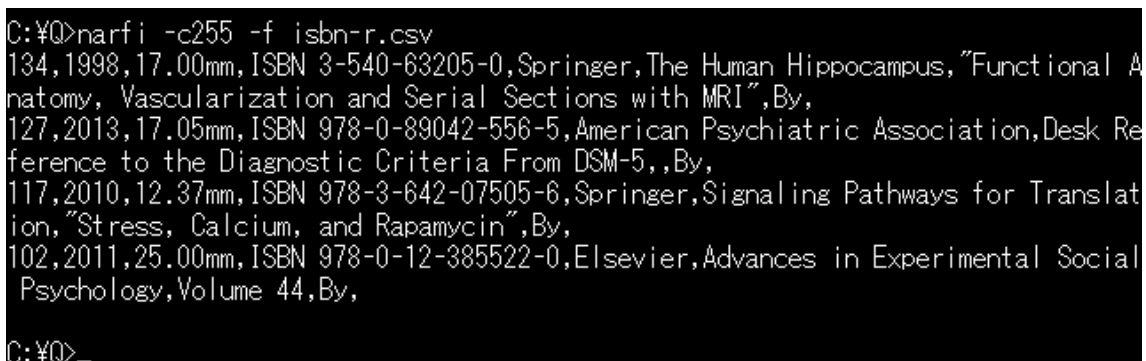


```
C:\Windows\system32\cmd.exe
C:\>narfi -c255 -f Field-21.csv
85,52,86,48,68,22,59,44,63,61,92,37,09,83,55,58,51,31,60,02,
18,53,10,36,22,69,60,98,02,20,39,25,29,44,99,60,60,26,03,73,
93,15,38,47,92,70,49,02,38,69,29,09,21,27,86,89,73,25,31,47,
32,43,69,31,69,38,13,47,18,60,16,13,30,66,02,80,88,30,92,30,
99,29,69,42,17,87,81,79,28,21,72,26,60,97,49,60,99,14,52,66,
25,04,79,44,93,20,63,94,43,93,44,93,29,64,86,97,49,80,70,24,
13,01,76,59,17,21,45,22,96,88,68,46,88,92,25,94,08,50,64,40,
67,93,45,92,76,73,41,52,46,83,39,76,70,76,30,85,77,18,92,43,
33,21,28,62,04,51,53,24,39,44,59,02,72,07,14,47,27,30,74,00,
84,67,32,53,65,25,84,47,21,80,29,93,21,63,73,24,44,75,79,68,
09,38,66,03,14,28,54,67,74,37,54,34,27,82,35,70,07,43,02,43,
97,58,86,22,59,80,53,71,90,91,58,37,33,41,59,53,10,06,99,27,
24,23,24,46,34,08,74,81,77,51,24,92,89,29,17,30,63,29,03,21,
87,56,05,03,86,61,83,59,81,40,87,98,24,90,27,07,22,38,74,04,
41,81,72,43,93,53,45,54,95,09,40,70,88,39,28,99,69,85,80,66,
```

不規則な表

テストファイルとして添付の isbn-r.csv は、9 欄ありますが、最終の欄は記入されていない行があります。

narfi -c255 -f isbn-r.csv



```
C:\>narfi -c255 -f isbn-r.csv
134,1998,17.00mm,ISBN 3-540-63205-0,Springer,The Human Hippocampus,"Functional A
natomy, Vascularization and Serial Sections with MRI",By,
127,2013,17.05mm,ISBN 978-0-89042-556-5,American Psychiatric Association,Desk Re
ference to the Diagnostic Criteria From DSM-5,,By,
117,2010,12.37mm,ISBN 978-3-642-07505-6,Springer,Signaling Pathways for Translat
ion,"Stress, Calcium, and Rapamycin",By,
102,2011,25.00mm,ISBN 978-0-12-385522-0,Elsevier,Advances in Experimental Social
Psychology,Volume 44,By,
C:\>
```

255 ではない選択

上記のとおり、-c255 -f が付加された場合、利用者が第 1 欄から第 255 欄まで選択したことになり、narfi コマンドは、欄をカウントします。9 欄であることがわかっていれば、9 より大きい数を指定してください。

CSV の入力ミスを発見

CSV エディタ、表計算ソフトウェアではなくテキストエディタで CSV を作成した場合、入力ミスで表がずれるおそれがあります。表の欄数(表の列数)が異なる部分を出力す

ることで、CSV の入力ミスを発見できる場合があります。

rufeful コマンド

添付の rufeful コマンドは、テキストファイルの各行の末尾から不要な文字を除去できます。

EUC-JP テキストの行末からコンマを削除

一般論として、UNIX のテキストファイルは、0Ah のみで改行します。rufeful コマンドのパラメータが-u1 であるならば、0Ah に先行する 1 バイトを無視します。EUC-JP のテキストである場合、しかも-u1 が指定された場合、rufeful コマンドは、0Ah の直前にあるコンマを削除できます。

0Dh も数えてください

Windows のテキストファイルは、0Dh および 0Ah で改行します。0Dh も数えてください。rufeful コマンドのパラメータが-u3 ならば、0Ah に先行する 3 バイトを無視します。-u1 が指定された場合、rufeful コマンドは、0Dh のみ削除します。

0Dh を挿入

-r が指定された場合、rufeful コマンドは、0Dh を挿入します。-u1 -r が指定された場合、rufeful コマンドは、0Dh を削除したにもかかわらず、0Dh を挿入するため、変化がありません。

行末からコンマを削除する

Windows のテキストファイルである場合、行末からコンマを削除するには、rufeful コマンドのパラメータとして、-u2 -r を付加してください。-m は、-u2 -r を意味します。rufeful コマンドは、0Ah が出現したとき、2 バイト(コンマおよび 0Dh)を無視して、改行します。-r が指定されたため、rufeful コマンドは、0Dh および 0Ah で改行します。

パラメータ	ほぼ同等な入力	説明
-m	-u2 -r	先行する 2 バイトを無視する。1 個目の 0Ah が出現したとき、0Dh および 0Ah で改行する。
-x2	-u3 -w2 -r	先行する 3 バイトを無視する。2 個目の 0Ah が出現したとき、0Dh および 0Ah で改行する。
なし	-w1	1 個目の 0Ah が出現したとき、0Ah のみで改行する。

Windows における EUC-JP

Windows のテキストファイルは、0Dh および 0Ah で改行します。-u2 が指定された場合、rufeful コマンドは、0Ah の直前にある 0Dh およびコンマを削除できます。下記の例では、EUC-JP テキストが 0Ah のみで改行されたことを、MIFES は、LF および下向き矢印で表現しています。

```
C:¥Q>TYPE setting.txt
C:¥s jisEUCjpCS¥MS932~EUC-JP.bin table
@ MS932
廿 substitute
sortk.txt bars.txt
0 -d
-w -w

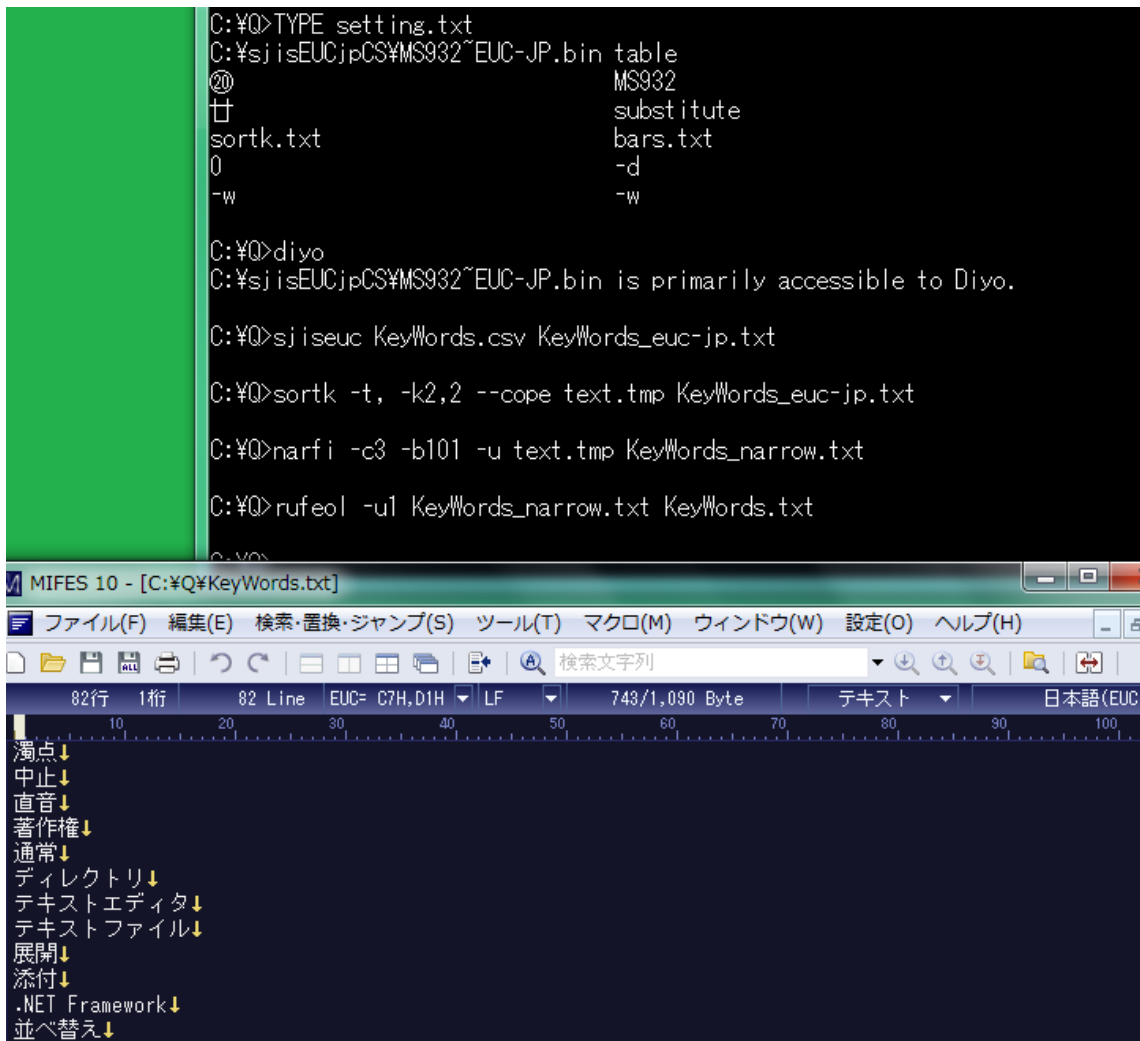
C:¥Q>diyo
C:¥s jisEUCjpCS¥MS932~EUC-JP.bin is primarily accessible to Diyo.

C:¥Q>sjiseuc KeyWords.csv KeyWords_euc-jp.txt

C:¥Q>sortk -t, -k2,2 --cope text.tmp KeyWords_euc-jp.txt

C:¥Q>narfi -c3 -b101 -u text.tmp KeyWords_narrow.txt

C:¥Q>rufcol -u1 KeyWords_narrow.txt KeyWords.txt
```



MIFES 10 - [C:¥Q¥KeyWords.txt]

ファイル(F) 編集(E) 検索・置換・ジャンプ(S) ツール(T) マクロ(M) ウィンドウ(W) 設定(O) ヘルプ(H)

検索文字列

82行 1桁 82 Line EUC= C7H,D1H LF 743/1,090 Byte テキスト 日本語(EUC)

濁点↓
中止↓
直音↓
著作権↓
通常↓
ディレクトリ↓
テキストエディタ↓
テキストファイル↓
展開↓
添付↓
.NET Framework↓
並べ替え↓

hanzen コマンド

添付の hanzen コマンドは、シフト JIS のテキストファイルを読み込んで、半角カタカナを全角カタカナに置換して、シフト JIS のテキストファイルに書き出します。

清音および濁点から濁音に置換

半角カタカナの濁点があれば、hanzen コマンドは、直前にあるカナと濁点を結合して、濁音のカタカナに置換します。たとえば、hanzen コマンドは、がを、カ`ではなくガに置換します。濁点を無視するには、hanzen コマンドに-c-を付加してください。

```
C:¥Q>TYPE gapa.csv
が使用できます、
パを使用できます、
ガを使用します、
バを使用します、

C:¥Q>hanzen -f2 -e1 gapa.csv gapa_zen.csv

C:¥Q>TYPE gapa_zen.csv
ガを使用できます、
バを使用できます、
ガを使用します、
バを使用します、

C:¥Q>
```

ファイルのサイズ

sortk.txt にしたがって、diyo コマンドは、たとえば、ヤを G に置換します。フを 7 に置換します。クを?に置換します。すなわち、1 バイトのカタカナを1バイトの ASCII(英字、数字、記号)に置換します。ファイルのサイズは、変化しません。diyo コマンドと異なり、hanzen コマンドは、1 バイトのカタカナを 2 バイトのカタカナに置換します。hanzen コマンドの結果として、ファイルのサイズが増大します。

行末のコンマを削除したい

narfi コマンドと同様に、hanzen コマンドは、CSV を読み込んで、ある欄を編集できます。CSV ではないテキストファイルを読み込んで、行末にコンマが書き出される場合があります。-m が付加された rufeol コマンドは、[行末のコンマ](#)を削除します。

rufeol -m marunana_zen.txt


```
C:¥Q>TYPE marunana.txt
```

```
ナ
```

```
ナ
```

```
菜
```

```
⑦
```

```
VI
```

```
C:¥Q>hanzen marunana.txt marunana_zen.txt
```

```
C:¥Q>rufeorl -m marunana_zen.txt marunana_zen_eol.txt
```

```
C:¥Q>TYPE marunana_zen_eol.txt
```

```
ナ
```

```
ナ
```

```
菜
```

```
⑦
```

```
VI
```

```
C:¥Q>
```

パラメータ	ほぼ同等な入力	説明
-8		UTF-8 のテキストを読み込む。
-a_	-a¥x5F	半角カタカナを廃止して、下線で代用する。
-b,	-b¥x2C	連結文字としてコンマを使用する。バイナリファイルとして binary.tmp を保存する。
-c-		濁点、半濁点を削除する。
-d		変換表を探索してはならない。
-e2		第 2 欄のみ半角カタカナを置換する。
-f3 -e2		各行にある 3 欄のうち、第 2 欄のみ半角カタカナを置換する。
--help	-VC	バージョン番号、用法、短い説明を表示する。
-i		0Ah 以外の制御文字を削除する。0Dh および 0Ah で改行する。
-L	-LF	0A のみで改行する。
-L-	-CRLF	0D および 0A で改行する。
-m	-s¥x0D -b¥x0A	区切り文字として 0Dh を使用する。連結文字として 0Ah を使用する。バイナリファイルとして binary.tmp を保存する。
-s,	-s¥x2C	区切り文字としてコンマを使用する。
-U		標準エラー出力ではなく標準出力に書き出す。
-u		EUC-JP のテキストを読み込む。
-w		どの変換表を使用するか表示する。
-x-		.csv がファイル名に含まれるか点検してはならない。
-V		バージョン番号を表示する。

バイナリファイルを作成

パラメータとして -b¥xHH が指定された場合 (H は、十六進数 [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F])、hanzen コマンドは、改行しないで、1 バイトの文字を挿入します。十六進数で文字コードを指定してください。たとえば、-b¥x00 が指定された場合、hanzen コマンドは、0 を挿入して連結して、バイナリファイルとして binary.tmp を保存します。

改行文字を挿入する

-b¥x0A が指定された場合、hanzen は、改行文字 (0Ah) を挿入します。すなわち、改行が可能です。

1 欄しかない表

CSV ではないテキストファイルは、1 欄しかない表であるとみなすことができます。利用者が、区切り文字として 0Dh を指定して、挿入文字として 0Ah を指定した場合、hanzen は、バイナリファイルとして binary.tmp を保存します。Windows では、0Dh および 0Ah で改行するため、バイナリファイルではなくテキストファイルとして、binary.tmp を利用できます。

2 個のファイル

利用者が 2 個のファイルを指定した場合、hanzen は、2 個目のファイル名で binary.tmp を改名します。-m は、-s¥x0D -b¥x0A を意味します。

```
C:¥Q>TYPE marunana.txt
ナ
ナ
菜
⑦
VI

C:¥Q>hanzen -m marunana.txt marunana_zen.txt

C:¥Q>TYPE marunana_zen.txt
ナ
ナ
菜
⑦
VI

C:¥Q>
```

キーワードの読み方の欄

キーワードの読み方を半角カタカナで入力した欄は、hanzen コマンドで、全角カタカナに置換できます。第 1 欄、第 2 欄、第 3 欄のうち、第 2 欄のみで半角カタカナを全角カタカナに置換するには、コマンドプロンプトに、下記のコマンドを入力して、Enter を押してください。

hanzen -f3 -e2 KeyWords.csv

全角カタカナで並べ替える

濁点を削除するために-c-を付加した hanzen コマンドで、まず、KeyWords.csv の第 2 欄にある半角カタカナを全角カタカナに置換しました。次に、sjiseuc コマンドで、全角カタカナのファイルを読み込んで、EUC-JP のテキストファイルを作成しました。さらに、キーワードの読み方の欄(第 2 欄)に関して行を並べ替えました。

```
C:¥Q>hanzen -f3 -e2 -c- Keywords.csv Keywords_zen.txt
C:¥Q>sjiseuc Keywords_zen.txt Keywords_zen_euc-jp.txt
C:¥Q>sortk -t, -k2,2 -e -o text.tmp Keywords_zen_euc-jp.txt
C:¥Q>
```

```
84行 1桁 84 Line EUC= 50H LF 1,877/2,734 Byte テキスト 日本語(EUC)
濁音, タクオン, ↓
濁点, タクテン, ↓
中止, チュウシ, ↓
直音, チョクオン, ↓
著作権, チョウサクケン, ↓
通常, ツウショウ, ↓
ディレクトリ, テイレクトリ, ↓
テキストエディタ, テキストエディタ, ↓
テキストファイル, テキストファイル, ↓
展開, テンカイ, ↓
添付, テンフ, ↓
.NET Framework, トットネットフレエムワアク, ↓
並べ替え, ナラヘカエ, ↓
日本語, ニホンゴ, ↓
バージョン, ハアション, ↓
廃止, ハイシ, ↓
バイナリファイル, ハイナリファイル, ↓
```

画像をクリップボードにコピー

Windows の画面をクリップボードにコピーするには、[Print/Screen]キーを押してください。Windows アクセサリのペイントを起動して、貼り付けることができます。Paint Shop Pro など、他のソフトウェアも新しいファイルに貼り付けることができます。

文字をクリップボードにコピー

コマンドプロンプトに表示された文字をテキストファイルにコピーするには、まず、コマンドプロンプトでドラッグして、反転表示を行ってください(ドラッグしても反転表示ができない場合、コマンドプロンプトのタイトルを右クリックして、メニューから編集を選択して、範囲指定を選択してください)。反転表示された部分が選択されています。次に、Enter キーを押してください。さらに、テキストエディタを起動して、新しいファイルを開いて、Ctrl を押しながら V を押してください。

obot コマンド

添付の obot コマンドは、ファイルの開始部分を 2 桁の十六進数で表現します。たとえば、テキストファイルの 5 行ぶんを表示するには、obot コマンドのパラメータとして -m5 を付加してください。

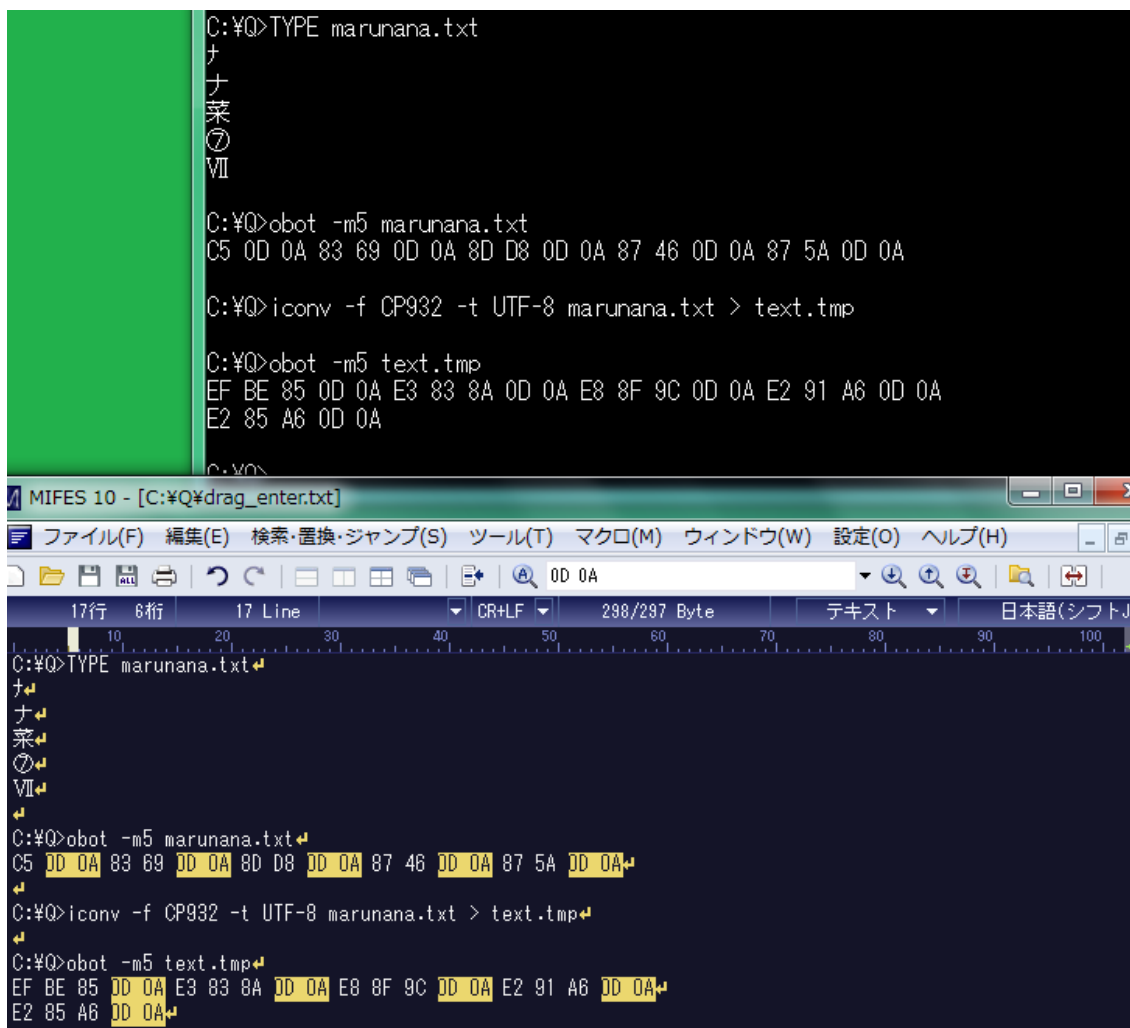
3 バイト文字

まず、GNU Win32 の iconv コマンドで、MS932 のテキストファイルを UTF-8 の一時ファ

イルに変換しました。次に、obot コマンドで、一時ファイルを読み込んで、十六進数で表現しました。半角カタカナが 1 バイトではなく 3 バイトになったことがわかります。

丸付きの数字を変換

リダイレクトで一時ファイルを作成したため、0Dh および 0Ah で改行されています。丸付きの数字を変換するには、iconv コマンドのパラメータとして、Shift_JIS ではなく CP932 を iconv に入力します。



```
C:\%Q>TYPE marunana.txt
ナ
ナ
菜
㊦
VII


C:\%Q>obot -m5 marunana.txt
05 0D 0A 83 69 0D 0A 8D D8 0D 0A 87 46 0D 0A 87 5A 0D 0A

C:\%Q>iconv -f CP932 -t UTF-8 marunana.txt > text.tmp

C:\%Q>obot -m5 text.tmp
EF BE 85 0D 0A E3 83 8A 0D 0A E8 8F 9C 0D 0A E2 91 A6 0D 0A
E2 85 A6 0D 0A
```

The screenshot shows a Windows command prompt window with a green background. The user enters several commands: 'TYPE marunana.txt' (displaying Japanese text), 'obot -m5 marunana.txt' (displaying hex dump), 'iconv -f CP932 -t UTF-8 marunana.txt > text.tmp', and 'obot -m5 text.tmp' (displaying hex dump). Below the command prompt is a Notepad window titled 'MIFES 10 - [C:\%Q\drag_enter.txt]'. The search bar contains '0D 0A'. The search results show the same commands and hex dumps as the command prompt, with the hex values '0D 0A' highlighted in yellow. The Notepad window also shows a search bar with '0D 0A' and a search button.

すべて強調表示

MIFES の検索ウィンドウは、右端にある  をクリックすれば、ウィンドウの幅が拡大して、[すべて強調表示]のチェックボックスが表示されます。一斉表示とも言います。この設定が有効であれば、検索に成功した文字列は、すべて強調表示が行われます。上記の例では、コマンドプロンプトをドラッグして、Enter キーを押してクリップボードにコピーして、drag_enter.txt に貼り付けて、0D 0A を検索しました。半角カタカナ

が 1 バイトではなく 3 バイトになったことがわかりやすい。

読み込む量を制限するパラメータ

-b 数でバッファのサイズを指定できます。テキストファイルもバイナリファイルとして処理されます。-m 数でテキストファイルの行数を制限できます。行が指定されていない場合、obot は、十六進数の記法で 3 行を表現します。

書き出す量を制限するパラメータ

-c 数で表の欄数(表の列数)を指定できます。-r 数で表の行数を制限できます。

パラメータ	ほぼ同等な入力	説明
-b2048		2048 バイトを読み込む。
-c20		出力される表の各行は、20 欄ある。
-m3		テキストファイルの開始部分にある 3 行を十六進数で表現する。
-s,	-s¥x2C	区切り記号として、空白ではなくコンマを使用する。
-r20		出力される表は、高々 20 行である。
-V		バージョン番号を表示する。
-VC		用法、短い説明を表示する。

タブで区切ることができる

obot の設定ファイルは、set_obot.txt です。設定ファイルは、いわゆるシフト JIS (MS932) のファイルです。obot の設定ファイルは、6 行からなります。設定内容、タブ、コメントの順に入力してください。

```
C:¥$jisEUCjpCS¥test_files>TYPE set_obot.txt
obot.exe      file
640           buffer
3             lines
20h          separator
20            fields
2147483647    rows

C:¥$jisEUCjpCS¥test_files>cd ¥Q
C:¥Q>
```

行	設定内容	コメント	説明
1	obot.exe	file	バイナリファイルとして読み込む。
2	640	buffer	640 バイトのバッファーに読み込む。
3	3	lines	テキストの開始部分にある 3 行を読み込む。
4	20h	separator	区切り記号として、空白(20h)を使用する。
5	20	fields	出力される表は、20 欄である。
6	2147483647	rows	出力される表は、2147483647 行を超えない

設定ファイルは必須ではない

obot の設定ファイルは、set_obot.txt です。設定ファイルは、いわゆるシフト JIS (MS932) のファイルです。set_obot.txt は、必須ではありません。obot が set_obot.txt を開くことができる場合も、利用者は、obot コマンドのパラメータで設定を変更できます。set_obot.txt による設定を、obot コマンドのパラメータで変更できないと思われる場合、カレントディレクトリから set_obot.txt を削除してください。