

F*TRAN + Ver. 8.0

操作説明書／解説編

第1版

株式会社富士通ビー・エス・シー

はじめに

F*TRAN+ V8.0をお買い上げいただき、ありがとうございます。

F*TRAN+ (エフトラン・プラス) は、汎用機／オフコン／UNIXなどのホストのファイル転送データと、パソコンの標準であるWindowsファイルとのデータ交換をする汎用性の高いファイル変換ユーティリティです。Windowsファイル間のデータ変換もできます。発売以来、多くのユーザにささえられている、実績あるソフトウェアです。

F*TRAN+ V8.0には、動作するOS別に2種類の製品があります。

- F*TRAN+ V8.0 Server
Server系、およびDesktop系OSで動作
- F*TRAN+ V8.0 Desktop
Desktop系OSで動作

F*TRAN+ V8.0では、Windows10上での動作をサポートしています。

F*TRAN+のマニュアルには、導入編、解説編(本書)、コマンド編、マルチレコード編、プログラム応用編があります。

2017年1月
株式会社 富士通ビー・エス・シー

目 次

第1章 概 要

1.1	F*TRAN+とは	2
1.2	ホストファイルとは	4
1.3	Windowsのファイル(DOS形式)とは	5
1.4	3つのファイル変換機能(汎用機・オフコン系)	6
1.5	3つのファイル変換機能(Unix系)	8
1.6	3つのファイル変換機能(Windows系)	10
1.7	補助機能	12
1.8	セットアップ機能	13
1.9	特長のまとめ	15

第2章 変換の仕様

2.1	Unicodeについて	20
2.2	漢字シフトコードKI/KOについて	23
2.3	ファイル変換機能の使い分け	25
2.4	ファイル変換機能の使い方のポイント	27
2.5	ホスト側の扱えるデータ形式	29
2.6	Windows側の扱えるデータ形式	30
2.7	テキストファイル変換の仕様	32
2.8	データファイル変換1/プリント形式の仕様	39
2.9	データファイル変換2/デリミタ形式の仕様	49
2.10	ランダムファイル変換の仕様	61
2.11	ANK変換	69
2.12	漢字変換	72
2.13	コード変換方式のサポートパターン	74
2.14	ホスト漢字のレパートリー	78
2.15	漢字コード変換の仕組み	79
2.16	漢字対応表について	80
2.17	変換できないホストの拡張漢字の扱い	81
2.18	ホストの一般漢字と拡張漢字の区別	83
2.19	ホストの拡張漢字の領域	85
2.20	変換できないWindowsの拡張漢字の扱い	86
2.21	数値変換	87
2.22	汎用機・オフコン系のゾーン/パック形式	89

2.23	オープン系 (Windows・Unix) のゾーン／パック形式	9 3
2.24	BCD形式	9 5
2.25	2進形式	9 6
2.26	パック／BCD／2進形式の桁数とバイト数の換算	9 7
2.27	定数挿入	9 9
2.28	ホスト→Winテキストファイル変換のしくみ	1 0 2
2.29	ホスト→Winデータファイル変換のしくみ	1 0 4
2.30	ホスト→Winランダムファイル変換のしくみ	1 0 6
2.31	Win→ホストテキストファイル変換のしくみ	1 0 8
2.32	Win→ホストデータファイル変換のしくみ	1 1 0
2.33	Win→ホストランダムファイル変換のしくみ	1 1 2

第3章 操作の基礎

3.1	ファイルの指定	1 1 5
3.2	式の指定	1 1 7
3.3	ピクチャの指定	1 1 9
3.4	2進ピクチャの指定	1 2 1
3.5	日付データの指定	1 2 3

第4章 操作

4.1	F*TRAN+ 起動と終了	1 2 7
4.2	ホスト→Winファイル変換のファイル指定と共通オプション	1 3 0
4.3	ホスト→Winテキストファイル変換	1 3 9
4.4	ホスト→Winデータファイル変換	1 4 7
4.5	ホスト→Winランダムファイル変換	1 9 0
4.6	Win→ホストファイル変換のファイル指定と共通オプション	2 3 5
4.7	Win→ホストテキストファイル変換	2 4 4
4.8	Win→ホストデータファイル変換	2 5 3
4.9	Win→ホストランダムファイル変換	2 9 9
4.10	スクリプトファイル	3 4 2
4.11	Winファイラ	3 5 0
4.12	環境設定	3 5 7
4.13	変換設定	3 6 2
4.14	漢字対応表エディタ	3 8 1
4.15	ログの設定と表示	4 0 3
4.16	ヘルプ	4 1 0

第5章 ログ

5.1	ログについて	4 1 1
5.2	標準インストールモードのログ	4 1 3

付録A

エラーメッセージ (共通)

付録B

ログメッセージ一覧

本書で用いる表記法

●本文と画面のパラメータ類の表記法

{ A B C }	A、B、またはCのうち、どれか1つを選択します。省略はできません。
$\left \begin{array}{c} A \\ B \\ C \end{array} \right $	同上。
(A / B / C)	同上。
[A]	Aは省略できます。
[A / <u>B</u> / C]	A、B、またはCのうち、どれか1つを選択します。省略が可能で、その場合、下線を引いたBを選択したものとみなします。
$\left[\begin{array}{c} A \\ \underline{B} \\ C \end{array} \right]$	同上。
(A / [B] / C)	同上。ただし、[] でくくったBを選択したものとみなします。
X . . .	X類を A B C のように列挙します。
n、n n、< n >	1 0進数を指定します。 (< > は表記上の記号で、入力はしません)
x x H	1 6進で x x です。Hを省くこともあります。
↓	改行を意味します。リターンキーのシンボルです。
<u>a</u>	下線部を入力します。
<u>a b c</u> ↓	下線部を入力し、リターンキーを押します。
CTRL + A	コントロール (CTRL) キーを押しながら、Aキーを押します。コントロールAと読みます。
^ A	同上。
d :	ドライブ A : や C : など、任意のドライブ指定を表します。

◆注意 ----- 実画面と少し差異がある

本書に示す画面と実際の画面には、若干の差異がある場合があります。あらかじめ、ご了承ください。

第1章



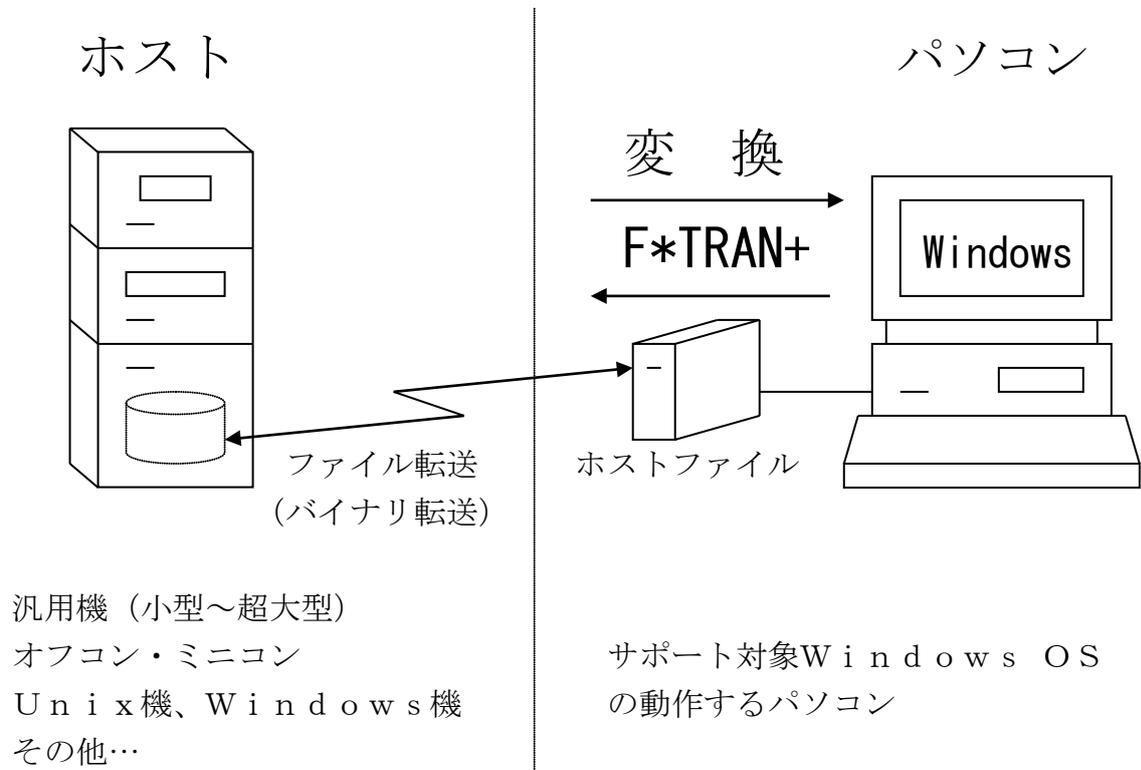
概 要

1. 1 F*TRAN+とは

F*TRAN+は、汎用機やオフコン、Unix機などのホストのファイル転送データとパソコンのデータ交換を実現するファイルトランスレータです。Windows OS上で動作し、オンライン・データとWindowsファイルの相互変換ができます。Windowsファイル間のデータ変換もできます。

●さまざまなホストに対応

F*TRAN+の対象とするデータは、基本的にホストのコード体系のままでファイル転送（バイナリ転送）したデータになります。F*TRAN+を使えば、**さまざまなホストとのデータ交換**がパソコン上で簡単にできるようになります。



●オンラインファイル転送の欠点

ホストとのオンライン接続によるファイル転送は、非常に便利なものです。しかし欠点もあり、現状では

データ加工能力が不十分

だといえます。テキスト転送とバイナリ転送ぐらいしかできない場合が多く、まだ汎用性が低いのです。

● **F * TRAN+**によるファイル変換の利点

そこで、オンラインファイルを扱う**F * TRAN+**はつぎのような利点を持ちます。まず、最大の特長は、

メーカーの壁を越え、さまざまなホストに対応でき、
データ加工能力が高く、Windowsのさまざまなソフトに対応でき、
さらに、巨大ファイルにもにも対応できる

ことです。汎用性が高いのです。また、

変換速度が速く、自動化も容易

です。このため、運用コストが安くすむのも大きな利点です。このように、オンラインファイル転送の欠点を補う特長を持っています。そして、**F * TRAN+**は、

ウインドウ上の項目を設定するだけで操作することも
コマンド行方式でバッチ処理として実行することも
OLEオートメーションを利用してプログラムから操作することも *1

できます。言い替えば、

ウインドウ操作による使いやすさ・覚えやすさ
コマンド行方式やOLEオートメーションによる定形業務の自動化のしやすさ *1

を両立させているのです。また、**F * TRAN+**は

マルチファイル対応

です。

パソコンを含め複数メーカーの機種が混在している“マルチベンダ環境”では、**F * TRAN+**は上手に使いこなしたいソフトウェアです。

*1) OLEオートメーションとは、外部から**F * TRAN+**を操作するインターフェースのことです。

1. 2 ホストファイルとは

● F * T R A N + の「ホストファイル」とは、つぎのファイルを指します。

F * T R A N + では、ホスト端末エミュレータや F T P などによってファイル転送（バイナリ転送）された、または、するファイルを「ホストファイル」として定義しています。その他、W i n d o w s 間のデータ変換の際には、W i n d o w s ファイルをホストファイルとして扱うこともできます。

ホストは、つぎの3つのタイプに分けることができます。

汎用機・オフコン系（汎用機またはオフコン）：

ホストのコード体系により構成された、
固定長レコードの W i n d o w s ファイル（D O S 形式）
（レコード長情報を持たないため、ユーザーがレコード長を指定する必要あり）

U n i x 系（U n i x 機）：

U n i x のコード体系により構成された、
可変長／固定長の W i n d o w s ファイル（D O S 形式）

W i n d o w s 系（W i n d o w s の動作するパソコン）：

W i n d o w s のコード体系により構成された、
可変長／固定長の W i n d o w s ファイル（D O S 形式）

1. 3 Windows のファイル (DOS 形式) とは

●パソコンの標準 ---- DOS形式

インテル80x86系のCPUを搭載したパソコンの標準OSとなった、マイクロソフト社のDOS。そのDOSのディスク・ファイル管理形式をDOS形式と呼びます。もはや、DOS形式はパソコンの標準形式といえるでしょう。

●DOS形式の特長

DOS形式には、今述べた互換性の高さ以外に、

バイトストリーム指向で、スペース効率がよい
階層的ディレクトリ管理ができる
ファイル名に拡張子をつけることができ、整理が容易である
動的な空き領域管理ができる

などの特長があります。

●ロングファイル名

DOS形式では、ファイル名(8文字以内)、ファイル拡張子(3文字以内)という制約がありました。Windowsでは、最高255文字までのファイル名(複数のピリオド[.]の使用も可)を使うことができるようになりました。

F*TRAN+では、DOS形式をベースにロングファイル名をサポートしたファイルをWindowsファイルと表現しています。

1. 4 3つのファイル変換機能（汎用機・パソコン系）

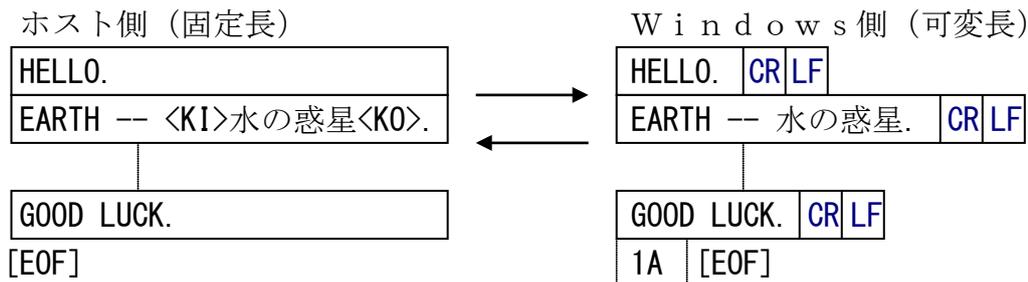
F*TRAN+には、3つの（実質的には4つの）ファイル変換機能があり、

- テキストファイル変換 ----- おもにソースプログラムの変換に
- データファイル変換1／プリント形式 ---- // 市販ソフトとのデータ交換に
- データファイル変換2／デリミタ形式 ---- // 市販ソフトとのデータ交換に
- ランダムファイル変換 ---- Windows COBOL の順ファイルや BASIC のランダムファイルなどの交換に

と呼んでいます。これらを目的に応じて使い分けます。どれもホスト↔Windows 双向サポートです。

●テキストファイル変換

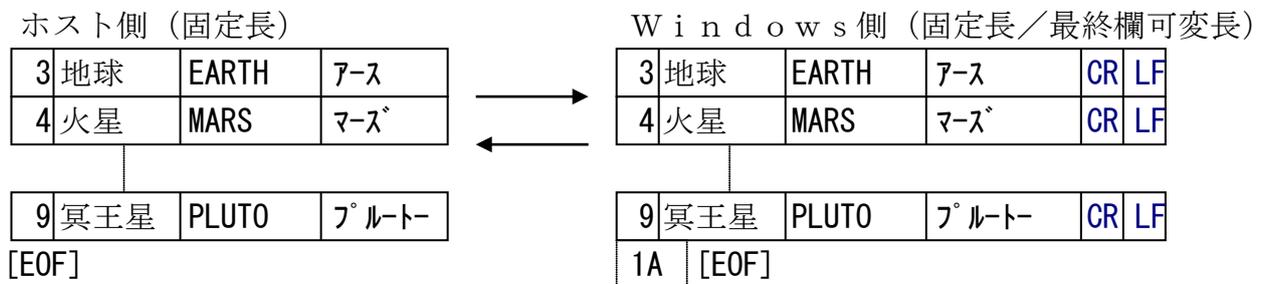
おもに、ソースプログラムの変換に使います。生データの変換やリストファイルの変換にも使えます。



●データファイル変換1／プリント形式

おもに、ホストのCOBOLデータと、市販ソフトの入出力用テキストファイルの変換に使います。

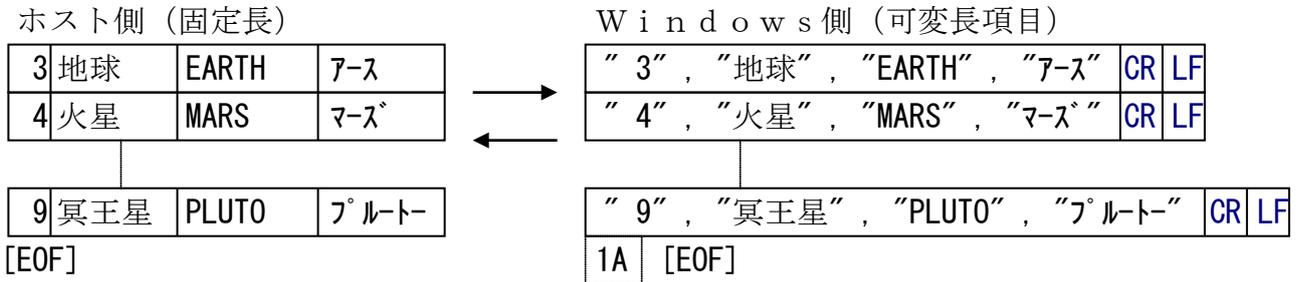
固定長項目、区切り文字なしのテキストファイルを、項目別に分けて変換できます。もちろんレコード全体で1項目とみなす、単純な変換もできます。



●データファイル変換2／デリミタ形式

おもに、ホストのCOBOLデータと、市販ソフトの入出力用テキストファイルの変換に使います。

可変長項目、区切り文字（デリミタ）付きのテキストファイルを項目別に分けて変換でき、コンマ区切り形式、タブ区切り形式、スペース区切り形式の3タイプが扱えます。



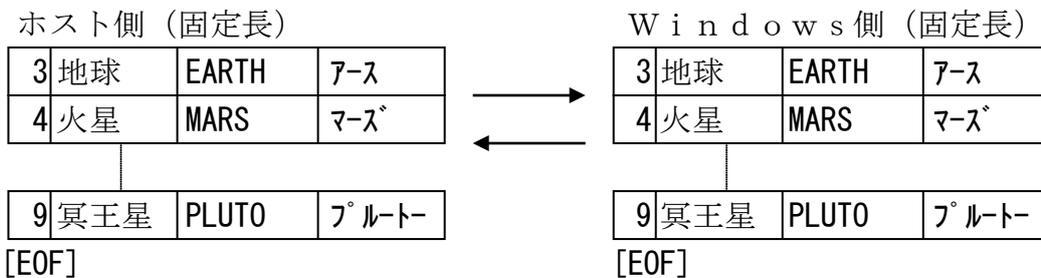
一般に、市販ソフトで「テキストファイルの入出力ができます」といえば、以上のうちコンマ区切り形式か、プリント形式ファイルを指すのがふつうです。

とくに、コンマ区切り形式は、パソコンのデータベースや表計算ソフトの最も標準的な入出力ファイル形式で、CSV (Comma Separated Value) 形式という特別な名前がついています。この形式は、市販ソフトへの入力操作がとても簡単なのが特長です。

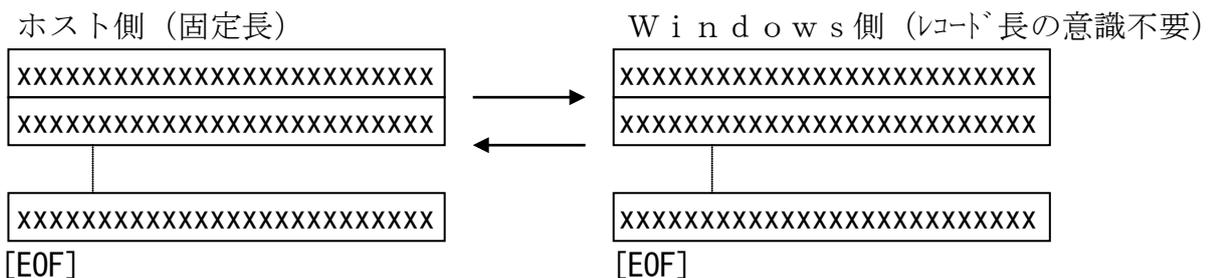
●ランダムファイル変換

おもに、Windows COBOLの順ファイルやBASICのランダムファイルの変換に使います。

項目別の変換ができる点では、データファイル変換1 / プリント形式とよく似ています。しかし、改行コードも区切り文字もなく、固定長で、レコード長を意識しながら扱う必要があります。



バイナリ変換（レコード全体 / 一部を無変換で通すこと）もできます。機械語ファイル等を扱うときに便利です。



1. 5 3つのファイル変換機能 (Unix系)

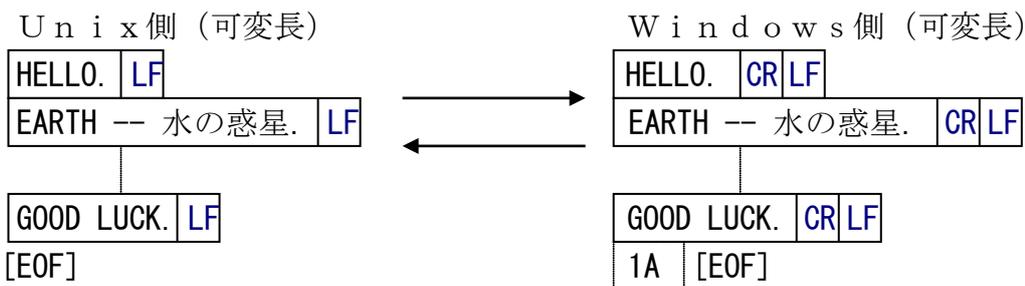
F*TRAN+には、3つの（実質的には4つの）ファイル変換機能があり、

- テキストファイル変換 ----- テキストファイルの変換に
- データファイル変換1 / プリント形式 ---- おもに市販ソフトとのデータ交換に
- データファイル変換2 / デリミタ形式 ---- " 市販ソフトとのデータ交換に
- ランダムファイル変換 ---- Windows COBOL の順ファイルや BASIC のランダムファイルなどの交換に

と呼んでいます。これらを目的に応じて使い分けます。どれもUnix ↔ Windows 双方向サポートです。

● テキストファイル変換

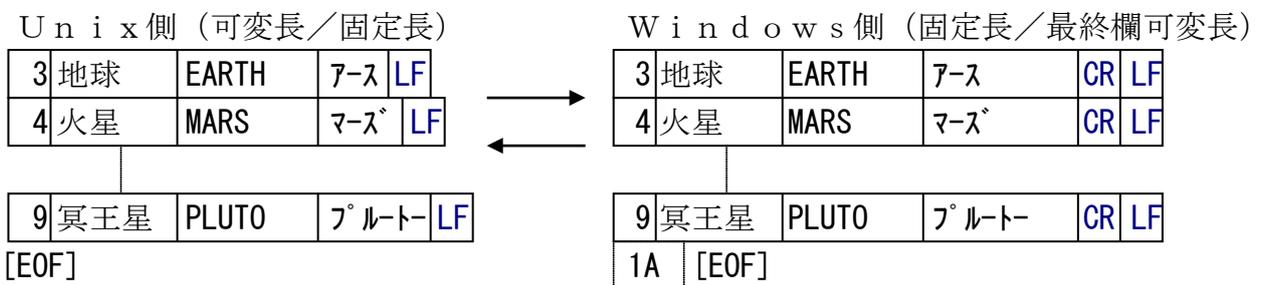
テキストファイル（ソースプログラム等）の変換に使います。生データの変換にも使えます。



● データファイル変換1 / プリント形式

おもに、Unixのデータと、市販ソフトの入出力用テキストファイルの変換に使います。

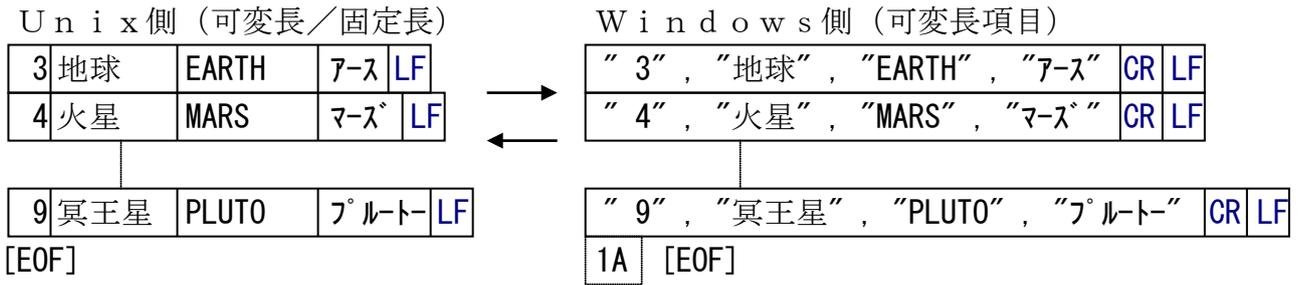
Unixの可変長（プリント / デリミタ形式）または固定長のテキストファイルを、項目別に分けて変換できます。



● データファイル変換2 / デリミタ形式

おもに、Unixのデータと、市販ソフトの入出力用テキストファイルの変換に使います。

Unixの可変長（プリント / デリミタ形式）または固定長のテキストファイルを項目別に分けて変換でき、コンマ区切り形式、タブ区切り形式、スペース区切り形式の3タイプが扱えます。



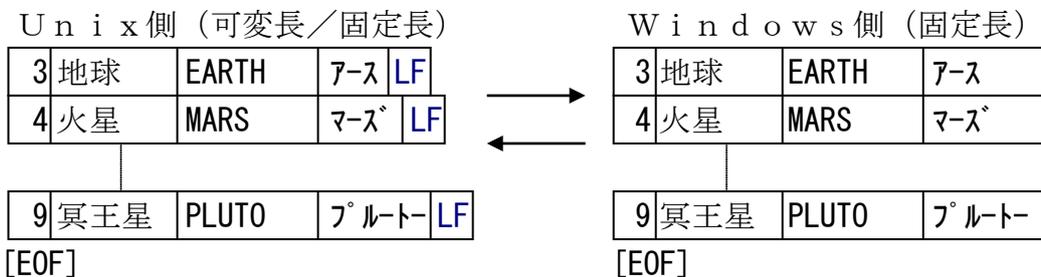
一般に、市販ソフトで「テキストファイルの入出力ができます」といえば、以上のうちコンマ区切り形式か、プリント形式ファイルを指すのがふつうです。

とくに、コンマ区切り形式は、パソコンのデータベースや表計算ソフトの最も標準的な入出力ファイル形式で、CSV (Comma Separated Value) 形式という特別な名前がついています。この形式は、市販ソフトへの入力操作がとても簡単なのが特長です。

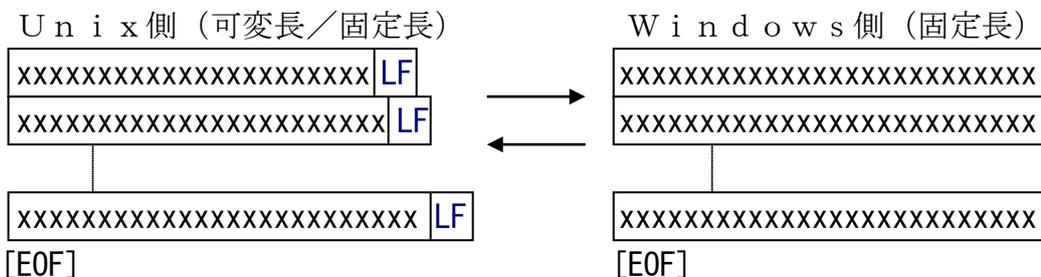
●ランダムファイル変換

おもに、Windows COBOLの順ファイルやBASICのランダムファイルの変換に使います。

Unixの可変長 (プリント/デリミタ形式) または固定長のテキストファイルを項目別に分けて変換できる点では、データファイル変換1/プリント形式とよく似ています。しかし、改行コードも区切り文字もなく、固定長で、レコード長を意識しながら扱う必要があります。



バイナリ変換 (レコード全体/一部を無変換で通すこと) もできます。



1. 6 3つのファイル変換機能 (Windows系)

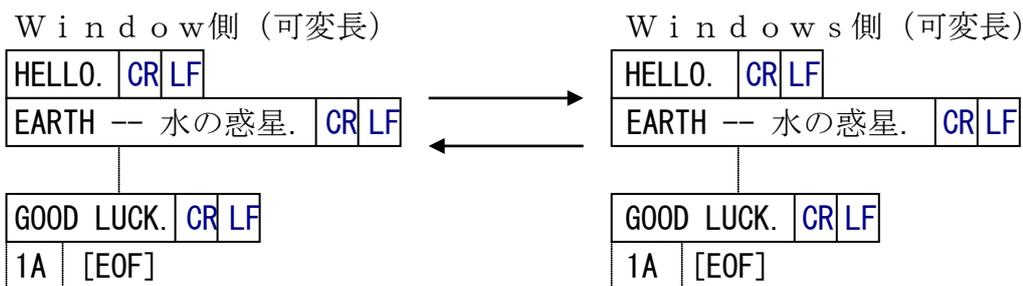
F * TRAN+には、3つの（実質的には4つの）ファイル変換機能があり、

- テキストファイル変換 ----- テキストファイルの変換に
- データファイル変換1 / プリント形式 ---- おもに市販ソフトとのデータ交換に
- データファイル変換2 / デリミタ形式 ---- " 市販ソフトとのデータ交換に
- ランダムファイル変換 ---- Windows COBOL の順ファイルや BASIC のランダムファイルなどの交換に

と呼んでいます。これらを目的に応じて使い分けます。どれも、Windowsファイル間の変換です。SJIS ⇄ UCS-2 / UTF-8 や UCS-2 ⇄ UTF-8 の変換も行えます。

● テキストファイル変換

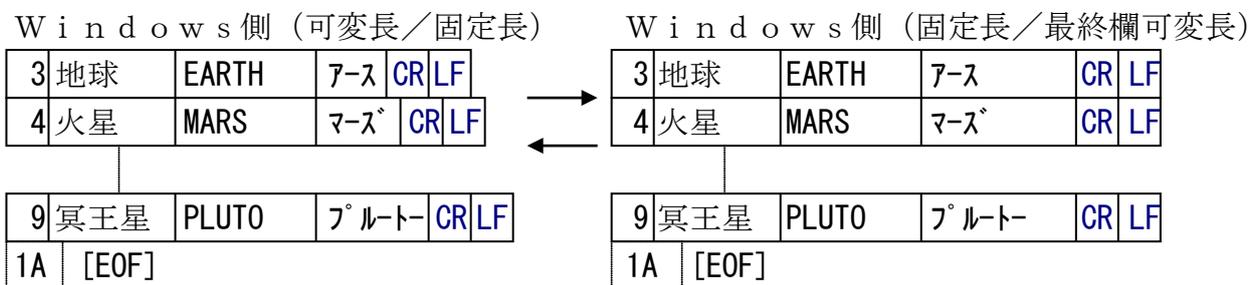
テキストファイルの加工に使います。生データの加工にも使えます。



● データファイル変換1 / プリント形式

おもに、Windows市販ソフトの入出力用テキストファイルの加工に使います。

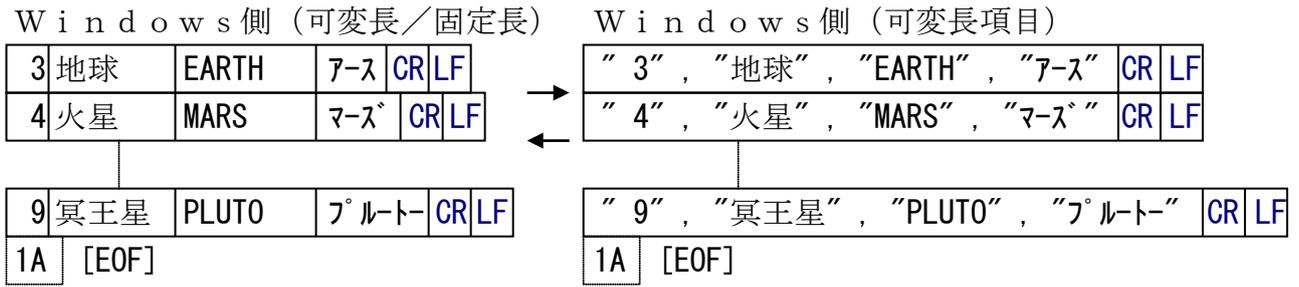
Windowsの可変長（プリント / デリミタ形式）または固定長のテキストファイルを、項目別に分けて変換できます。



● データファイル変換2 / デリミタ形式

おもに、Windows市販ソフトの入出力用テキストファイルの加工に使います。

Windowsの可変長（プリント / デリミタ形式）または固定長のテキストファイルを項目別に分けて変換でき、コンマ区切り形式、タブ区切り形式、スペース区切り形式の3タイプが扱えます。



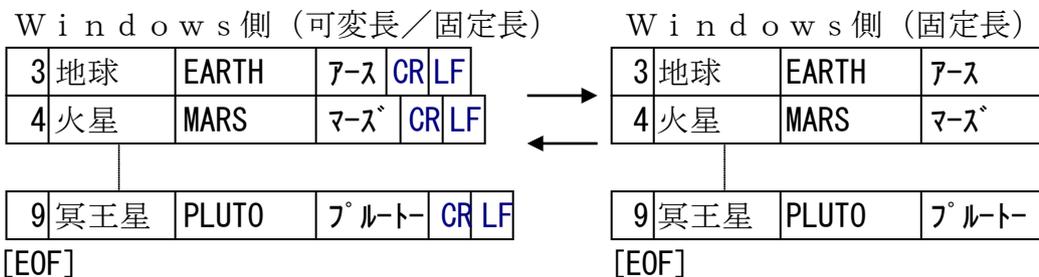
一般に、市販ソフトで「テキストファイルの入出力ができます」といえば、以上のうちコンマ区切り形式か、プリント形式ファイルを指すのがふつうです。

とくに、コンマ区切り形式は、パソコンのデータベースや表計算ソフトの最も標準的な入出力ファイル形式で、CSV (Comma Separated Value) 形式という特別な名前がついています。この形式は、市販ソフトへの入力操作がとても簡単なのが特長です。

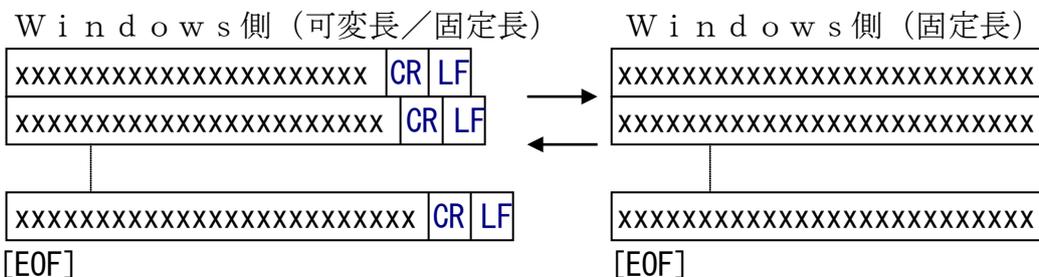
●ランダムファイル変換

おもに、Windows COBOLの順ファイルやBASICのランダムファイルの加工に使います。

Windowsの可変長 (プリント/デリミタ形式) または固定長のテキストファイルを項目別に分けて変換できる点では、データファイル変換1/プリント形式とよく似ています。しかし、改行コードも区切り文字もなく、固定長で、レコード長を意識しながら扱う必要があります。



バイナリ変換 (レコード全体/一部を無変換で通すこと) もできます。



1.7 補助機能

F*TRAN+は、補助機能を提供しています。ホストファイルやWindowsファイル进行管理します。

●Winファイラ

Winファイラはつぎの機能を持っています。

①ファイル名一覧

ホストファイルやWindowsファイルの一覧を見ることができます。

②ファイルの削除

ホストファイルやWindowsファイルを削除できます。

③ファイル編集

ホストファイルやWindowsファイルの中身の表示・修正ができるファイルエディタです。

1. 8 セットアップ機能

F*TRAN+は、このセットアップ機能でさまざまなホストへの対応を実現しています。コード変換の方法を設定する機能がその中心です。F*TRAN+を購入しインストールをすませたら、必ずセットアップの作業をしてください。セットアップはホスト選択と変換設定により行います。

●ホスト選択

ホスト系を、汎用機・オフコン系（汎用機またはオフコン）、Unix系（Unix機）、Windows系（Windows機）の3つから選択し、使用するホストに適するコード変換表を選択します。

●ANKコードの設定

ホストが汎用機・オフコンの場合、ホストで採用しているANKコード系がEBCDICコードかJIS8/ASCIIコードかを設定します。この設定はANK変換のときだけでなく、バッファのクリア、ゾーン形式やパック形式の数値の変換、その他でも参照される、とても重要なものです。

EBCDICコードのカタカナ版/英小文字版の切り替えもできます。

●コード変換方式の設定

ホストで採用しているコード体系がどの方式かを設定します。ふつうは、F*TRAN+が標準提供しているコード変換方式の中から、どれか1つを選ぶだけで済みます。

各コード変換方式の設定内容はすべて修正可能です。たとえば、漢字対応表に定義されていない拡張漢字の扱い方の変更などができます。

●漢字対応表の設定

ホストの拡張漢字や外字（ユーザー定義文字）などを、利用者の意図した漢字に割り当てるための漢字対応表を設定します。Unicodeとホストコードの対応では任意の範囲を外字領域とすることもできます。漢字対応表に登録された内容が変換時の漢字コードに反映されます。普通は、標準提供の漢字対応表を利用します。

●ホストCOBOLベンダ/ホスト種別の設定

ホスト上で使用しているCOBOLがどのベンダかを設定します。この設定は、Unix、WindowsのCOBOLデータと、Windows COBOLデータの変換をする場合に重要になります。

ホストが汎用機・オフコンの場合はホスト種別の選択が可能です。ホスト機の中では、IBM社のオフコン（eServer iSeries、AS/400）において、ゾーン形式・パック形式の符号表現が他と若干異なっています。一般的な設定かIBM iSeries、AS/400かを設定します。

●Windows COBOLベンダの設定

Windows上で使用しているCOBOLがどこのベンダかを設定します。この設定は、Windows COBOLで使用するデータを変換する場合に重要になります。

●ホストエンディアンの設定

ホストの2進数値項目の格納順序（正順、逆順）を設定します。この設定は、ホストのCOBOLの2進数値項目データ等を変換する場合に重要になります。また、ホストがWindowsでコード系がUnicode（UCS-2）の場合、コードの格納順序の指定としても使用されます。

●コメントの変更

設定を変更したら、適当なコメントをつけておくことができます。

●ANK変換表

多くのホストはEBCDICコードを採用しています。一方、パソコンはJIS8/ASCIIコード、またはUnicodeです。両者のコード変換の方法には確固とした標準がありません。

そこで、F*TRAN+ではほぼ標準的と思われるANK変換表を提供し、それを利用者が個々の事情に合わせて修正できるようにしています。修正操作はとても簡単です。

<コード変換表>

以上の設定は、ファイルに保存できます。それを「コード変換表（ファイル）」と呼んでいます。拡張子は、CCT（Code Conversion Table）で、F*TRAN+の動作を決める重要なファイルです。複数のホストとのデータ交換が必要なときは、ホストの種類に応じてこれを何通りかにセットアップしておき、使い分けることができます。

●ログ設定

ログの出力に関する設定を行います。出力先を変更する事も可能です。

<環境設定>

コード変換表以外にF*TRAN+の動作に必要なファイルとして、漢字対応表ファイル(*.KKT)、FTRAN.INIファイルがあります。これらのファイルは初期状態として、ユーザデータフォルダ配下にあるファイルが使用されますが、別のフォルダにあるファイルを使用する事も可能です。この設定は環境設定で行います。

1. 9 特長のまとめ

F*TRAN+の特長を、以下にまとめました。

●ファイル変換がパワフルである

マルチファイル処理機能と、ハードディスク上での変換のため、大量・大容量のファイルが容易に扱えます。

●実質4通りのファイル変換機能をもつ

テキストファイル変換、データファイル変換、ランダムファイル変換と呼ぶ3通りの変換ができます。データファイル変換は、さらにプリント形式のファイルの変換とCSV形式に代表されるデリミタ形式のファイルの変換に分けられます。つまり、実質4通りのファイル変換機能が用意されており、それらを目的に応じ柔軟に使い分けできます。これにより、標準的なエミュレータのテキスト転送ではできない変換の要求に対応することができます。

●マルチファイル対応で、ワイルドカード文字が使える

Windowsファイルの指定にも、ホストファイルの指定にもワイルドカード文字(*、?)が使い、マルチファイル処理ができます。関連ファイルの一括処理が容易です。ワイルドカードは、ファイル名が固定していないとき、短い指定ですませたいときにも役に立ちます。

●問い合わせモードがある

ファイルの変換を実行するかどうか問い合わせるモードがあります。ワイルドカード文字を使ったとき、ファイル名を確認しながら安全に処理を進めることができます。

●ウインドウ設定方式とコマンド行方式を両立させている

ウインドウ設定方式では、ウインドウ上の項目を選択・入力するだけの簡単な操作での実行が可能です。試行錯誤が容易で、初心者にも向いています。また、コマンド行方式で使えば、定形処理のバッチファイル化が容易に行えます。Visual Basicなどからも呼び出すことができ、定形業務化や組み込み用途にも向いています。終了コード(エラーレベル)で正常・異常の判断ができ、バッチファイルのなかである程度のエラー処理ができます。

●OLEオートメーション対応

OLEオートメーションに対応していますので、F*TRAN+をユーザアプリケーションの「部品」として使用することができ、より細かいコントロールをすることができます。

●スクリプトファイル機能をサポート

一度、設定/実行した変換作業をスクリプトファイルとして保存しておくことができます。このスクリプトファイルを使って、試行錯誤(変換設定の修正)や自動実行(エクスプローラ等からスクリプトファイルを起動して変換を実行させること)も容易にできます。

●パワフルで使いやすいレコードレイアウト設定機能

マップ設定オプションで細かい変換方法を指示できます。マルチレコードレイアウトもサポートしています。設定はウインドウ上の項目を選択・入力する方式ですので、簡単に行うことができます。設定時にホストファイルやWindowsファイルの内容を確認することができます。さらに、設定に従って変換をした場合の結果をシミュレートすることもできます。

●ヘッダやトレーラを付け加えることができる

Windowsファイルにしてもホストファイルにしても、固定的なヘッダ(見出しレコード)やトレーラ、エンドレコードなどを必要とするものは多いものです。それらを本体ファイルの変換と同時に付け加えることができます。

これにより、Windowsファイルの前処理・後処理の必要性を、大幅に減らすことができます。

●定数を挿入することができる

ファイル変換の際に、各種の型(ANKや漢字の文字列、バイナリデータ、表示形式、ゾーン形式、パック形式、2進項目など)の任意の固定データ(定数)を挿入することができます。たとえばCSV形式に変換するときに、あらかじめExcelの式を埋め込むこともできます(ただし、どんな式でもというわけではありません)。LOW-VALUE(00)やHIGH-VALUE(FF)、空白を埋め込むことも容易です。

●COBOLのCopy句(登録集)を読み込むことができる

COBOLのCopy句(登録集のメンバ)から、マップ設定のひな型を自動生成できます。変換したいファイルを定義しているCopy句が入手できるときは、マップ設定の手間を大幅に減らすことができます。

多くのホストのCOBOL、Windows COBOLに対応しています。ソースの形式においては、固定形式、可変形式、自由形式をサポートするなど、一部では次期COBOLの仕様を先取りしています。

項目名を見出しレコードに反映させる機能もあります。

●コード変換表をカスタマイズできる

ANK変換表やコード変換方式などを設定するコード変換表を、利用者が自由にカスタマイズできます。自社用にカスタマイズしたコード変換表をいくつか作っておき、それを切り替えながら使うこともできます。

●複数の環境を作成できる

F*TRAN+で使用する設定ファイル類をフォルダ単位でまとめて複数保存できます。F*TRAN+の起動時にこのフォルダ群のどれを使用するかを指定でき、たとえばさまざまな作業グループ毎に複数の環境を切り替えながらの使用が可能です。

●各種コード変換方式をサポートしている

さまざまなコード変換方式をサポートしています。K I（漢字イン）／K O（漢字アウト）があるときでも、K I／K Oがないときでも、それぞれに適した変換ができます。

コード変換方式として、各種の方式をコード変換表に登録しておくことができ、きめ細かな調整ができます。

コード変換のレパートリーは、富士通J E F（12ポイント／9ポイント）、富士通F 9 4 5 0内部コード、日立K E I S、NEC内部コード（E／J）、NEC J I P S（E／J）、東芝漢字、三菱M E L C O M漢字、日本ユニシスL E T S - J、カシオ漢字、新J I S（83年版）、旧J I S（78年版）、I B M漢字、E U C、シフトJ I S、U n i c o d e（U C S - 2）、U n i c o d e（U T F - 8）などです。

F * T R A N +のサポートする基本パターンに合うなら、さらに新しいコード変換方式を追加登録することもできます。

●拡張漢字変換をサポートしている

ホストの拡張漢字（ユーザー定義文字を含む）にも対応しており、おもなホストベンダ（富士通、日立、NEC、東芝、三菱、日本ユニシス、日本I B M、サンマイクロ、日本H P 他）の漢字対応表を標準提供しています。漢字対応表は漢字対応表エディタ上で編集できるので、利用者の意図した漢字を対応づけることができます。また、任意の領域を外字領域として定義できます。漢字対応表に定義されない範囲外の漢字や外字領域として指定した領域の漢字は、ゲタ変換、クエスチョン変換と呼ぶ2通りの変換方式のなかから1つを選んで変換できます。

●ホストのゾーン形式、パック形式をサポートしている

ホストのCOBOLの内部数値形式であるゾーン形式（符号なし、符号つき）、パック形式（符号なし、符号つき）の変換をサポートしています。小数点もサポートしています。

J I S 8系（東芝のシステム）のゾーン形式、パック形式にも対応しています。

●POS端末採用のBCD形式数値をサポートしている

POS端末などで採用しているBCD形式数値（符号なし）をサポートしています。

●Windows COBOLのゾーン形式、パック形式をサポートしている

Windows COBOLのゾーン形式（符号なし、符号つき）、パック形式（符号なし、符号つき）、BCD形式数値（符号なし）の変換をサポートしています。それらとホストのゾーン形式（符号なし、符号つき）、パック形式（符号なし、符号つき）、BCD形式数値（符号なし）の変換ができます。

●日付データ変換をサポートしている

日付データを指定フォーマット（YYYYMMDD、YY-MM-DDなど）で編集して変換する機能をサポートしています。西暦年下2桁、和暦年のデータを指定範囲の西暦年4桁に拡張する機能もあり、西暦2000年対応にも幅広く活用できます。

●2進数値変換をサポートしている

COBOL、Visual Basic、C/C++などで定義できる2進数値をサポートしています。それらと文字形式、ゾーン/パック形式数値の変換ができます。

●バイナリ変換をサポートしている

バイナリ変換をサポートしています。たとえば、ファイルの一部にふつうの変換方法が適用できないデータがあるときなどに、最後の手段としてバイナリ変換を使うことができます。

●Windowsファイルの編集/加工ができる

プリント形式ファイル（可変長/固定長）、デリミタ形式ファイル（可変長）、ランダムファイル（固定長）間のデータ編集/加工が自由にできます。

●ファイル変換中に処理の中断ができる

ファイル変換中に中止ボタンをクリックすれば、いつでも処理を中断することができます。

●Winファイラ、Winファイルエディタを内蔵している

Winファイラは、ホストファイルやWindowsファイルの一覧を見ることができ、ホストファイルやWindowsファイルの削除もできます。WinファイラからWinファイルエディタを起動できます。

Winファイルエディタを使えば、ホストファイルやWindowsファイルの内容表示・修正ができます。マップ設定オプションと並列に動作させ、データの確認をしながらレコードレイアウトを設定することもできます。

第2章



変換の仕様

2. 1 Unicodeについて

F*TRAN+ではWindows側のコードとしてシフトJISのほかにUnicodeをサポートしています。Unicodeは、UCS-2とUTF-8の2種類が扱えます。さらにUCS-2はエンディアンによりUCS-2 LE、UCS-2 BEの2種類をサポートしています。

●UCS-2について

UCS-2のコード体系は以下のようになっています。

0000	ASCII互換
0080	各国アルファベット類
2000	記号
3000	CJK表音文字と記号
3400	CJK統合漢字
A000	
D800	サロゲート
E000	私用領域
F900	全角英数・記号、半角カタカナ等
10000	

- 0000～007F ASCII 互換文字
- 0080～1FFF 各国アルファベット
- 2000～2FFF 記号
- 3000～33FF CJK表音文字と記号
- 3400～9FFF CJK統合漢字

- US-ASCIIと互換性を持たせた領域。
- 各国アルファベット。
- 様々な記号類。
- ひらがな、カタカナ等。
- CJKの漢字領域。中国・日本・韓国の漢字がこの領域に重ねて定義されている。基本的にこの3国で同じ字形を持つ漢字は同じコードを共有している。

D800～EFFF サロゲート UCS-2を16面分拡張するためのコード
 E000～F8FF 私用領域 いわゆる外字領域。先頭の1880文字分はシフトJISの外字領域と1対1に対応している。
 F900～FFFF 全角英数記号／半角カタカナ シフトJISと互換をとるために、ここに全角英数・記号や半角カタカナが定義されている。

※C J K : C h i n a (中国)、J a p a n (日本)、K o r e a (韓国) の略

UCS-2では『1234H』は『U+1234』とコードの頭に『U+』を追加して表現します。UCS-2では半角文字や全角文字、漢字など全ての文字を2バイトで表現します。

UCS-2は2バイト(16ビット)で1つの文字を表しますが、UCS-2は、この2バイトの上位バイト、下位バイトの並び順により2種類あります。それぞれ『Big Endian (BE、ビッグエンディアン)』、『Little Endian (LE、リトルエンディアン)』と呼ばれています。例えば16進で『1234H』がBEならそのまま『1234H』と格納されますが、LEなら『3412H』と格納されています。それぞれファイルの先頭にBOM(バイトオーダーマーク)をつけて区別します。BOMが『FEFFH』ならBE、『FFFEH』ならLEです。なお、ファイルによってはBOMがない場合もあります。

●UTF-8について

UTF-8は、UCS-2などで定義された文字をバイト列として表現した文字コードの1つで、文字により1バイト～3バイトで表現されています。UCS-2とUTF-8の変換方法は以下の通りです。

UCS-2	ビットパターン	第1バイト	第2バイト	第3バイト
U+0000～U+007F	00000000-0xxxxxxx	0xxxxxxx		
U+0080～U+07FF	0000xxx-xyyyyyyy	110xxxxx	10yyyyyy	
U+0800～U+FFFF	xxxxyyyy-yyzzzzzz	1110xxxx	10yyyyyy	10zzzzzz

たとえば以下ようになります。

半角英字『A』はUCS-2でU+0041ですが、これをビットパターンで表すと00000000-01000001となります。U+0041は最初のパターンの範囲なのでこのビットパターンは01000001となり、16進で表すと41となります。この41がUTF-8での表現となります。

U+0000からU+007FはASCII互換部分です。UTF-8ではこの範囲はそのまま1バイトで表されるのでASCII文字とは完全に互換性があります。

漢字の『富』はU C S - 2 でU+5BCCですが、これをビットパターンで表すと01011011-11001100となります。U+5BCCは3番目のパターンの範囲なのでこのビットパターンは11100101-10101111-10001100となり、16進で表すとE5-AF-8Cとなります。このE5-AF-8CがU T F - 8での表現になります。

半角カタカナも3番目のパターンに入ります。半角カタカナや漢字をU T F - 8に変換すると3バイトで表されます。

U T F - 8にはエンディアンの違いはありませんが、ファイルの先頭にBOMとして『**E F B B B F H**』をつけてファイルがU T F - 8である事を表す場合があります。ただし、基本的にはU T F - 8の場合はBOMは付けません。

2. 2 漢字シフトコードK I / K Oについて

以降の説明で、頻繁にK I（漢字イン）、K O（漢字アウト）という用語がでてきます。単にK I / K Oと表記しますが、F * TRAN+は実際にはホストの3つの漢字シフト方式をサポートしています。ホストが汎用機・オフコン系（汎用機またはオフコン）の時に多種多用に使用されています。

● K I / K O方式

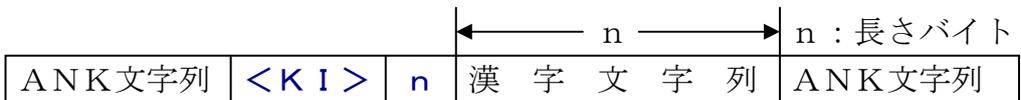
狭い意味のK I / K O方式のことで、漢字文字列をK IとK Oではさみます。一番多く使われている方式です。



● K I + 長さバイト方式

東芝のシステムで使われている方式です。漢字文字列のおわりを示すK Oがない代わりに、K Iのあとにその長さ（通常2～254）を示すバイトをつけます。

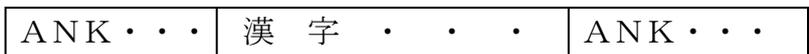
この方式は、少し取り扱いが難しいのが欠点です。



● NEC暗黙シフト方式

NECの内部コード（E / J）で使われている方式です。K I / K Oをしません。文字コードを調べて、それがANK文字か漢字の第一バイトかを判定します。

パソコンのシフトJISコードと、少し似ていますが、それほど使いやすくはありません。また、英小文字と数個の特殊文字が使えないという大きな欠点も持っています。



●EUCシフトコード

Unix系のオペレーティングシステムでは、漢字コードとしてEUC (Extended Unix Code) が使われる場合がほとんどですが、EUCには1バイトの半角コードと、2バイトの漢字コードの他に、つぎのコードが存在します。

半角カナコード (2バイト) : 8EH + A1H~DFH

拡張漢字コード (3バイト) : 8FH + A1A1H~FEFEH

F*TRAN+では、この 8EH、8FH をEUCシフトコードとして認識し、処理しています。

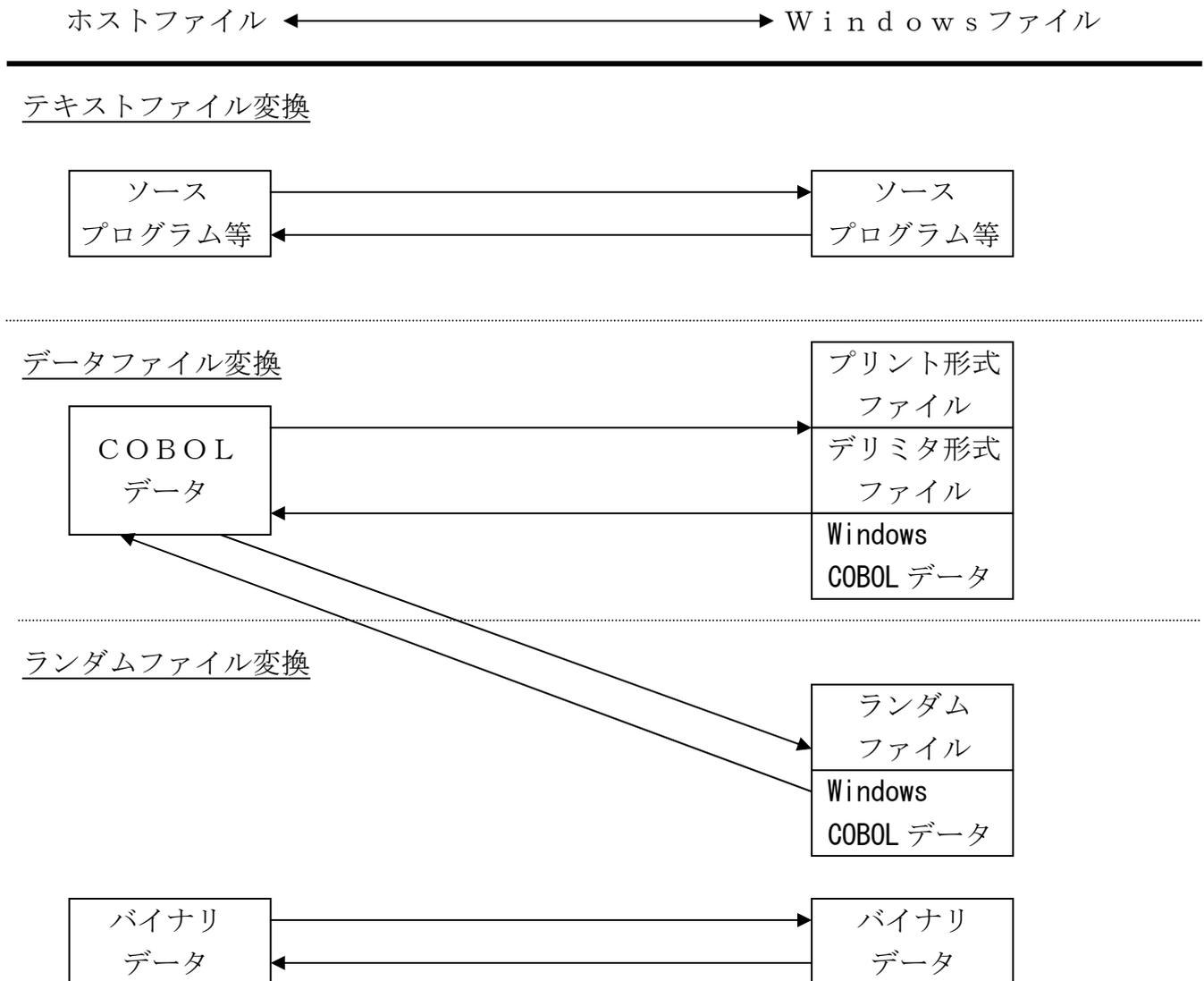
2. 3 ファイル変換機能の使い分け

F*TRAN+では、3つの（実質4つ）のファイル変換機能、

- テキストファイル変換
- データファイル変換1 / プリント形式
- データファイル変換2 / デリミタ形式
- ランダムファイル変換

を使い分けることができます。“うまく使い分けなければいけない” といったほうが、正しいかもしれません。

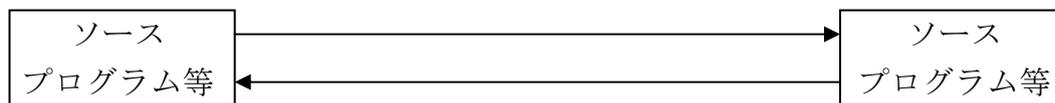
● 典型的なデータ形式（ホストが汎用機・オフコンの場合）



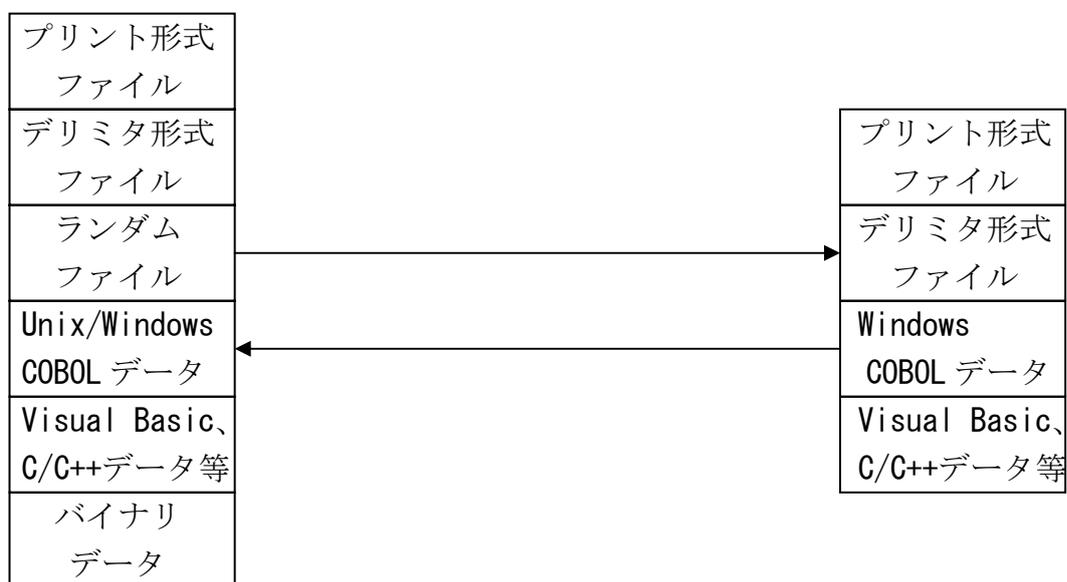
●典型的なデータ形式（ホストがUnix、Windowsの場合）

ホストファイル ←-----> Windowsファイル

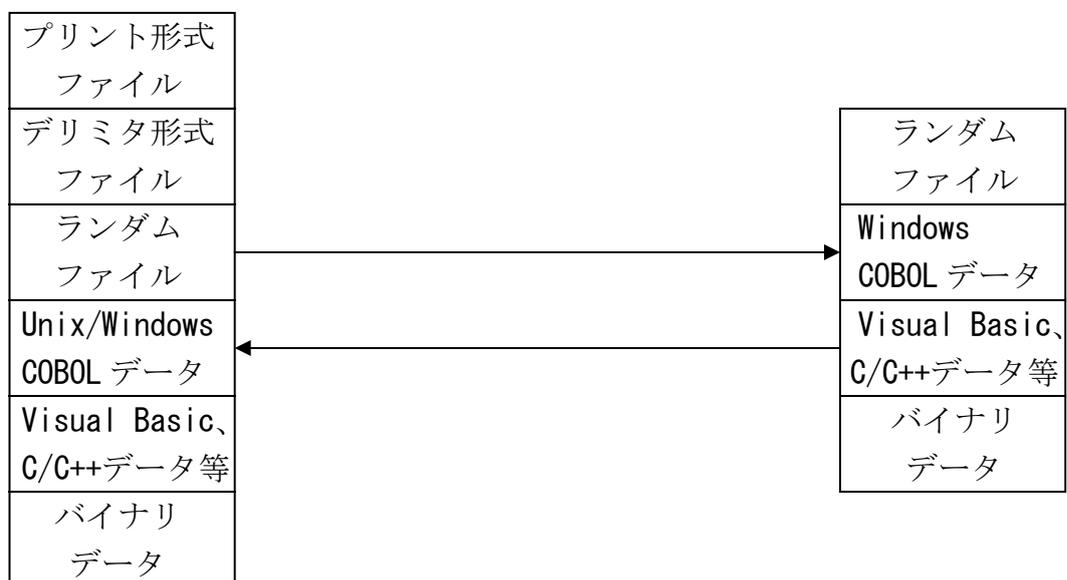
テキストファイル変換



データファイル変換



ランダムファイル変換



2. 4 ファイル変換機能の使い方のポイント

■テキストファイル変換

テキストファイル変換は、おもに、ソースプログラムの変換に使用します。

●ホスト→Winテキストファイル変換

ホストが汎用機・オフコンの場合、コードがAnkかAnk・漢字(KI/KO付)混在かを指定するコード設定オプションが重要です。

●Win→ホストテキストファイル変換

ホストが汎用機・オフコンの場合、コードがAnkかAnk・漢字(KI/KO付)混在かを指定するコード設定オプションがとくに重要です。

Winファイル側の各レコード末尾に、必ず改行コード(CR/LF=0D0AH)がついていなければいけません(ただし、最終レコードに限り改行コードがなくてもよい)。

■データファイル変換1 / プリント形式

データファイル変換1 / プリント形式は、おもに、ホストのCOBOLのデータと市販ソフトの入出力データ形式(プリント形式≒固定長テキスト)の変換に使用します。

●ホスト→Winデータファイル変換

項目別に詳細な変換指定をするマップ設定オプションがとくに重要です。

●Win→ホストデータファイル変換

項目別に詳細な変換指定をするマップ設定オプションがとくに重要です。

Winファイル側の各レコード末尾に、必ず改行コード(CR/LF=0D0AH)がついていなければいけません(ただし、最終レコードに限り改行コードがなくてもよい)。

■データファイル変換2／デリミタ形式

データファイル変換2／デリミタ形式は、おもに、ホストのCOBOLのデータと市販ソフトの入出力データ形式（デリミタ形式）の変換に使用します。コンマ区切り（CSV）形式の変換がとくに重要です。

●ホスト→Winデータファイル変換

項目別に詳細な変換指定をするマップ設定オプションがとくに重要です。

●Win→ホストデータファイル変換

項目別に詳細な変換指定をするマップ設定オプションがとくに重要です。

Winファイル側の各レコード末尾に、必ず改行コード（CR/LF=0D0AH）がついていなければいけません（ただし、最終レコードに限り改行コードがなくてもよい）。

■ランダムファイル変換

ランダムファイル変換は、おもに、ホストのCOBOLのデータとBASICなどのランダムファイル（ただの固定長ファイル）の変換に使用します。バイナリ変換もできます。

●ホスト→Winランダムファイル変換

項目別に詳細な変換指定をするマップ設定オプション、Winファイル側のレコード長を指定する出力レコード長オプションがとくに重要です。

●Win→ホストランダムファイル変換

項目別に詳細な変換指定をするマップ設定オプション、Winファイル側のレコード長を指定する入力レコード長オプションがとくに重要です。

2.5 ホスト側の扱えるデータ形式

ホスト側のデータ形式をどう扱えるか示します。◎○△の意味はつぎのとおりです。

- ◎ 直接扱える
- 直接扱えるものと扱えないものがある
- △ 直接は扱えない。前処理や後処理が必要

ソースプログラム（固定長 *1）	○
漢字あり	○
サポートしているタイプ	◎
サポートしていないタイプ	△
漢字なし	◎
COBOLの順ファイル（固定長 *1）	○
漢字あり	○
サポートしているタイプ	◎
サポートしていないタイプ	△
KI/KOあり	◎
KI/KOなし	◎
漢字なし	◎
数値項目（ゾーン形式、パック形式）あり	◎
数値項目（ゾーン形式、パック形式）なし	◎
その他の数値項目あり	○
その他の数値項目なし	◎
日付データ項目あり	○
日付データ項目なし	◎
レコードタイプあり	○
レコードタイプなし	◎
COBOLのISAMファイル	△
データベースからの抽出データ	◎
その他の言語の入出力ファイル	○
文字データのみ	◎
バイナリデータあり	○
POS端末のBCD形式数値項目	○
各種バイナリファイル	○
・	
・	

*1) ホストがUnix、Windowsの場合は、可変長／固定長になる

2.6 Windows側の扱えるデータ形式

Windows側のデータ形式をどう扱えるか示します。◎○△の意味はつぎのとおりです。

- ◎ 直接扱える
- 直接扱えるものと扱えないものがある
- △ 直接は扱えない。前処理や後処理が必要

ソースプログラム（可変長）	◎
テキストファイル形式の文書ファイル	○
行末に改行コードを入れたもの	◎
行末に改行コードを入れていないもの	△
バイナリ文書ファイル	△
プリント形式ファイル（固定長テキスト形式、SDF形式）	◎
デリミタ形式ファイル	○
コンマ区切り（CSV）形式	◎
固定長／可変長項目	◎
全項目引用符くくり	◎
文字項目引用符くくり	◎
K3フォーマット（見出しつき）	◎
K3フォーマット（データのみ）	◎
コンマを含む項目のみ引用符くくり	◎
引用符なし	◎
タブ区切り形式（可変長項目）	◎
文字項目引用符くくり	◎
引用符なし	◎
スペース区切り形式（可変長項目）	◎
全項目引用符くくり	◎
文字項目引用符くくり	◎
空白を含む文字項目のみ引用符くくり	◎
引用符なし	◎
ランダムファイル（ただの固定長ファイル）	○

BASICのランダムファイル	○
2進数値項目あり（整数）	◎
2進数値項目あり（浮動小数点）	△
2進数値項目なし	◎
レコードタイプあり	◎
レコードタイプなし	◎
Windows COBOLの順ファイル/行順ファイル	○
数値項目（ゾーン形式、パック形式）あり	○
数値項目（ゾーン形式、パック形式）なし	◎
その他の数値項目あり	○
その他の数値項目なし	◎
日付データ項目あり	○
日付データ項目なし	◎
レコードタイプあり	◎
レコードタイプなし	◎
C/C++のデータ形式ファイル	○
各種ISAMファイル	△
データベースや表計算ソフトの内部形式ファイル	△
その他の市販ソフトの内部形式ファイル	△
POS端末のBCD形式数値項目	○
各種バイナリファイル	○
ダウンロード/アップロードファイル	○
.	
.	
.	

2. 7 テキストファイル変換の仕様

■ホストが汎用機・オフコンの場合

ホストが汎用機・オフコンの場合、テキストファイル変換ではホストファイル（固定長レコード）とWindowsテキストファイル（可変長レコード）間の変換を行います。1レコードが1行に対応する改行コード付きのデータが対象です。ソースプログラムがその典型です。

以下に、テキストファイル変換のコード変換機能、レコード編集機能、EOFコードの扱い、BOMコードの扱いを示します。

●コード変換機能

テキストファイル変換のコード変換機能は、下表に示すとおり比較的単純です。

コード変換機能

ホスト側コード	↔	Win側コード	変換方法
J I S 8 / A S C I I	↔	J I S 8 / A S C I I、漢字なし	ANK変換
E B C D I C (カタカナ)		U n i c o d e (U C S - 2)、漢字なし	"
E B C D I C (英小文字)		U n i c o d e (U T F - 8)、漢字なし	"
ANK・漢字混在 K I / K O 付き	↔	J I S 8、シフトJ I S 漢字混在 U n i c o d e (U C S - 2)、漢字あり U n i c o d e (U T F - 8)、漢字あり	ANK・漢字 混在変換

◆注意 ---- K I / K O がついていないと漢字変換はできない

ホスト→Winテキストファイル変換のときは、K I / K O がついていないと漢字の変換はできません（NEC内部コード（E / J）を除く）。K I / K O がないときは、代わりにホスト→Winデータファイル変換を使って、項目別に分けて変換する必要があります。

●レコード編集機能

テキストファイルの代表であるソースプログラムを例に、レコード編集機能を説明します。

ホストのソースプログラムは、レコード長は80バイト（80欄カードイメージ）が中心で固定長である、改行コードに相当するものはない、タブは使わない、などの特徴があります。漢字を使わない限りホスト間での互換性は高いといえます。でも、漢字を使うと、表示イメージと実際の記録形式が違い桁位置がずれていく（K I / K O が入るため）、システム間で互換性がなくなる、などの欠点もあります。

一方、パソコン側のソースプログラムは、改行コード（C R / L F = 0 D 0 A H）でレコードの区切りとし、可変長である、レコード長にはあまり制限がない、タブ（T A B = 0 9 H）が自由に使える、スペース効率がよい、漢字が自由に使えて半角・全角の区別をあまり意識する必要がない、といえます。

ホスト文化とパソコン文化の違い、とでもいったらいいでしょうか。このギャップを埋めるために、かなりのレコード編集機能が必要です。それを下表に示します。

レコード編集機能

機 能	説 明
改行コード付加	無条件にレコード末尾に改行コードを付加する(ホスト→Win)
改行コード削除	無条件にレコード末尾の改行コードを削除する(Win→ホスト)
行末圧縮	無条件に行末のスペース類を切り捨てる(ホスト→Win)
行末伸長	無条件に行末にスペースを詰める(Win→ホスト)
タブ圧縮	連続スペースをタブに置き換える(ホスト→Win)
タブ拡張	タブを連続スペースに置き換える(Win→ホスト)

●EOFコードの扱い

F*TRAN+のテキストファイル変換では、ホスト→Win方向の変換時にはEOFコード(1AH)をつけない、Win→ホスト方向の変換時にはEOFコード検出でも物理EOF検出でもEOFにする、というのが省略時の動作です。

EOFコードの扱い

機 能	説 明
EOFコード付加	ファイル末尾にEOFコードを付加する・しない(ホスト→Win)
EOFコード検出	EOFコード検出でEOFにする(Win→ホスト)
EOFコード透過	EOFコードを一般の文字扱いする(Win→ホスト)

●BOMコードの扱い

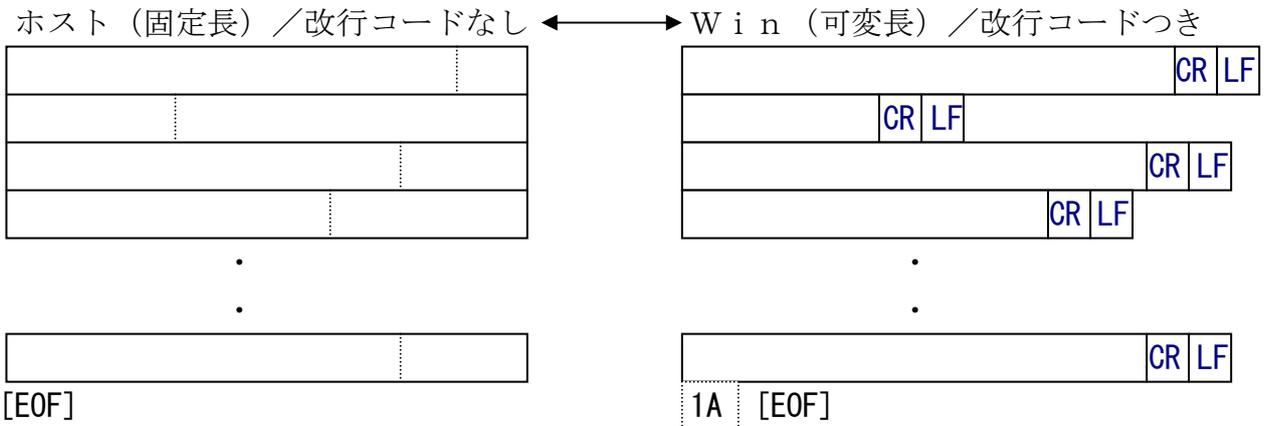
Windows側のコードとしてUnicodeを選択した場合、BOM(バイトオーダーマーク)の扱いについての指定が可能です。

F*TRAN+のテキストファイル変換では、ホスト→Win方向の変換時でコードがUCS-2の時にはBOMコードをつける、UTF-8の時にはBOMコードをつけない、Win→ホスト方向の変換時でコードがUCS-2の時には入力ファイルにBOMコードあり、UTF-8の時には入力ファイルにBOMコードなしとして変換する、というのが省略時の動作です。

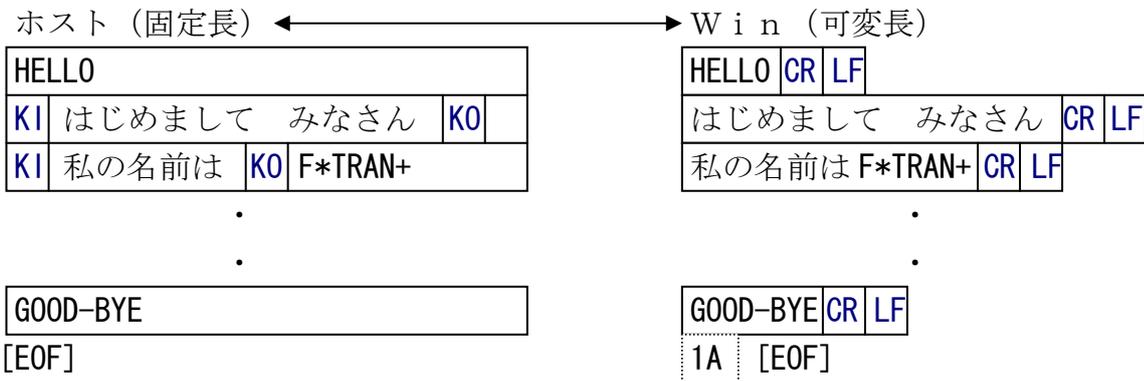
BOMコードの扱い

機 能	説 明
BOM自動	UCS-2の時BOMあり、UTF-8の時BOMなし
BOMあり	BOMを付加する(ホスト→Win) BOMを検査する(Win→ホスト)
BOMなし	BOMを付加しない(ホスト→Win) BOMを検査しない(Win→ホスト)

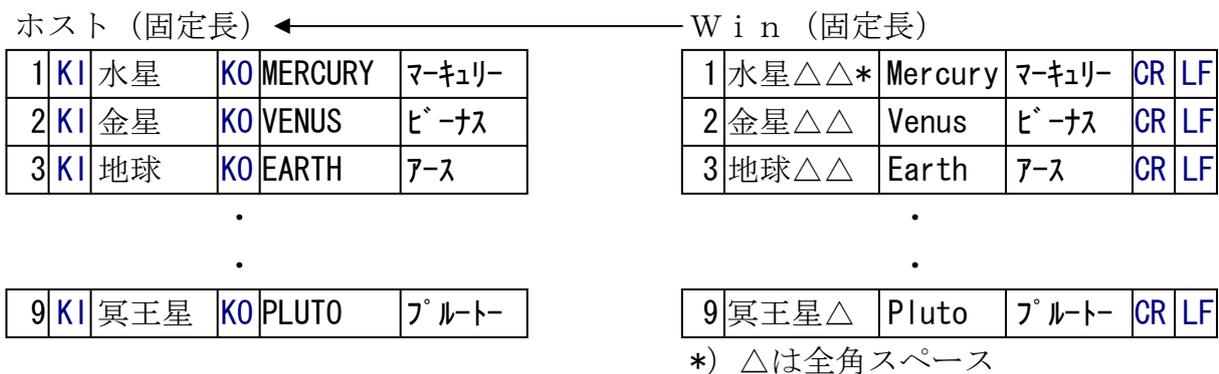
例1) ふつう、固定長⇔可変長の変換になる



例2) ANKと漢字が混在なら、KI/KOの挿入・削除処理が入る



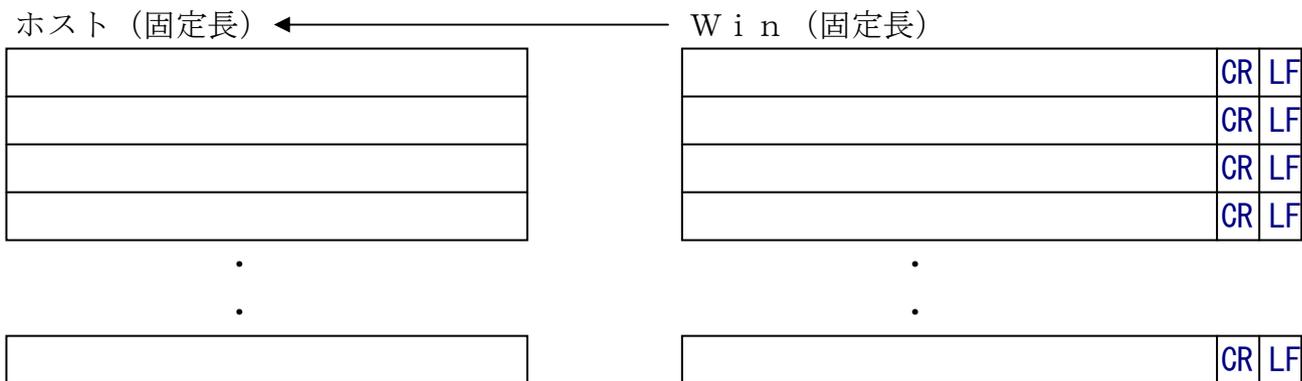
例3) 全角・半角スペースを使い分けていれば、Win→ホスト方向の変換にも使える



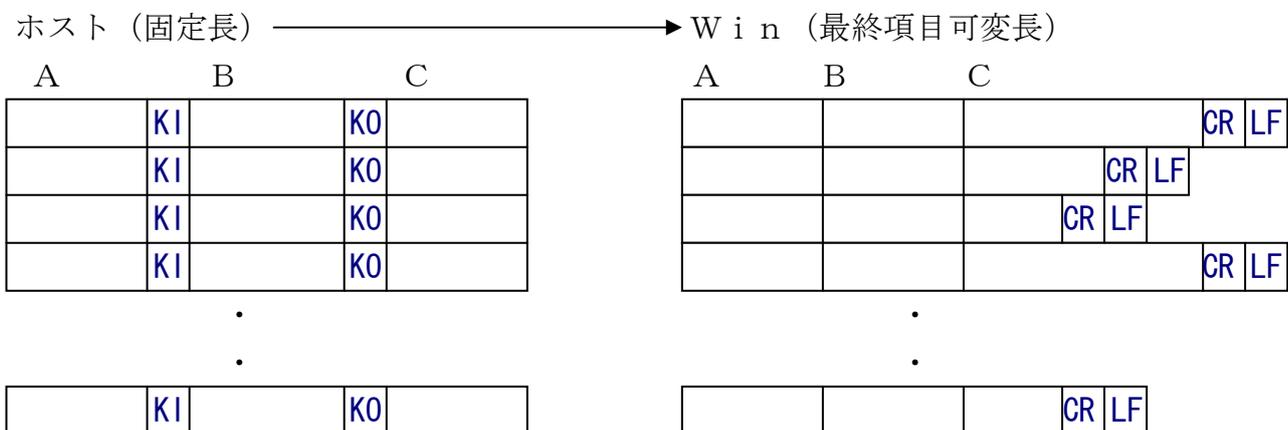
◆注意 ---- Win→ホストファイル変換時のシフト節約度を弱にすべし

この例3のような変換をするときは、変換設定/漢字変換方式の「Win→ホストファイル変換時のシフト節約度」を「弱」に設定してください。そうしないと、KOの位置がレコードごとにずれます。

例4) ANKのみで、Win→ホスト方向の変換なら、固定長変換もできる



例5) ANKと漢字のみでKI/KO付きのホストファイルは、項目別変換できる



■ホストがUnix、Windowsの場合

ホストがUnix、Windowsの場合、テキストファイル変換ではホストファイル（可変長レコード）とWindowsテキストファイル（可変長レコード）間の変換を行います。1レコードが1行に対応する改行コード付きのデータが対象です。ソースプログラム等がその典型です。

以下に、テキストファイル変換のコード変換機能、レコード編集機能、EOFコードの扱い、BOMコードの扱いを示します。

●コード変換機能

テキストファイル変換のコード変換機能は、下表に示すとおり比較的単純です。

コード変換機能

ホスト側コード	↔ Win側コード	変換方法
JIS8・漢字混在 (EUC、シフトJIS)	JIS8、シフトJIS漢字混在 Unicode (UCS-2) Unicode (UTF-8)	ANK・漢字 混在変換固定
Unicode (UCS-2)		
Unicode (UTF-8)		

●レコード編集機能

ソースプログラムを例に、レコード編集機能を説明します。

Unix、Windowsのソースプログラムは、改行コード（UnixはLF=0AH、WindowsはCR/LF=0D0AH）でレコードの区切りとし、可変長である、レコード長にはあまり制限がない、タブ（TAB=09H）が自由に使える、スペース効率がよい、漢字が自由に使えて半角・全角の区別をあまり意識する必要がない、といえます。

レコード編集機能

機能	説明
改行コード正規化	レコード末尾の改行コードを正規化する
行末圧縮	無条件に行末のスペース類を切り捨てる
タブ圧縮	連続スペースをタブに置き換える
タブ拡張	タブを連続スペースに置き換える

● E O Fコードの扱い

F * T R A N +のテキストファイル変換では、ホスト→W i n方向の変換時にはE O Fコード(1 A H)をつけない、W i n→ホスト方向の変換時にはE O Fコード検出でも物理E O F検出でもE O Fにする、というのが省略時の動作です。

E O Fコードの扱い

機 能	説 明
E O Fコード付加	ファイル末尾にE O Fコードを付加する・しない(ホスト→W i n *)
E O Fコード検出	E O Fコード検出でE O Fにする(W i n→ホスト *)
E O Fコード透過	E O Fコードを一般の文字扱いする(W i n→ホスト *)

*) ホストがW i n d o w sの場合は、ホスト⇔W i n

● B O Mコードの扱い

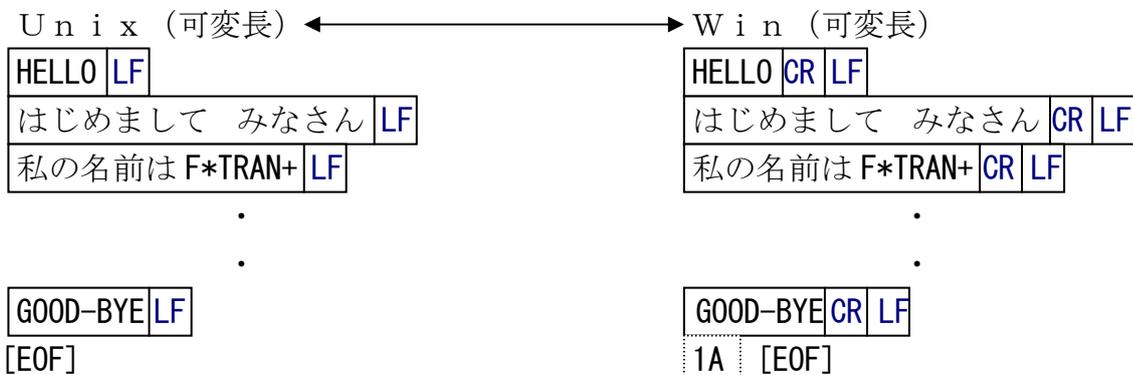
W i n d o w s側のコードとしてU n i c o d eを選択した場合、B O M(バイトオーダーマーク)の扱いについての指定が可能です。ホスト→W i n方向の変換時でコードがU C S - 2の時にはB O Mコードをつける、U T F - 8の時にはB O Mコードをつけない、W i n→ホスト方向の変換時でコードがU C S - 2の時には入力ファイルにB O Mコードあり、U T F - 8の時には入力ファイルにB O Mコードなしとして変換する、というのが省略時の動作です。

B O Mコードの扱い

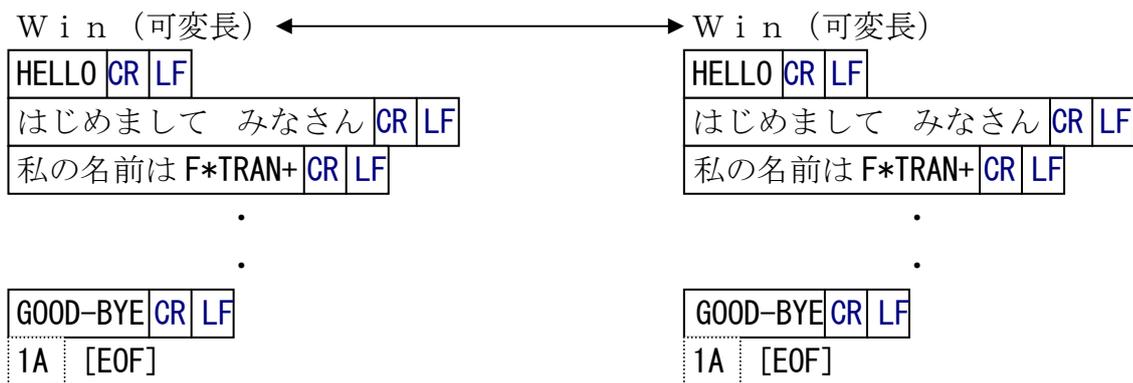
機 能	説 明
B O M自動	U C S - 2の時B O Mあり、U T F - 8の時B O Mなし
B O Mあり	B O Mを付加する(ホスト→W i n) B O Mを検査する(W i n→ホスト)
B O Mなし	B O Mを付加しない(ホスト→W i n) B O Mを検査しない(W i n→ホスト)

例) 可変長⇔可変長の変換になる

<ホストがUnixの場合>



<ホストがWindowsの場合>



2. 8 データファイル変換1 / プリント形式の仕様

■ホストが汎用機・オフコンの場合

ホストが汎用機・オフコンの場合、データファイル変換1 / プリント形式は、おもにホストのCOBOLデータと、市販ソフトの入出力形式の1つであるプリント形式ファイルの変換に使用します。

プリント形式ファイルとは、各レコードが改行コードで区切られている固定長のテキストファイルのことです。各項目の桁位置、幅が一定で、固定長・固定欄でなければいけません。なお、プリント形式ファイルはSDF形式ファイルとも呼ばれています。

例外として、Win→ホスト方向の変換のときは、最終項目に限り可変長でもかまいません。

以下に、データファイル変換1 / プリント形式で可能なデータ変換機能、レコード編集機能、EOFコードの扱い、BOMコードの扱いを示します。

●データ変換機能

データファイル変換1 / プリント形式のデータ変換機能を下表に示します。

データ変換機能		変換方法
ホスト側データ	↔ Win側データ	
JIS8 / ASCII	↔ JIS8 / ASCII	ANK変換
EBCDIC (カタカナ)	↔ Unicode (UCS-2)	〃
EBCDIC (英小文字)		〃
漢字 (KI / KOなし)		漢字項目変換
〃	↔ Unicode (UCS-2)	〃
〃	↔ Unicode (UTF-8)	〃
〃	← シフトJIS漢字、JIS8混在	〃
ANK・漢字混在	↔ { JIS8、シフトJIS漢字混在 Unicode (UCS-2) Unicode (UTF-8)	ANK・漢字混在変換
KI / KOつき		
文字形式数値	↔ 文字形式、Windows COBOL *1	ニューメリック変換 *3
符号なしゾーン形式	↔ 〃	ゾーン変換 *4
符号つきゾーン形式	↔ 〃	〃
符号なしパック形式	↔ 〃	パック変換 *5
符号つきパック形式	↔ 〃	〃
符号なしBCD形式	↔ 〃	〃
2進形式数値 *2	↔ 〃	2進数値変換*6
日付データ	↔ 日付データ	日付項目変換
特殊データ	↔ 特殊データ	特殊データ変換

- *1) Windows COBOL の符号なしゾーン形式、符号つきゾーン形式
- *2) COBOL の COMP-4 や COMP-5 などの形式
- *3) Numeric、DispZone、ZoneDisp、AlphaZone、ZoneAlpha 変換
- *4) ZoneDisp、DispZone、ZoneZone、ZoneAlpha、AlphaZone 変換
- *5) PackDisp、DispPack、PackZone、ZonePack、PackAlpha、AlphaPack 変換
- *6) BinDisp、DispBin、BinZone、ZoneBin、BinAlpha、AlphaBin 変換

●レコード編集機能

プリント形式ファイルは、基本的には固定長ファイルです。その意味では、ホストの標準的なデータ形式と似ています。しかし、項目単位の編集・加工ができないと使いものになりません。下表に、データファイル変換1 / プリント形式のレコード編集機能を示します。

レコード編集機能

機 能	説 明
タブ拡張	タブを連続スペースに置き換える(W i n→ホスト)
改行コード付加	無条件にレコード末尾に改行コードを付加する(ホスト→W i n)
改行コード削除	無条件にレコード末尾の改行コードを削除する(W i n→ホスト)
空行無視	W i n側の、改行・改頁コードのみの行を無視する・しない
項目別変換	項目別に分けて変換できる
項目削除	不要な項目を削除できる
項目長変更	項目長を変更できる
項目組み替え	項目の順番の組み替えができる
空項目挿入	空項目を作ることができる
定数挿入	任意の値を任意の形式で挿入することができる
K I / K O 挿入	漢字項目の前後にK I / K Oを挿入できる(W i n→ホスト)

●E O Fコードの扱い

プリント形式ファイルはテキストファイルの一種ですから、ホスト→W i n方向の変換時にはE O Fコード(1 A H)をつけない、W i n→ホスト方向の変換時にはE O Fコード検出でも物理E O F検出でもE O Fにする、というのが省略時の動作です。

E O Fコードの扱い

機 能	説 明
E O Fコード付加	ファイル末尾にE O Fコードを付加する・しない(ホスト→W i n)
E O Fコード検出	E O Fコード検出でE O Fにする(W i n→ホスト)
E O Fコード透過	E O Fコードを一般の文字扱いする(W i n→ホスト)

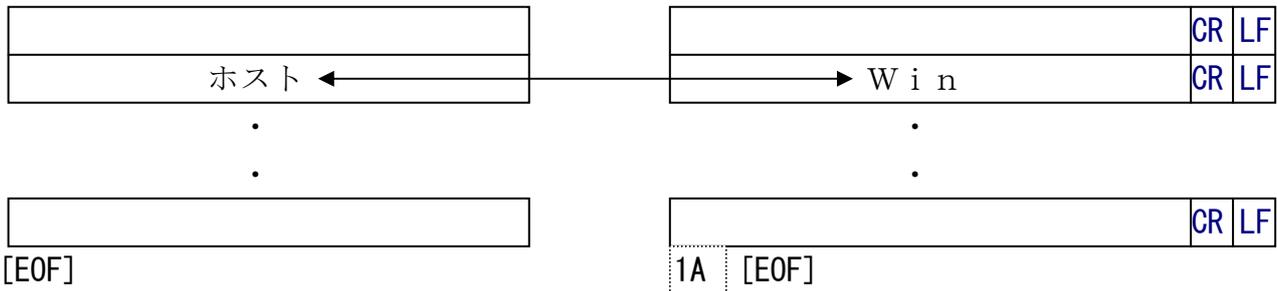
●BOMコードの扱い

Windows側のコードとしてUnicodeを選択した場合、BOM(バイトオーダーマーク)の扱いについての指定が可能です。ホスト→Win方向の変換時でコードがUCS-2の時にはBOMコードをつける、UTF-8の時にはBOMコードをつけない、Win→ホスト方向の変換時でコードがUCS-2の時には入力ファイルにBOMコードあり、UTF-8の時には入力ファイルにBOMコードなしとして変換する、というのが省略時の動作です。

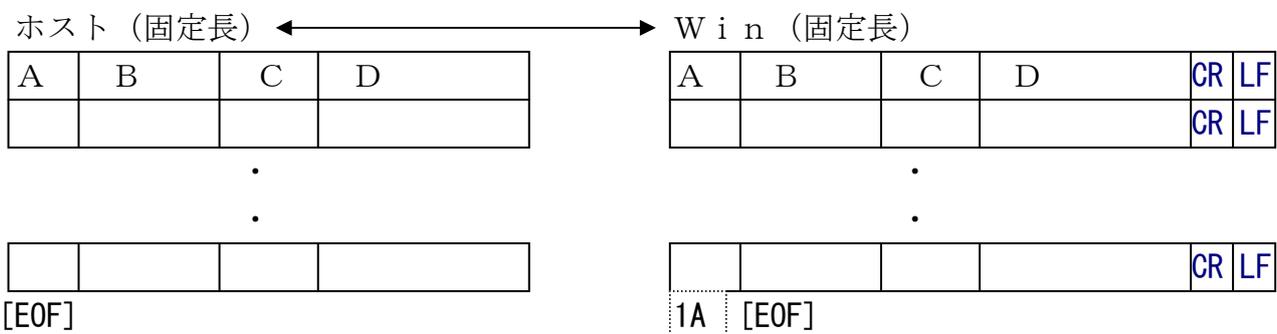
BOMコードの扱い

機能	説明
BOM自動	UCS-2の時BOMあり、UTF-8の時BOMなし
BOMあり	BOMを付加する(ホスト→Win)
	BOMを検査する(Win→ホスト)
BOMなし	BOMを付加しない(ホスト→Win)
	BOMを検査しない(Win→ホスト)

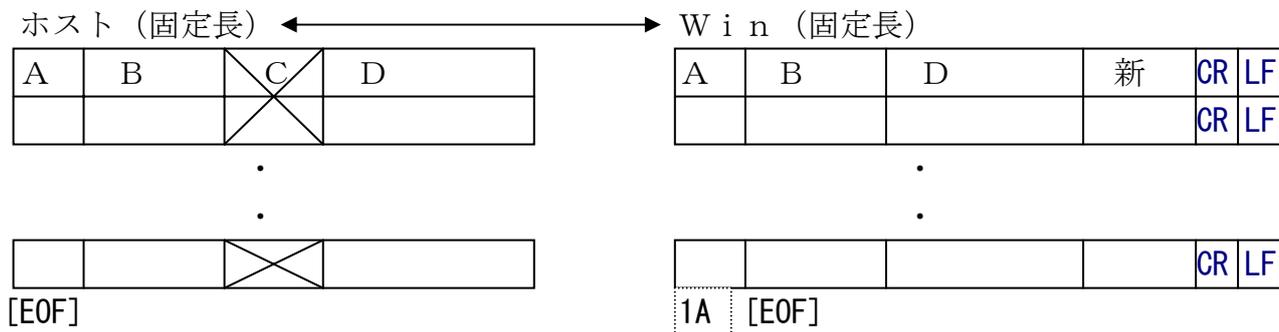
例1) ふつう、固定長⇔固定長の変換になる



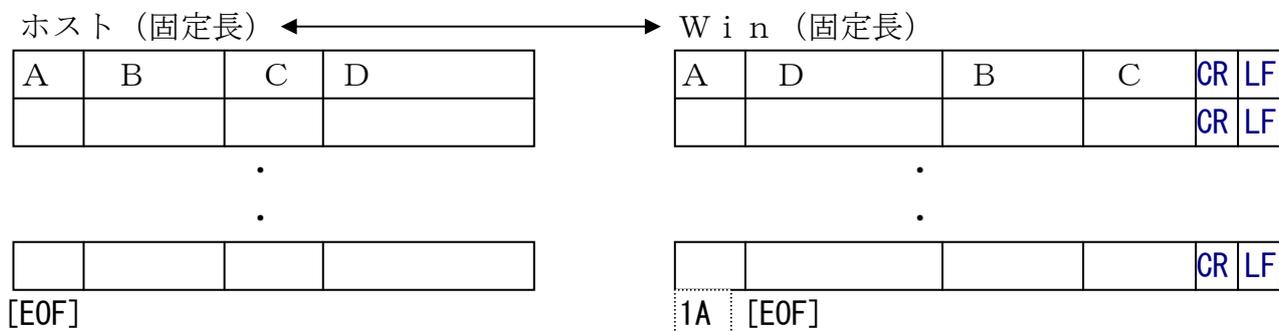
例2) 項目別変換ができる



例6) 不要な項目をスキップしたり、逆に空項目を作ったりできる



例7) 項目の組み替えができる



■ホストがUnix、Windowsの場合

ホストがUnix、Windowsの場合、データファイル変換1/プリント形式は、ホストのCOBOLデータ等と、市販ソフトの入出力形式の1つであるプリント形式ファイルの変換に使用します。

プリント形式ファイルとは、各レコードが改行コードで区切られているテキストファイルのことです。各項目の桁位置、幅が一定していなければいけません。

以下に、データファイル変換1/プリント形式で可能なデータ変換機能、レコード編集機能、EOFコードの扱い、BOMコードの扱いを示します。

●データ変換機能

データファイル変換1/プリント形式のデータ変換機能を下表に示します。

データ変換機能		変換方法
ホスト側データ	Win側データ	
JIS8/ASCII	JIS8/ASCII	ANK変換
Unicode(UCS-2)	Unicode(UCS-2)	〃
Unicode(UTF-8)	Unicode(UTF-8)	〃
漢字(EUC、シフトJIS)	シフトJIS漢字のみ	漢字項目変換
〃	Unicode(UCS-2)	〃
〃	Unicode(UTF-8)	〃
〃	シフトJIS漢字、JIS8混在	〃
JIS8・漢字混在	JIS8、シフトJIS漢字混在 Unicode(UCS-2) Unicode(UTF-8)	ANK・漢字混在変換
文字形式数値	文字形式、Windows COBOL *1	ニューメリック変換 *3
符号なしゾーン形式	〃	ゾーン変換 *4
符号つきゾーン形式	〃	〃
符号なしパック形式	〃	パック変換 *5
符号つきパック形式	〃	〃
符号なしBCD形式	〃	〃
2進形式数値 *2	〃	2進数値変換*6
日付データ	日付データ	日付項目変換
特殊データ	特殊データ	特殊データ変換

*1) Windows COBOL の符号なしゾーン形式、符号つきゾーン形式

*2) COBOL の COMP-4 や COMP-5 などの形式

*3) Numeric、DispZone、ZoneDisp、AlphaZone、ZoneAlpha 変換

*4) ZoneDisp、DispZone、ZoneZone、ZoneAlpha、AlphaZone 変換

*5) PackDisp、DispPack、PackZone、ZonePack、PackAlpha、AlphaPack 変換

*6) BinDisp、DispBin、BinZone、ZoneBin、BinAlpha、AlphaBin 変換

●レコード編集機能

プリント形式ファイルは、基本的には固定長ファイルです。しかし、項目単位の編集・加工ができないと使いものになりません。下表に、データファイル変換1 / プリント形式のレコード編集機能を示します。

レコード編集機能

機 能	説 明
タブ拡張	タブを連続スペースに置き換える
改行コード正規化	レコード末尾の改行コードを正規化する
空行無視	改行・改頁コードのみの行を無視する・しない
項目別変換	項目別に分けて変換できる
項目削除	不要な項目を削除できる
項目長変更	項目長を変更できる
項目組み替え	項目の順番の組み替えができる
空項目挿入	空項目を作ることができる
定数挿入	任意の値を任意の形式で挿入することができる

●EOFコードの扱い

プリント形式ファイルはテキストファイルの一種ですから、ホスト→Win方向の変換時にはEOFコード(1AH)をつけない、Win→ホスト方向の変換時にはEOFコード検出でも物理EOF検出でもEOFにする、というのが省略時の動作です。

EOFコードの扱い

機 能	説 明
EOFコード付加	ファイル末尾にEOFコードを付加する・しない(ホスト→Win*)
EOFコード検出	EOFコード検出でEOFにする(Win→ホスト*)
EOFコード透過	EOFコードを一般の文字扱いする(Win→ホスト*)

*) ホストがWindowsの場合は、ホスト⇔Win

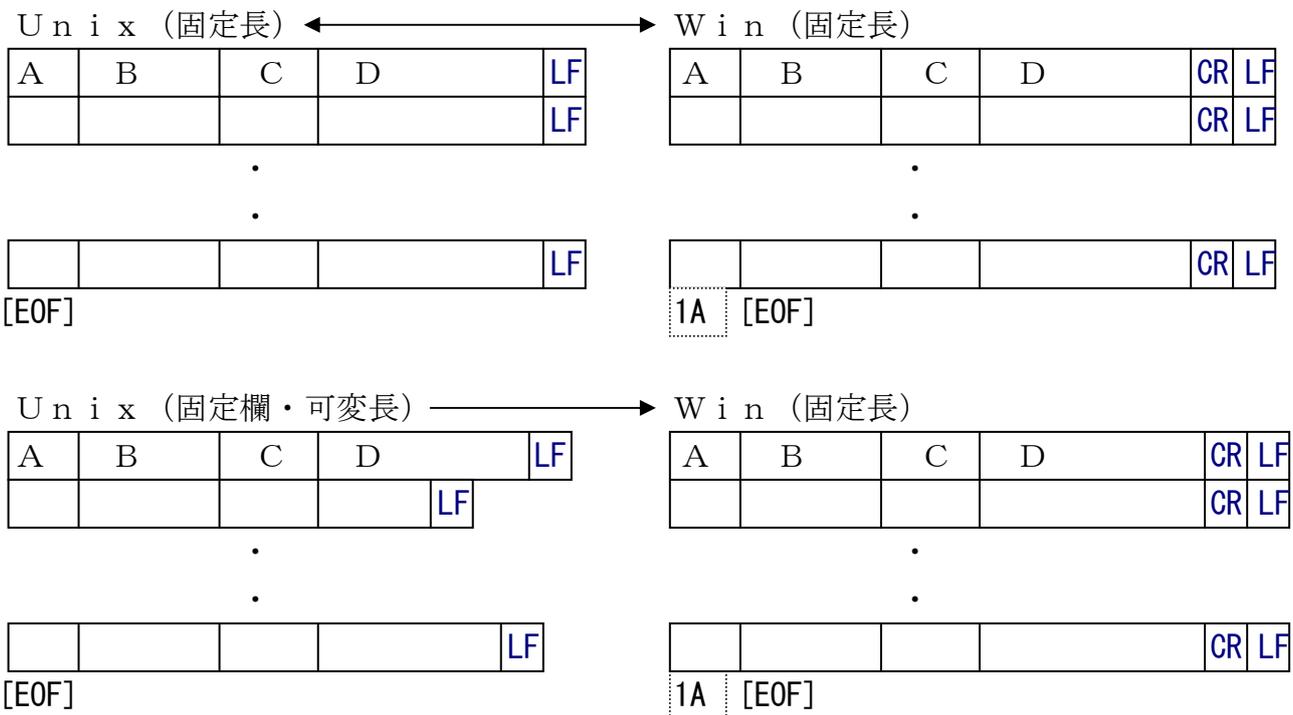
●BOMコードの扱い

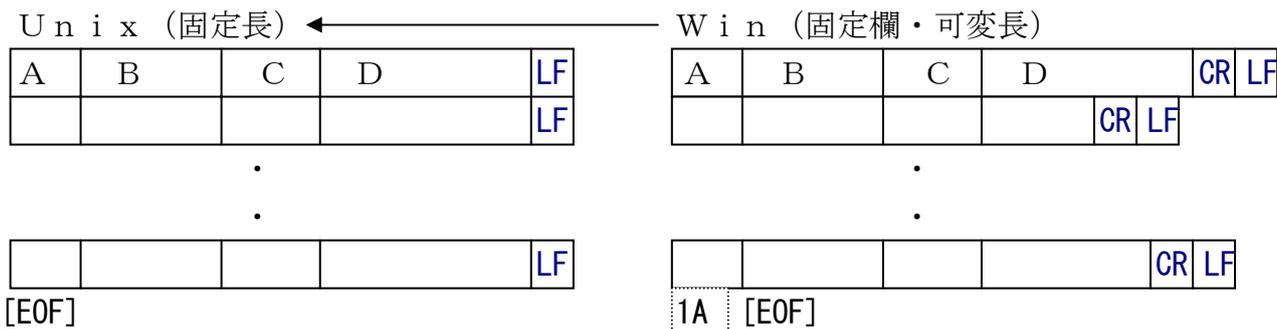
Windows側のコードとしてUnicodeを選択した場合、BOM(バイトオーダーマーク)の扱いについての指定が可能です。ホスト→Win方向の変換時でコードがUCS-2の時にはBOMコードをつける、UTF-8の時にはBOMコードをつけない、Win→ホスト方向の変換時でコードがUCS-2の時には入力ファイルにBOMコードあり、UTF-8の時には入力ファイルにBOMコードなしとして変換する、というのが省略時の動作です。

BOMコードの扱い

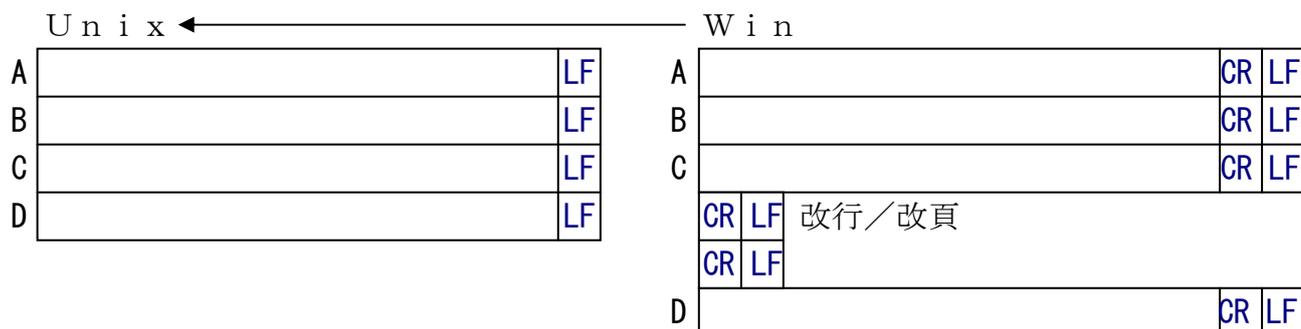
機能	説明
BOM自動	UCS-2の時BOMあり、UTF-8の時BOMなし
BOMあり	BOMを付加する(ホスト→Win)
	BOMを検査する(Win→ホスト)
BOMなし	BOMを付加しない(ホスト→Win)
	BOMを検査しない(Win→ホスト)

例1) 入力が可変長/固定長、出力が固定長の変換になり、項目別変換ができる



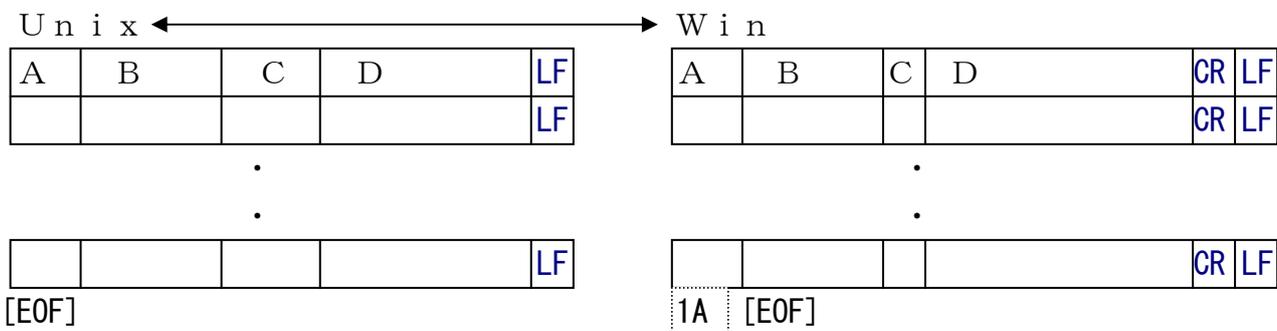


例2) 改行コード、または改頁コードのみの行は無視できる

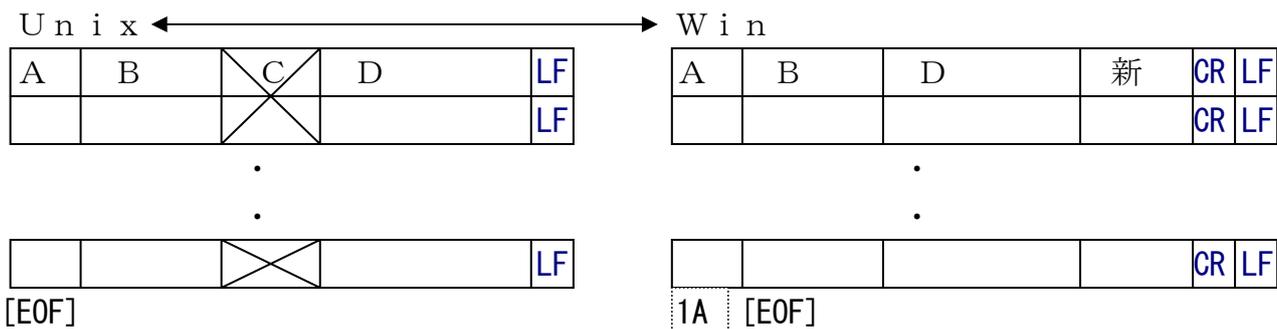


* 改頁コード=CR/FF (0D0CH) またはFF (0CH)

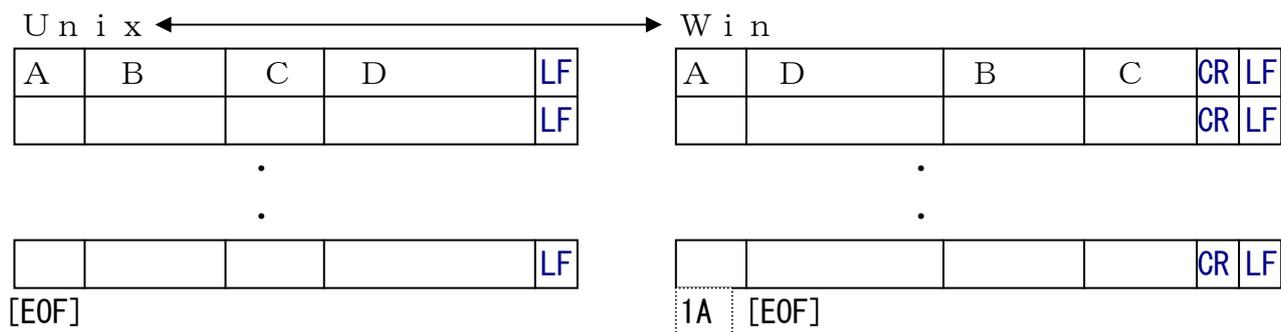
例3) 項目長の変更(縮小、拡張)ができる



例4) 不要な項目をスキップしたり、逆に空項目を作ったりできる



例5) 項目の組み替えができる



2.9 データファイル変換2 / デリミタ形式の仕様

データファイル変換2 / デリミタ形式は、おもにホストのCOBOLデータと、市販ソフトの入出力形式の1つである、デリミタ形式ファイルの変換に使用します。

F*TRANは、以下のようなバリエーションのデリミタ形式ファイルをサポートしています。

区切り文字

- | | | |
|--------------------|---|------------------|
| ・ コンマ区切り形 (CSV 形式) | : | 項目をコンマで区切る |
| ・ タブ区切り形式 (TSV 形式) | : | 項目を TAB(09) で区切る |
| ・ スペース区切り形式 | : | 項目を半角スペースで区切る |

引用符

- ・ なし
- ・ 単一引用符 (') を使う
- ・ 二重引用符 (") を使う

項目取り出し方式 (例外的データへの対処)

- ・ 旧 F*TRAN 風
- ・ 新 F*TRAN 風
- ・ MS-Excel 風
- ・ MS-Access 風

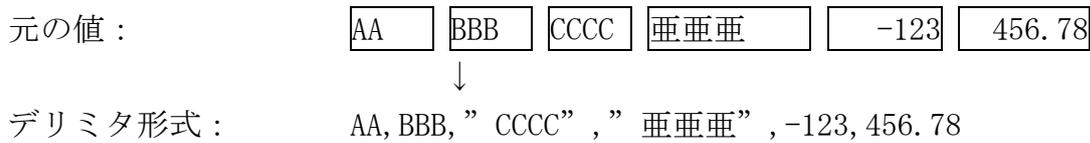
現在のところ、F*TRAN+は自由文字区切りには対応していません。また、項目内改行にも対応していません。

◆参考 ---- CSV形式（デリミタ形式）について

CSV (Comma Separated Value) 形式に代表される、F*TRANでいうデリミタ形式は、2次元の表のようなデータをテキスト表現するのに適した形式です。非常に多くのソフトが、この形式での入出力をサポートしています。デリミタ形式の規約としてきっちり定まったものはないのですが、ほぼ以下のルールに従ってデータを表現します。

- ◎ すべてのデータを文字表現する
- ◎ 可変長項目・可変長レコードである
- ◎ (原則として) 1行1レコードとする
- ◎ 項目同士は、コンマ(,)、タブ、スペース、その他特殊文字などの区切り文字で区切る
- ◎ 文字列項目は、両端を引用符(“、’)でくくることが多い
 - ・データとして区切り文字を含むときは必須
 - ・データとして同じ種類の引用符を含むときは必須
 - ・その際、データとしての引用符は” ” や’ ’ のように、2連化する

典型的な例を示します。



この例のように、区切り文字としてコンマを使う、CSV形式が最も一般的です。また、引用符としては2重引用符(“)を使うのが一般的です。

■ホストが汎用機・オフコンの場合

ホストが汎用機・オフコンの場合、データファイル変換2/デリミタ形式は、おもにホストのCOBOLデータと、市販ソフトの入出力形式の1つであるデリミタ形式ファイルの変換に使用します。

デリミタ形式ファイルとは、各レコードが改行コードで区切られ、さらに各項目がデリミタ(区切り文字)で区切られているファイルのことです。コンマ区切り(CSV)形式、タブ区切り形式、スペース区切り形式の3つのタイプに対応しています。コンマ区切り形式が代表的な形式です。

以下に、データファイル変換2/デリミタ形式で可能なデータ変換機能、レコード編集機能、EOFコードの扱い、BOMコードの扱いを示します。

●データ変換機能

データファイル変換2/デリミタ形式のデータ変換機能を、下表に示します。

データ変換機能		変換方法
ホスト側データ	Win側データ	
JIS8/ASCII	JIS8/ASCII	ANK変換
EBCDIC (カタカナ)	Unicode (UCS-2)	〃
EBCDIC (英小文字)	Unicode (UTF-8)	〃
漢字 (KI/KOなし)	シフトJIS漢字のみ	漢字項目変換
〃	Unicode (UCS-2)	〃
〃	Unicode (UTF-8)	〃
〃	シフトJIS漢字、JIS8混在	〃
ANK・漢字混在	JIS8、シフトJIS漢字混在	ANK・漢字
KI/KOつき	Unicode (UCS-2)	混在変換
	Unicode (UTF-8)	
文字形式数値	文字形式、Windows COBOL *1	ニューメリック変換 *3
符号なしゾーン形式	〃	ゾーン変換 *4
符号つきゾーン形式	〃	〃
符号なしパック形式	〃	パック変換 *5
符号つきパック形式	〃	〃
符号なしBCD形式	〃	〃
2進形式数値 *2	文字形式、Windows COBOL *1	2進数値変換*6
日付データ	日付データ	日付項目変換
特殊データ	特殊データ	特殊データ変換

- *1) Windows COBOL の符号なしゾーン形式、符号つきゾーン形式
- *2) COBOL の COMP-4 や COMP-5 などの形式
- *3) Numeric、DispZone、ZoneDisp、AlphaZone、ZoneAlpha 変換
- *4) ZoneDisp、DispZone、ZoneZone、ZoneAlpha、AlphaZone 変換
- *5) PackDisp、DispPack、PackZone、ZonePack、PackAlpha、AlphaPack 変換
- *6) BinDisp、DispBin、BinZone、ZoneBin、BinAlpha、AlphaBin 変換

●レコード編集機能

ホストの標準的なデータ形式は、COBOLのファイル形式です。パソコンの標準的なデータ形式は、プリント形式またはこのデリミタ形式です。両者にはとても大きな隔たりがあり、十分なレコード編集機能が要求されます。F*TRAN+の持つレコード編集機能を下表に示します。

レコード編集機能

機 能	説 明
タブ拡張	タブを連続スペースに置き換える(W i n→ホスト)
改行コード付加	無条件にレコード末尾に改行コードを付加する(ホスト→W i n)
改行コード削除	無条件にレコード末尾の改行コードを削除する(W i n→ホスト)
空行無視	W i n側の、改行・改頁コードのみの行を無視する・しない
項目別変換	項目別に分けて変換できる
可変長項目対応	可変長の項目でも変換できる
デリミタ挿入	項目を区切るデリミタを挿入できる(ホスト→W i n)
デリミタ検出	項目を区切るデリミタを検出できる(W i n→ホスト)
引用符くくり	項目を引用符(“、”)でくることができる(ホスト→W i n)
引用符はずし	引用符でくられた項目の引用符をはずす(W i n→ホスト)
不要空白削除	デリミタの前後や引用符の前の不要な空白を削除できる (ホスト→W i n)。削除しない指定も可能
項目削除	不要な項目を削除できる
項目長変更	項目長を変更できる
項目組み替え	項目の順番の組み替えができる
空項目挿入	空項目を作ることができる
定数挿入	任意の値を任意の形式で挿入することができる
K I / K O 挿入	漢字項目の前後にK I / K Oを挿入できる(W i n→ホスト)
改行コード挿入	レコードの途中にも、改行コードを挿入できる(ホスト→W i n)

● E O Fコードの扱い

デリミタ形式ファイルはテキストファイルの一種ですから、ホスト→W i n方向の変換時には E O Fコード (1 A H) をつけない、W i n→ホスト方向の変換時には E O Fコード検出でも物理 E O F検出でも E O Fにする、というのが省略時の動作です。

E O Fコードの扱い

機 能	説 明
E O Fコード付加	ファイル末尾に E O Fコードを付加する・しない(ホスト→W i n)
E O Fコード検出	E O Fコード検出で E O Fにする(W i n→ホスト)
E O Fコード透過	E O Fコードを一般の文字扱いする(W i n→ホスト)

● B O Mコードの扱い

W i n d o w s側のコードとして U n i c o d eを選択した場合、B O M (バイトオーダーマーク)の扱いについての指定が可能です。ホスト→W i n方向の変換時でコードが U C S - 2の時には B O Mコードをつける、U T F - 8の時には B O Mコードをつけない、W i n→ホスト方向の変換時でコードが U C S - 2の時には入力ファイルに B O Mコードあり、U T F - 8の時には入力ファイルに B O Mコードなしとして変換する、というのが省略時の動作です。

B O Mコードの扱い

機 能	説 明
B O M自動	U C S - 2の時 B O Mあり、U T F - 8の時 B O Mなし
B O Mあり	B O Mを付加する (ホスト→W i n) B O Mを検査する(W i n→ホスト)
B O Mなし	B O Mを付加しない (ホスト→W i n) B O Mを検査しない(W i n→ホスト)

例1) コンマ区切り (CSV) 形式の変換ができる

5	木星	JUPITER	ジュピター	317832	16
6	土星	SATURN	サターン	095160	17
7	天王星	URANUS	ウラヌ	014540	15
8	海王星	NEPTUNE	ネプチューン	017150	08

ホスト
 ↑↓
 Win " 5" , " 木星" , " JUPITER" , " ジュピター" , 317.832 , 16 ↓
 " 6" , " 土星" , " SATURN" , " サターン" , 95.160 , 17 ↓
 " 7" , " 天王星" , " URANUS" , " ウラヌ" , 14.540 , 15 ↓
 " 8" , " 海王星" , " NEPTUNE" , " ネプチューン" , 17.150 , 8 ↓

例2) タブ区切り形式の変換ができる

5	木星	JUPITER	ジュピター	317832	16
6	土星	SATURN	サターン	095160	17
7	天王星	URANUS	ウラヌ	014540	15
8	海王星	NEPTUNE	ネプチューン	017150	08

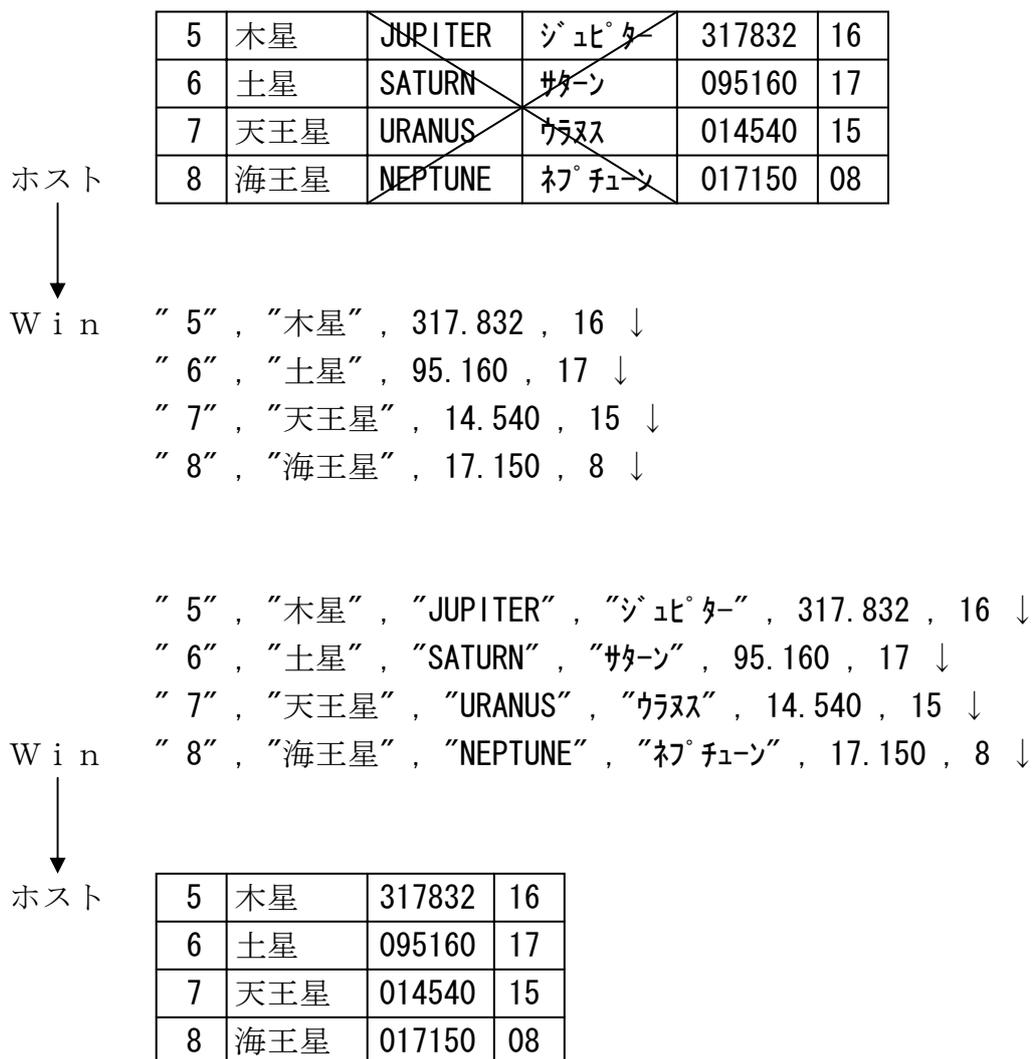
ホスト
 ↑↓
 Win 5.....>木星...>JUPITER.>ジュピター.>317.832>16 ↓
 6.....>土星...>SATURN..>サターン....>95.160.>17 ↓
 7.....>天王星.>URANUS..>ウラヌ....>14.540.>15 ↓
 8.....>海王星.>NEPTUNE.>ネプチューン.>17.150.>8 ↓
 (便宜的に...>でタブを表しています)

例3) スペース区切り形式の変換ができる

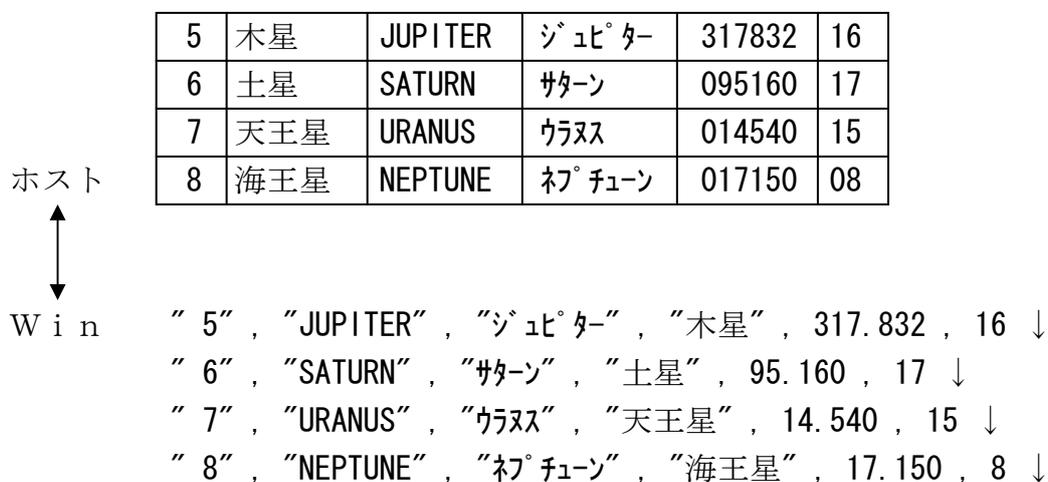
5	木星	JUPITER	ジュピター	317832	16
6	土星	SATURN	サターン	095160	17
7	天王星	URANUS	ウラヌ	014540	15
8	海王星	NEPTUNE	ネプチューン	017150	08

ホスト
 ↑↓
 Win 5 木星 JUPITER ジュピター 317.832 16 ↓
 6 土星 SATURN サターン 95.160 17 ↓
 7 天王星 URANUS ウラヌ 14.540 15 ↓
 8 海王星 NEPTUNE ネプチューン 17.150 8 ↓

例4) 不要項目をスキップできる



例5) 項目の組み替えができる



■ホストがUnix、Windowsの場合

ホストがUnix、Windowsの場合、データファイル変換2/デリミタ形式は、ホストのCOBOLデータ等と、市販ソフトの入出力形式の1つであるデリミタ形式ファイルの変換に使用します。

デリミタ形式ファイルとは、各レコードが改行コードで区切られ、さらに各項目がデリミタ(区切り文字)で区切られているファイルのことです。コンマ区切り(CSV)形式、タブ区切り形式、スペース区切り形式の3つのタイプに対応しています。コンマ区切り形式が代表的な形式です。

以下に、データファイル変換2/デリミタ形式で可能なデータ変換機能、レコード編集機能、EOFコードの扱い、BOMコードの扱いを示します。

●データ変換機能

データファイル変換2/デリミタ形式のデータ変換機能を、下表に示します。

データ変換機能	ホスト側データ ↔ Win側データ	変換方法
JIS8/ASCII	↔ JIS8/ASCII	ANK変換
Unicode(UCS-2)	↔ Unicode(UCS-2)	〃
Unicode(UTF-8)	↔ Unicode(UTF-8)	〃
漢字(EUC、シフトJIS)	↔ シフトJIS漢字のみ	漢字項目変換
〃	↔ Unicode(UCS-2)	〃
〃	↔ Unicode(UTF-8)	〃
〃	← シフトJIS漢字、JIS8混在	〃
JIS8・漢字混在	↔ { JIS8、シフトJIS漢字混在 Unicode(UCS-2) Unicode(UTF-8)	ANK・漢字混在変換
文字形式数値	↔ 文字形式、Windows COBOL *1	ニューメリック変換 *3
符号なしゾーン形式	↔ 〃	ゾーン変換 *4
符号つきゾーン形式	↔ 〃	〃
符号なしパック形式	↔ 〃	パック変換 *5
符号つきパック形式	↔ 〃	〃
符号なしBCD形式	↔ 〃	〃
2進形式数値 *2	↔ 〃	2進数値変換*6
日付データ	↔ 日付データ	日付項目変換
特殊データ	↔ 特殊データ	特殊データ変換

*1) Windows COBOL の符号なしゾーン形式、符号つきゾーン形式

*2) COBOL の COMP-4 や COMP-5 などの形式

*3) Numeric、DispZone、ZoneDisp、AlphaZone、ZoneAlpha 変換

- *4) ZoneDisp、DispZone、ZoneZone、ZoneAlpha、AlphaZone 変換
- *5) PackDisp、DispPack、PackZone、ZonePack、PackAlpha、AlphaPack 変換
- *6) BinDisp、DispBin、BinZone、ZoneBin、BinAlpha、AlphaBin 変換

●レコード編集機能

Unix、Windowsの標準的なデータ形式は、プリント形式またはこのデリミタ形式です。F*TRAN+の持つレコード編集機能を下表に示します。

レコード編集機能

機 能	説 明
タブ拡張	タブを連続スペースに置き換える
改行コード正規化	レコード末尾の改行コードを正規化する
空行無視	改行・改頁コードのみの行を無視する・しない
項目別変換	項目別に分けて変換できる
可変長項目対応	可変長の項目でも変換できる
デリミタ挿入	項目を区切るデリミタを挿入できる
デリミタ検出	項目を区切るデリミタを検出できる
引用符くくり	項目を引用符(“、”)でくることができる
引用符はずし	引用符でくられた項目の引用符をはずす
不要空白削除	デリミタの前後や引用符の前の不要な空白を削除できる 削除しない指定も可能
項目削除	不要な項目を削除できる
項目長変更	項目長を変更できる
項目組み替え	項目の順番の組み替えができる
空項目挿入	空項目を作ることができる
定数挿入	任意の値を任意の形式で挿入することができる
改行コード挿入	レコードの途中にも、改行コードを挿入できる

●EOFコードの扱い

デリミタ形式ファイルはテキストファイルの一種ですから、ホスト→Win方向の変換時にはEOFコード(1AH)をつけない、Win→ホスト方向の変換時にはEOFコード検出でも物理EOF検出でもEOFにする、というのが省略時の動作です。

EOFコードの扱い

機 能	説 明
EOFコード付加	ファイル末尾にEOFコードを付加する・しない(ホスト→Win*)
EOFコード検出	EOFコード検出でEOFにする(Win→ホスト*)
EOFコード透過	EOFコードを一般の文字扱いする(Win→ホスト*)

*) ホストがWindowsの場合は、ホスト⇔Win

●BOMコードの扱い

Windows側のコードとしてUnicodeを選択した場合、BOM(バイトオーダーマーク)の扱いについての指定が可能です。ホスト→Win方向の変換時でコードがUCS-2の時にはBOMコードをつける、UTF-8の時にはBOMコードをつけない、Win→ホスト方向の変換時でコードがUCS-2の時には入力ファイルにBOMコードあり、UTF-8の時には入力ファイルにBOMコードなしとして変換する、というのが省略時の動作です。

BOMコードの扱い

機能	説明
BOM自動	UCS-2の時BOMあり、UTF-8の時BOMなし
BOMあり	BOMを付加する(ホスト→Win) BOMを検査する(Win→ホスト)
BOMなし	BOMを付加しない(ホスト→Win) BOMを検査しない(Win→ホスト)

例1) コンマ区切り(CSV)形式の変換ができる

5	木星	JUPITER	ジュピター	317832	16	LF
6	土星	SATURN	サターン	095160	17	LF
7	天王星	URANUS	ウラヌス	014540	15	LF
8	海王星	NEPTUNE	ネプチューン	017150	08	LF

Unix

↑↓

Win

" 5", "木星", "JUPITER", "ジュピター", 317.832, 16 ↓
 " 6", "土星", "SATURN", "サターン", 95.160, 17 ↓
 " 7", "天王星", "URANUS", "ウラヌス", 14.540, 15 ↓
 " 8", "海王星", "NEPTUNE", "ネプチューン", 17.150, 8 ↓

例2) タブ区切り形式の変換ができる

5	木星	JUPITER	ジュピター	317832	16	LF
6	土星	SATURN	サターン	095160	17	LF
7	天王星	URANUS	ウラヌ	014540	15	LF
8	海王星	NEPTUNE	ネプチューン	017150	08	LF

U n i x

↕

W i n

5.....>木星...>JUPITER.>ジュピター.>317.832>16 ↓
 6.....>土星...>SATURN..>サターン....>95.160.>17 ↓
 7.....>天王星.>URANUS..>ウラヌ....>14.540.>15 ↓
 8.....>海王星.>NEPTUNE.>ネプチューン.>17.150.>8 ↓

(便宜的に...>でタブを表しています)

例3) スペース区切り形式の変換ができる

5	木星	JUPITER	ジュピター	317832	16	LF
6	土星	SATURN	サターン	095160	17	LF
7	天王星	URANUS	ウラヌ	014540	15	LF
8	海王星	NEPTUNE	ネプチューン	017150	08	LF

U n i x

↕

W i n

5 木星 JUPITER ジュピター 317.832 16 ↓
 6 土星 SATURN サターン 95.160 17 ↓
 7 天王星 URANUS ウラヌ 14.540 15 ↓
 8 海王星 NEPTUNE ネプチューン 17.150 8 ↓

例4) 不要項目をスキップできる

5	木星	JUPITER	ジュピター	317832	16	LF
6	土星	SATURN	サターン	095160	17	LF
7	天王星	URANUS	ウラヌ	014540	15	LF
8	海王星	NEPTUNE	ネプチューン	017150	08	LF

U n i x

↓

W i n

" 5", "木星", 317.832, 16 ↓
 " 6", "土星", 95.160, 17 ↓
 " 7", "天王星", 14.540, 15 ↓
 " 8", "海王星", 17.150, 8 ↓

Win

↓

Unix

5	木星	317832	16	LF
6	土星	095160	17	LF
7	天王星	014540	15	LF
8	海王星	017150	08	LF

例5) 項目の組み替えができる

Win

↑

Win

" 5"	"木星"	"JUPITER"	"ジュピター"	317.832	16 ↓
" 6"	"土星"	"SATURN"	"サターン"	95.160	17 ↓
" 7"	"天王星"	"URANUS"	"ウラヌス"	14.540	15 ↓
" 8"	"海王星"	"NEPTUNE"	"ネプチューン"	17.150	8 ↓

" 5"	"JUPITER"	"ジュピター"	"木星"	317.832	16 ↓
" 6"	"SATURN"	"サターン"	"土星"	95.160	17 ↓
" 7"	"URANUS"	"ウラヌス"	"天王星"	14.540	15 ↓
" 8"	"NEPTUNE"	"ネプチューン"	"海王星"	17.150	8 ↓

2. 10 ランダムファイル変換の仕様

■ホストが汎用機・オフコンの場合

ランダムファイル変換では、アプリケーションで使用する、おもに固定長・固定欄のデータファイルの変換を行います。BASICのランダムファイルやCOBOLの順ファイルなどがおもな対象です。バイナリ変換ができるのも大きな特長です。

以下に、ランダムファイル変換で可能なデータ変換機能、レコード編集機能、EOFコードの扱い、BOMコードの扱いを示します。

●データ変換機能

ランダムファイル変換のデータ変換機能を下表に示します。

データ変換機能		変換方法
ホスト側データ	↔ Win側データ	
バイナリ	↔ バイナリ	バイナリ変換
JIS8/ASCII	↔ { JIS8/ASCII Unicode (UCS-2) Unicode (UTF-8)	ANK変換
EBCDIC (カタカナ)		〃
EBCDIC (英小文字)		〃
漢字 (KI/KOなし)	↔ シフトJIS漢字のみ	漢字項目変換
〃	↔ Unicode (UCS-2)	〃
〃	↔ Unicode (UTF-8)	〃
〃	← シフトJIS漢字、JIS8混在	〃
ANK・漢字混在	↔ { JIS8、シフトJIS漢字混在 Unicode (UCS-2) Unicode (UTF-8)	ANK・漢字混在変換
KI/KOつき		
文字形式数値	↔ 文字、WinCOBOL *1、2進 *2	ニューメリック変換 *4
符号なしゾーン形式	↔ 〃	ゾーン変換 *5
符号つきゾーン形式	↔ 〃	〃
符号なしパック形式	↔ 〃	パック変換 *6
符号つきパック形式	↔ 〃	〃
符号なしBCD形式	↔ 〃	〃
2進形式数値 *3	↔ 〃	2進数値変換*7
日付データ	↔ 日付データ	日付項目変換
特殊データ	↔ 特殊データ	特殊データ変換

*1) Windows COBOL の符号なしゾーン/パック形式、符号つきゾーン/パック形式

*2) Windows COBOL の COMP-4 や COMP-5、Visual Basic や C/C++ の整数型などの形式

*3) COBOL の COMP-4 や COMP-5 などの形式

*4) Numeric、DispZone、ZoneDisp、DispPack、PackDisp、DispBin、BinDisp 変換
AlphaZone、ZoneAlpha、AlphaPack、PackAlpha、AlphaBin、BinAlpha 変換

- *5) ZoneDisp、DispZone、ZoneZone、ZonePack、PackZone、ZoneBin、BinZone 変換
ZoneAlpha、AlphaZone 変換
- *6) PackDisp、DispPack、PackZone、ZonePack、PackPack、PackBin、BinPack 変換
PackAlpha、AlphaPack 変換
- *7) BinDisp、DispBin、BinZone、ZoneBin、BinPack、PackBin、BinBin 変換
BinAlpha、AlphaBin 変換

◆注意 ----- 浮動小数点には未対応

ホスト側もWin側も、浮動小数点型には対応していません。

●レコード編集機能

ランダムファイル変換では、Win側のレコード長を明示的に指定します（指定を省略するとホストレコード長=Winレコード長になります）。固定長という点では、ホストのデータ形式と似ています。

しかし、データ内容に応じて項目長変更などのレコード編集機能が必要になります。明示的に指定しない限り改行コードをつけないことなどは、大きな特長です。

下表にランダムファイル変換のレコード編集機能を示します。

レコード編集機能

機 能	説 明
Winレコード長指定	Win側のレコード長を指定できる
項目別変換	項目別に分けて変換できる
項目削除	不要な項目を削除できる
項目長変更	項目長を変更できる
項目組み替え	項目の順番の組み替えができる
空項目挿入	空項目を作ることができる
定数挿入	任意の値を任意の形式で挿入することができる
KI/KO挿入	漢字項目の前後にKI/KOを挿入できる(Win→ホスト)
改行コード挿入	レコードの途中・末尾に、改行コードを挿入できる (ホスト→Win)

●EOFコードの扱い

ランダムファイル変換は、基本的にデリミタや改行コードを使わず、あってもただの一般文字扱いにします。EOFコードについても同様に一般文字扱いし、物理EOFでEOFにします。これは、バイナリ変換ができることを前提にしているためです。

EOFコードの扱い

機 能	説 明
EOFコード透過	EOFコードを一般の文字扱いする(Win→ホスト)

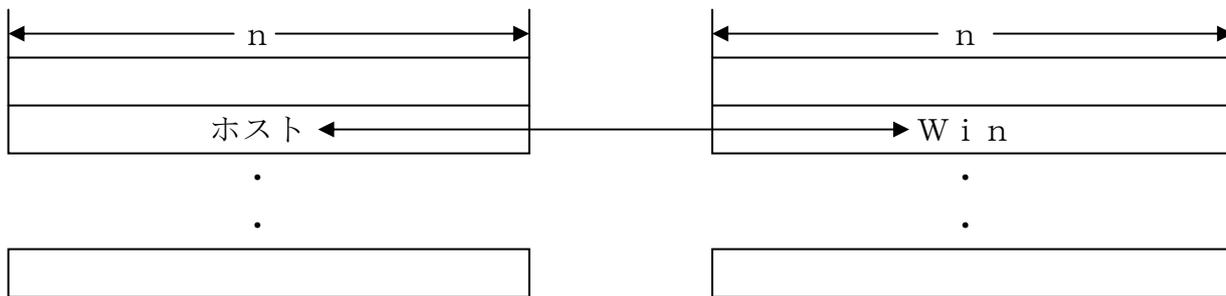
●BOMコードの扱い

ランダムファイル変換は、基本的にBOMコードを使わず、あってもただの一般文字扱いにします。これは、バイナリ変換ができることを前提にしているためです。

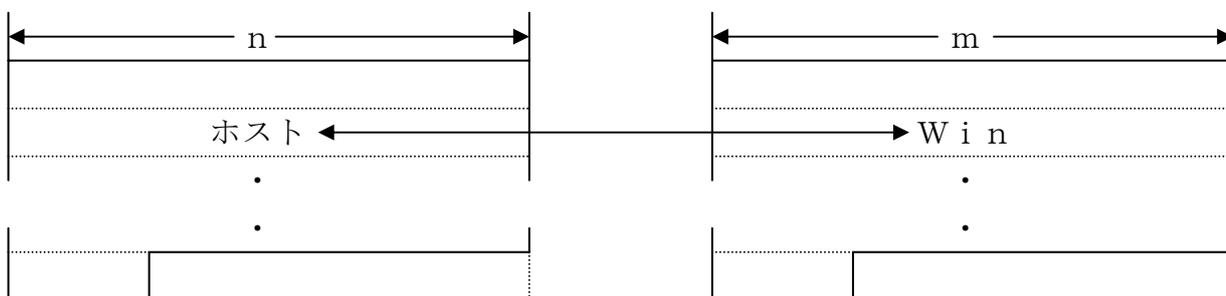
BOMコードの扱い

機 能	説 明
BOMコード透過	BOMコードを一般の文字扱いする

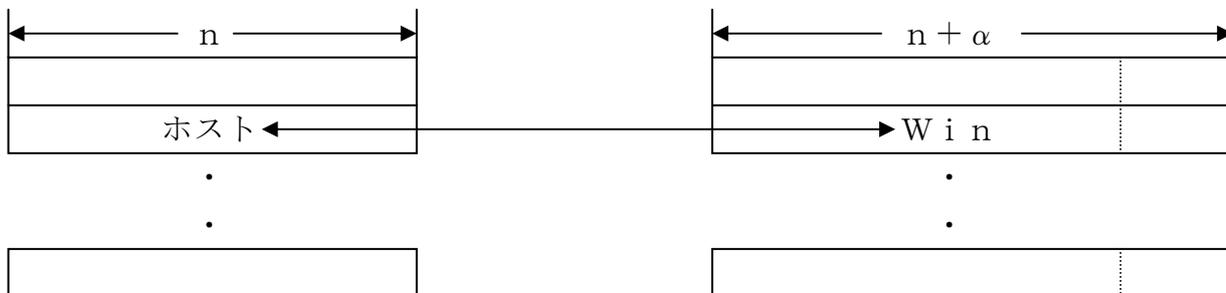
例1) ホスト側とWin側のレコード長を同じにするのがふつう



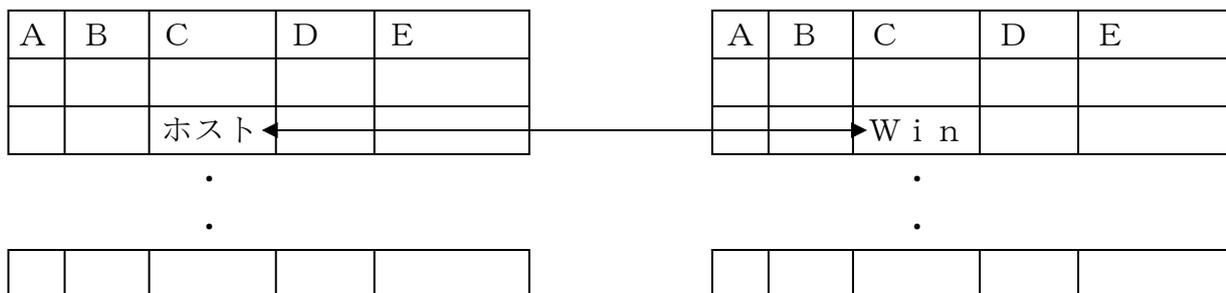
例2) バイナリ変換もできる。その際にレコード長を変えることもできる



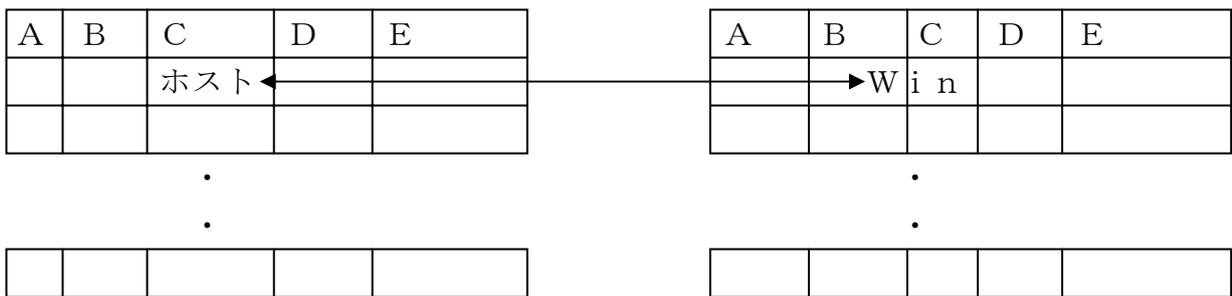
例3) レコード長の変更（縮小、拡大）ができる



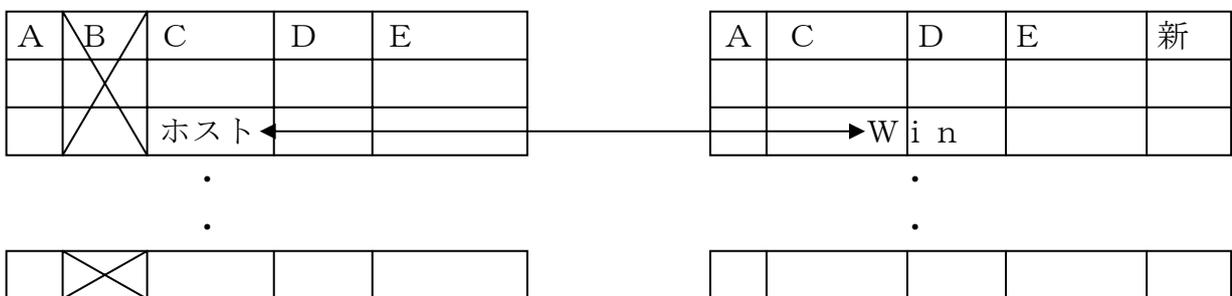
例4) 項目別の変換ができる



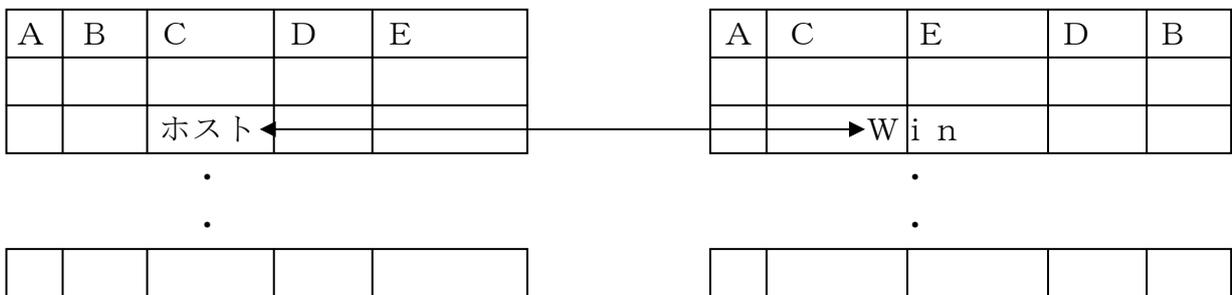
例5) 項目長の変更 (縮小、拡大) ができる



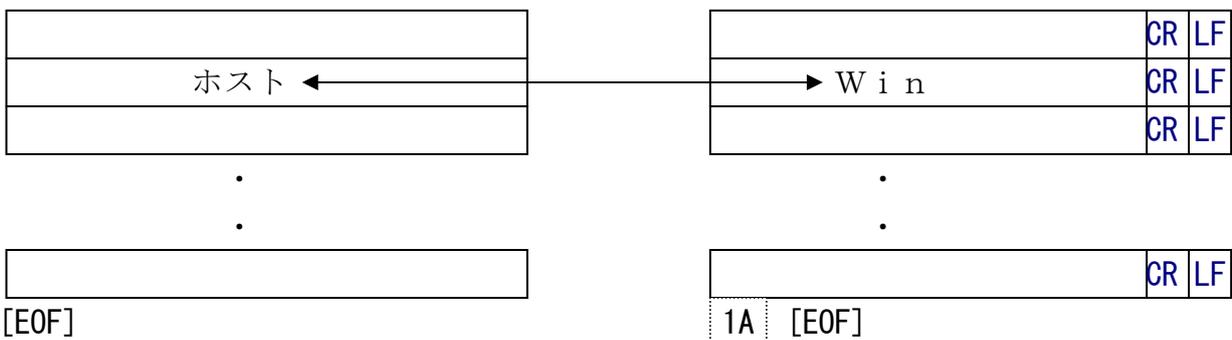
例6) 不要な項目をスキップしたり、逆に空項目を作ったりできる



例7) 項目の組み替えができる



例8) 改行コードのつく、固定長のテキストファイルも扱える



■ホストがUnix、Windowsの場合

ランダムファイル変換では、アプリケーションで使用する、おもに固定長・固定欄のデータファイルの変換を行います。BASICのランダムファイルやCOBOLの順ファイルなどがおもな対象です。バイナリ変換ができるのも大きな特長です。

以下に、ランダムファイル変換で可能なデータ変換機能、レコード編集機能、EOFコードの扱い、BOMコードの扱いを示します。

●データ変換機能

ランダムファイル変換のデータ変換機能を下表に示します。

データ変換機能		変換方法
ホスト側データ	↔ Win側データ	
バイナリ	↔ バイナリ	バイナリ変換
JIS8/ASCII	↔	JIS8/ASCII
Unicode(UCS-2)		Unicode(UCS-2)
Unicode(UTF-8)		Unicode(UTF-8)
漢字(EUC、シフトJIS)	↔	シフトJIS漢字のみ
〃	↔	Unicode(UCS-2)
〃	↔	Unicode(UTF-8)
〃	←	シフトJIS漢字、JIS8混在
ANK・漢字混在	↔	JIS8、シフトJIS漢字混在 Unicode(UCS-2) Unicode(UTF-8)
文字形式数値	↔	文字、WinCOBOL *1、2進 *2
符号なしゾーン形式	↔	〃
符号つきゾーン形式	↔	〃
符号なしパック形式	↔	〃
符号つきパック形式	↔	〃
符号なしBCD形式	↔	〃
2数形式数値 *3	↔	〃
日付データ	↔	日付データ
特殊データ	↔	特殊データ

- *1) Windows COBOL の符号なしゾーン/パック形式、符号つきゾーン/パック形式
- *2) Windows COBOL の COMP-4 や COMP-5、Visual Basic や C/C++ の整数型などの形式
- *3) COBOL の COMP-4 や COMP-5 などの形式
- *4) Numeric、DispZone、ZoneDisp、DispPack、PackDisp、DispBin、BinDisp 変換
AlphaZone、ZoneAlpha、AlphaPack、PackAlpha、AlphaBin、BinAlpha 変換
- *5) ZoneDisp、DispZone、ZoneZone、ZonePack、PackZone、ZoneBin、BinZone 変換
ZoneAlpha、AlphaZone 変換

- *6) PackDisp、DispPack、PackZone、ZonePack、PackPack、PackBin、BinPack 変換
PackAlpha、AlphaPack 変換
- *7) BinDisp、DispBin、BinZone、ZoneBin、BinPack、PackBin、BinBin 変換
BinAlpha、AlphaBin 変換

◆注意 ----- 浮動小数点には未対応

ホスト側もWin側も、浮動少数点型には対応していません。

●レコード編集機能

ランダムファイル変換では、Win側のレコード長を明示的に指定します。

しかし、データ内容に応じて項目長変更などのレコード編集機能が必要になります。明示的に指定しない限り改行コードをつけないことなどは、大きな特長です。

下表にランダムファイル変換のレコード編集機能を示します。

レコード編集機能

機 能	説 明
タブ拡張	タブを連続スペースに置き換える(ホスト→Win)
Winレコード長指定	Win側のレコード長を指定できる
項目別変換	項目別に分けて変換できる
項目削除	不要な項目を削除できる
項目長変更	項目長を変更できる
項目組み替え	項目の順番の組み替えができる
空項目挿入	空項目を作ることができる
定数挿入	任意の値を任意の形式で挿入することができる
改行コード挿入	レコードの途中・末尾に、改行コードを挿入できる

●EOFコードの扱い

ランダムファイル変換は、基本的にデリミタや改行コードを使わず、あってもただの一般文字扱いにします。EOFコードについても同様に一般文字扱いし、物理EOFでEOFにします。これは、バイナリ変換ができることを前提にしているためです。

EOFコードの扱い

機 能	説 明
EOFコード透過	EOFコードを一般の文字扱いする(Win→ホスト*)

*) ホストがWindowsの場合は、ホスト⇔Win

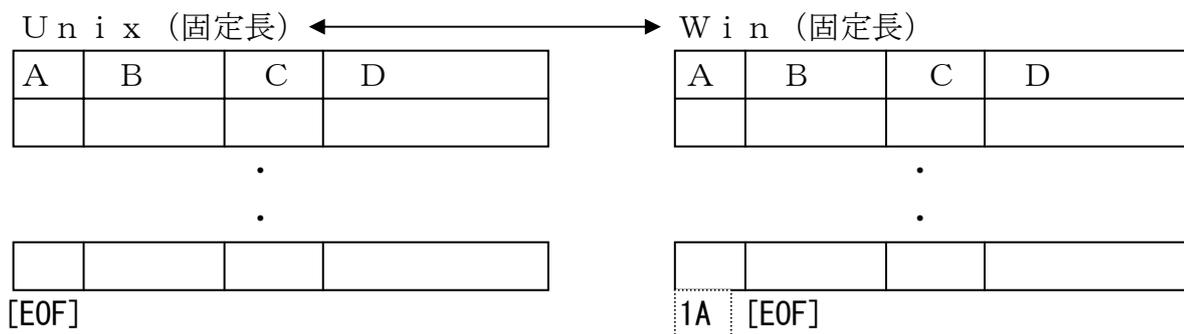
●BOMコードの扱い

ランダムファイル変換は、基本的にBOMコードを使わず、あってもただの一般文字扱いにします。これは、バイナリ変換ができることを前提にしているためです。

BOMコードの扱い

機 能	説 明
BOMコード透過	BOMコードを一般の文字扱いする

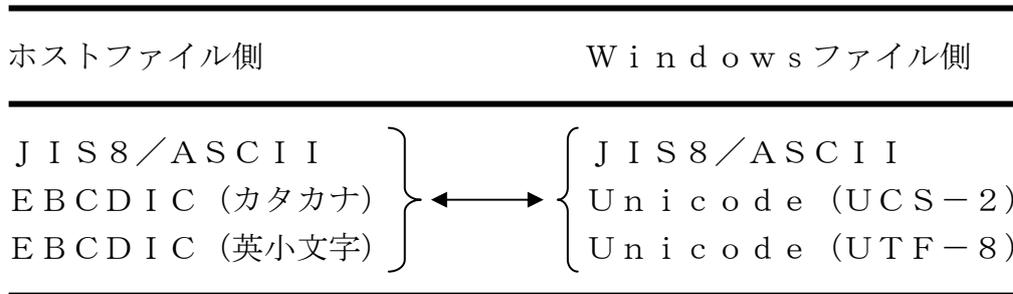
例) ホスト側が可変長／固定長、Win側が固定長の変換になり、項目別変換ができる



2. 11 ANK変換

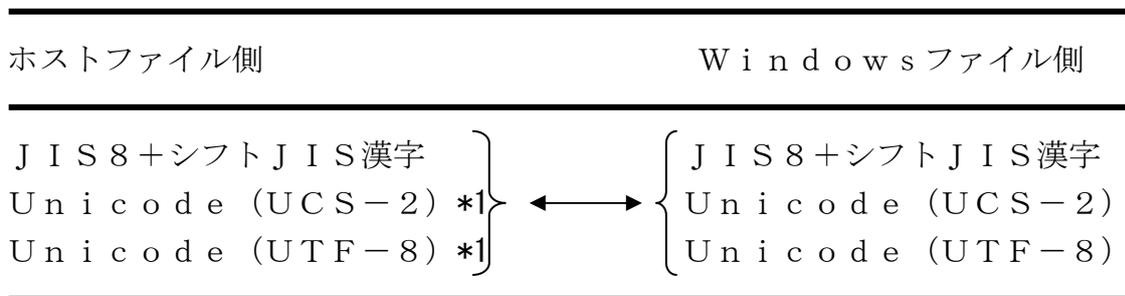
● ANK変換 (Ank変換)

ANK変換とは、半角文字の変換機能のことです。ホストが汎用機・オフコンの場合のANK変換は、つぎに示すように9通りの変換ができます。



ホストがUnixの場合のANK変換は、ホストファイル側がJ I S 8 / A S C I I固定になります。ホストがWindowsの場合は他にU n i c o d e (U C S - 2・U T F - 8)が指定できます。

この切り替えは、ホスト側は変換設定/ANKコードで行います。Windows側はWinオプションからエンコーディングの設定で行います。また、ANK変換という名前にそぐわず、特例としてシフトJ I S漢字を素通しにする変換もできます。



*1) ホストがWindowsのみ指定可能

● J I S 8 / A S C I Iコード

J I S 8 / A S C I Iコードは、パソコンで標準的に使われているコード系です。一部のホストでも使われています。

本書ではJ I S 8コードとA S C I Iコードを合わせてJ I S 8 / A S C I Iコードと呼んでいますが、本当は違うものです。A S C I Iコードは米国標準の7ビットコード系、J I S 8コードはそれにカタカナを加えて8ビットに拡張した日本のコード系、とっておいて大過ありません。なお、国内のパソコン関係で「A S C I Iナントカ」というときは、本当はJ I S 8コードのことです。つぎに示す、

スペース	2 0
“0” ~ “9”	3 0 ~ 3 9
“A” ~ “Z”	4 1 ~ 5 A
“a” ~ “z”	6 1 ~ 7 A

のコードぐらひはソラで覚えておくと、何かと便利です。

●EBCDICコード

EBCDICコードは、汎用機を中心として、ホストで多く用いられているコード系です。事実上の国際標準になっていますが、規格化はされていません。そのため、メーカー・システムごとに少しずつ違いがあります。

また、通常EBCDICコードではカタカナと英小文字が同時に使えないことも重要なポイントです。カタカナと英小文字のコードが重複しているためです。たとえば、8 1 Hというコードが表す文字は、

カタカナ版	8 1 = “ア”
英小文字版	8 1 = “a”

という具合です。なお、国内ではEBCDIC（英小文字）コードはあまり使われません。しかし、国外との電文の授受がある部署ではこれのごくふつうに使われています。国際的にはカタカナ版はまったく通用しないことを覚えておいたほうがよいでしょう。ファイル名にカタカナを使ったりすると、国外のシステムとはファイル転送すらできないことがあります。F*TRAN+では、

EBCDIC（カタカナ）用のANK変換表 EBCDIC（英小文字）用のANK変換表

の2つを提供し、どちらか一方に切り替えて使えるようにしています。つぎに示す、

スペース	4 0
“A” ~ “I”	C 1 ~ C 9
“J” ~ “R”	D 1 ~ D 9
“S” ~ “Z”	E 2 ~ E 9
“0” ~ “9”	F 0 ~ F 9

のコードぐらひはソラで覚えておくと、何かと便利です。

● J I S 8 / A S C I I コードと E B C D I C コードの対応関係

J I S 8 / A S C I I コードと E B C D I C コードでは、一部の特殊文字の対応がとれていません。一方にあってもう一方にはない特殊文字がいくつかあるのです。そのため、両者のコード変換規則はメーカー・システムによってバラバラなのが現状です。

F * T R A N + では、ほぼ標準的と思われる A N K 変換表を提供し、必要ならそれを利用者が簡単に変換できるようにして、この問題に対処しています。

2. 12 漢字変換

F * TRAN+の漢字変換機能には、

Ank・漢字 (KI/KO付) 混在変換 漢字項目変換

と呼ぶ、2つの変換方法があります。

● Ank・漢字 (KI/KO付) 混在変換

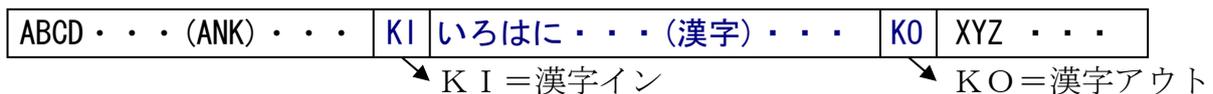
テキストファイル変換では、ホストファイル側で、レコードごとに漢字の位置が固定していないタイプのファイルを変換できます。ソースプログラムがその典型です。

データファイル変換やランダムファイル変換では、ホストファイル側にANK・漢字混在の項目があるタイプのファイルを変換できます。ただし、扱いにくいのであまり使われません。

どちらも、ANKから漢字への切り替えをKIと呼ぶコードで行い、漢字からANKへの切り替えをKOと呼ぶコードで行います。KI/KOがついているからどの桁に漢字が現れてもよいのです。

そして、EUC (Unix)、シフトJIS、Unicode (Windows) のように、KI/KOの概念がない漢字データ (EUCの場合は、G3等のシフトコード付きの漢字を含む) も扱えます。

このタイプの漢字サポートをF * TRAN+ではANK・漢字 (KI/KO付) 混在変換と呼んでいます。



● 漢字項目変換

データファイル変換とランダムファイル変換の場合、ホストファイルからWindowsファイルへの変換のときは、KI/KOがなくても変換できます。逆のWindowsファイルからホストファイルへの変換のときも、KI/KOをつけなくて変換することができます。

どちらも、ANK何バイト、漢字何バイト . . . と指示できるからです。これを漢字項目変換と呼んでいます。



テキストファイル変換の場合は、これに相当する機能はありません。

●半角⇄全角変換機能（文字種変換）

Windowsファイルでは、半角文字・全角文字混在の項目が、ごくふつうに使われています（禁止しているソフトもあります）。

一方、ホストファイル側では（ホスト上では）、半角文字・全角文字混在の項目は非常に扱いにくいものです。

そのため、F*TRAN+では文字種変換機能として半角⇄全角変換機能をサポートしています。全角文字を半角化する場合には

ANK化（AnkiZe）変換

半角文字を全角化する場合には

漢字化（KanjiZe）変換

にて行います。半角文字・全角文字混在の項目に対して適用すると、ANK化変換は自動的に全角文字を対応する半角文字に変換し、半角文字のみの項目にします。また、漢字化変換は自動的に半角文字を対応する全角文字に変換し、全角文字のみの項目にします。

Win→ホスト方向の変換（ホストがUnix、Windowsの場合は、ホスト→Win方向も対象）で、半角・全角混在の項目に対して漢字項目変換をすると半角文字が全角文字になりますが、この場合、半角スペースは特別扱いです。半角スペースは2個1組で1個の漢字スペースとして扱います。半角スペースが奇数個つづくとき、最後に1個の半角スペースが残ることになります。それは漢字スペースに変換します。

例えば、半角から全角に変換する場合、

ワタシ△ハ△△△かもめ → ワタシ△ハ△△かもめ

となります。△は半角スペース、△は漢字スペースを意味しています。漢字化変換の場合は半角スペース1個を全角スペース1個に変換します。

◆注意 --- オーバーフローしやすい

この機能を使うときは、項目長やレコード長が全角化の場合は増え（最大で2倍になる）、半角化の場合は減る（最小で半分）になることを計算に入れておかなければいけません。特に全角化の場合はオーバーフローに注意してください。

2. 13 コード変換方式のサポートパターン

利用者が自由に以下の項目を設定し、コード変換方式をアレンジしたり、新たに登録したりできます。

●ホストが汎用機・オフコンの場合

名前	コード変換方式につける名前（8文字以内）
コメント	それにつけるコメント（40文字以内）
ANKコード	J I S 8 / A S C I I E B C D I C（カタカナ／英小文字）
漢字コード	J I S 漢字 J I S 漢字 + 8 0 8 0 H J I S 漢字 + 8 0 8 0 H（ユニシス） N E C J I P S（E） " J I P S（J） " 内部コード（E） " 内部コード（J） 東芝漢字（漢字コード自体は J I S） I B M 漢字
漢字イン *1	0～3バイト
漢字アウト *1	0～3バイト
シフト節約度 （Win→ホストファイル変換時）	弱／中／強
ホスト漢字スペース	全角／半角2個（ユニシスは全角1／全角2あり）
W i n 漢字スペース *5	全角／半角2個（Unicodeは半角1個あり）
ホスト拡張漢字 *2 “=”の代わり	ゲタ変換／クエスチョン変換 ゲタ変換時の“=”の代わりに全角文字 半角スペース2個（2020H）にも設定できる
W i n 拡張漢字 *3 “=”の代わり	ゲタ変換に固定 “=”の代わりに全角文字 半角スペース2個（2020Hか4040H）にも設定できる
ホストエンディアン	ビッグ（正順）／リトル（逆順）
漢字対応表	漢字コード変換テーブルの設定 *4

●ホストがUnix (EUC) の場合

名前	コード変換方式につける名前 (8文字以内)
コメント	それにつけるコメント (40文字以内)
ホスト漢字スペース	全角 (A1A1H) / 半角2個 (2020H) 置換する / しない
Win漢字スペース*5	全角 / 半角2個 (Unicodeは半角1個あり) 置換する / しない
半角カタカナ	通す / ゲタ変換
“=” の代わり (Win→ホストファイル変換時)	ゲタ変換時の“=”の代わりの半角文字
未定義カタカナ (ホスト→Winファイル変換時)	“=” 変換固定
ホスト拡張漢字 *2 “=” の代わり	ゲタ変換 / クエスチョン変換 ゲタ変換時の“=”の代わりの全角文字 半角スペース2個 (2020H) にも設定できる
Win拡張漢字 *3 “=” の代わり	ゲタ変換に固定 “=” の代わりの全角文字 半角スペース2個 (2020H) にも設定できる
ホストエンディアン	ビッグ (正順) / リトル (逆順)
ホスト COBOL ベンダ	ホスト COBOL のベンダを指定する
漢字対応表	漢字コード変換テーブルの設定 *4

●ホストがWindows (シフトJIS)、Unix (シフトJIS) の場合

名前	コード変換方式につける名前 (8文字以内)
コメント	それにつけるコメント (40文字以内)
ホスト漢字スペース	全角 (8140H) / 半角2個 (2020H) 置換する / しない
Win漢字スペース*5	全角 / 半角2個 (Unicodeは半角1個あり) 置換する / しない
ホスト拡張漢字 *2 “=” の代わり	ゲタ変換 / クエスチョン変換 ゲタ変換時の“=”の代わりの全角文字 半角スペース2個 (2020H) にも設定できる
Win拡張漢字 *3 “=” の代わり	ゲタ変換に固定 “=”の代わりの全角文字 半角スペース2個 (2020H) にも設定できる
ホストエンディアン	ビッグ (正順) / リトル (逆順)
ホスト COBOL ベンダ	ホスト COBOL のベンダを指定する
漢字対応表	漢字コード変換テーブルの設定 *4

●ホストがWindows (Unicode) の場合

名前	コード変換方式につける名前 (8文字以内)
コメント	それにつけるコメント (40文字以内)
ホスト漢字スペース	全角/半角2個/半角1個 置換する/しない
Win漢字スペース*5	全角/半角2個 (Unicodeは半角1個あり) 置換する/しない
半角カタカナ “=”の代わり (Win→ホストファイル変換時)	8EH+カナコード/全角化/ゲタ変換 ゲタ変換時の“=”の代わりの半角文字
未定義カタカナ (ホスト→Winファイル変換時)	“=”変換固定
ホスト拡張漢字 *2 “=”の代わり	ゲタ変換/クエスチョン変換 ゲタ変換時の“=”の代わりの全角文字 半角スペース2個 (2020H)にも設定できる
Win拡張漢字 *3 “=”の代わり	ゲタ変換に固定 “=”の代わりの全角文字 半角スペース2個 (2020H)にも設定できる
ホストエンディアン	ビッグ (正順) / リトル (逆順)
ホスト COBOL ベンダ	ホスト COBOL のベンダを指定する
バイトオーダーマーク	あり/なし/自動*6
漢字対応表	漢字コード変換テーブルの設定 *4

*1) 漢字項目変換のときは使われない

*2) 漢字対応表に定義されないホスト拡張漢字、外字 (ユーザー定義文字) も含む

*3) 漢字対応表に定義されないWin拡張漢字、94区を越える漢字

*4) 通常は、対応づけができる拡張漢字コードを登録する

*5) Windowsコードにより以下の値をとります。

シフトJIS : 全角 (8140H)、半角 (20H)

Unicode (UCS-2) : 全角 (U+3000)、半角 (U+0020)

Unicode (UTF-8) : 全角 (E38080H)、半角 (20H)

*6) BOMの値はUnicodeの種類により以下の値をとります。

UCS-2リトルエンディアン : FFFE H

UCS-2ビッグエンディアン : FEFF H

UTF-8 : EFBBBF H

基本的に、このパターンに合えばコード変換ができます。

2.14 ホスト漢字のレパートリー

先に述べたサポートパターンに合うものとしては、Ank・漢字(KI/KO付)混在変換の場合、

富士通	J E F (12ポイント) ---- 一番ふつうのもの J E F (9ポイント) F 9 4 5 0内部コード (F C O N VのN C指定で変換したもの)
日立	K E I S
三菱	M E L C O M漢字
NEC	J I P S (E) NEC標準の漢字体系 J I P S (J) (ANKも漢字もJ I Sコードにしたもの) 内部コード (E) オフコン・ミニコンで多く使われる 内部コード (J) (めったに使われない)
東芝	東芝標準漢字
日本ユニシス	L E T S - J (旧日本ユニバック系)
カシオ	カシオ標準漢字
I B M	I B M漢字
新J I S	各社ホストのユーティリティで処理したもの
旧J I S	〃
U n i x	E U C シフトJ I S
W i n d o w s	シフトJ I S U n i c o d e (U C S - 2) U n i c o d e (U T F - 8)

などがあります。

漢字項目変換の場合は、KI/KOを合わせる必要がないので、さらに多くのメーカー・システムに適合します。

漢字変換方式は、1つのコード変換表につき15種類まで登録できます。よく使用されるものは、はじめから標準提供のコード変換表に登録されています。

◆注意 ---- I B M漢字 (I B M社の内部漢字コード) もサポートしています

F * T R A N +では、I B M漢字コードを内部テーブル変換によって求めています。そのため、わざわざホスト側のユーティリティを使ってJ I S漢字に直す必要はありません。

I B M漢字以外の漢字コードはJ I S - C 6 2 2 6 / X 0 2 0 8と同一、またはそれに準拠している必要があります (漢字コードを計算で求めるため)。

2. 15 漢字コード変換の仕組み

F * TRAN+の漢字コード変換の仕組みはつぎのとおりです。

●ホストの漢字

ホストの漢字は、大別するとつぎの2つになります。

1. 一般漢字（JIS第1／第2水準、多くのホストで共通の漢字）
2. 拡張漢字（ホストによって異なる固有の漢字、ユーザー定義文字を含む）

（注）ホストによっては、一般漢字エリアの中に特殊な漢字が含まれる場合があります。

●Windowsの漢字

Windowsの漢字も、大別するとつぎの2つになります。

1. 一般漢字（JIS第1／第2水準、多くのホストと共通の漢字）
2. 拡張漢字（一般漢字以外で追加された漢字、ユーザー定義文字を含む）

（注）一般漢字エリアの一部の漢字は、ホストの一般漢字エリアの同位置に定義されていない場合が多くみられます。

●ホストの漢字 ⇔ Windowsの漢字 処理順序

基本的に、一般漢字と一部の拡張漢字は正当な変換が可能で、多くの拡張漢字は正当な変換が不可能です。ただし、漢字変換表をうまく利用すれば意図した変換ができます。

F * TRAN+では、つぎの順序で漢字コードの変換を行います。

漢字対応表に定義されている漢字の変換

ふつうは、“株”のような対応づけができる拡張漢字を対象にしますが、一般漢字エリアの中に含まれる特殊な漢字の変換を含めることもできます。



一般漢字エリアの漢字変換

一般漢字エリアの漢字は、計算によって求められます。
（IBM漢字は内部テーブル変換）



漢字対応表に定義されない拡張漢字

ホストとWindowsで対応づけができない漢字を、ユーザーが選択した方式で変換します。

2. 16 漢字対応表について

漢字対応表は、F * TRAN+の漢字コード変換に反映することができる漢字変換テーブルです。漢字対応表のメンテナンスは、F * TRAN+に標準提供されている漢字対応表エディタによって簡単に行うことができます。ふつうは、メーカー単位で標準提供されている漢字対応表をベースに使用します。

漢字対応表に定義された情報は、データ変換時に読み込まれ、最優先の変換になります。漢字対応表に定義される基本パターンはつぎのとおりです。

1. Windows 漢字 (またはコード) \longleftrightarrow ホストの漢字コード
(双方向)
2. Windows 漢字 (またはコード) \longrightarrow ホストの漢字コード
(片方向)
3. Windows 漢字 (またはコード) \longleftarrow ホストの漢字コード
(片方向)

詳細は、4. 14 漢字対応表エディタをご覧ください。

2.17 変換できないホストの拡張漢字の扱い

F*TRAN+は、漢字対応表に定義されないホストの拡張漢字の変換をつぎのように扱います。ここでいう拡張漢字とは、ふつうの拡張漢字と外字（ユーザー定義文字）の総称です。つぎの、

ゲタ変換

クエスチョン変換

と呼ぶ、2通りの変換ができます。以前サポートしていたゼロイー変換が設定されている古いコード変換表ファイルを読み込んだ場合はゲタ変換として扱われます。

以下、富士通 J E F の「株」 = 7 8 D 5 H を例にとって、どのように変換できるかを見ていきます。

●ゲタ変換

ホストファイルの拡張漢字を、Windows ファイルではゲタ (=)、あるいはその代替りの全角文字に変換させる方式です。例外として、半角スペース 2 個 (2 0 2 0 H) も代替りの文字として使えます。

ホストファイル	Windows ファイル
XXXX	→ 81AC = “=” (あるいは、代替りの全角文字)
78D5 = “株”	→ 81AC = “=” (あるいは、代替りの全角文字)

この方式には、**どこに拡張漢字が使われているか、ひと目でわかる**
“=” 以外の代替りの全角文字に設定することもできる
2バイト→2バイト変換なので、桁ずれを起こさない

などの特長があります。

その反面、**パソコンからホストへの変換には使えない**
ホストの元のコードを知る手掛かりがない

という難点もあります。

なお、出荷時設定はすべてこの「ゲタ変換」になっています。拡張漢字をそれほど使わないなら、この設定が無難です。

●クエスチョン変換

ホストの拡張漢字を、パソコンでは全角“?” + 半角16進4桁のパターンで表す方式です。

ホストファイル		Windowsファイル
XXXX	←→	“(XXXX)”
78D5 = “株”	←→	“(78D5)”

この方式には、
どこに拡張漢字が使われているか、わかりやすい
ホストのコードがひと目でわかる
文字表現なので、エディタによる置き換えなどがしやすい
パソコンからホストへ変換するときも、元のホストコードに戻る

などの特長があります。いろいろと、扱いやすいのです。

その反面、
2バイト⇔8バイト（UCS-2の場合は14バイト）変換なので、
桁ずれを起こす
言い替えれば、オーバーフローしやすい

という難点もあります。

2. 18 ホストの一般漢字と拡張漢字の区別

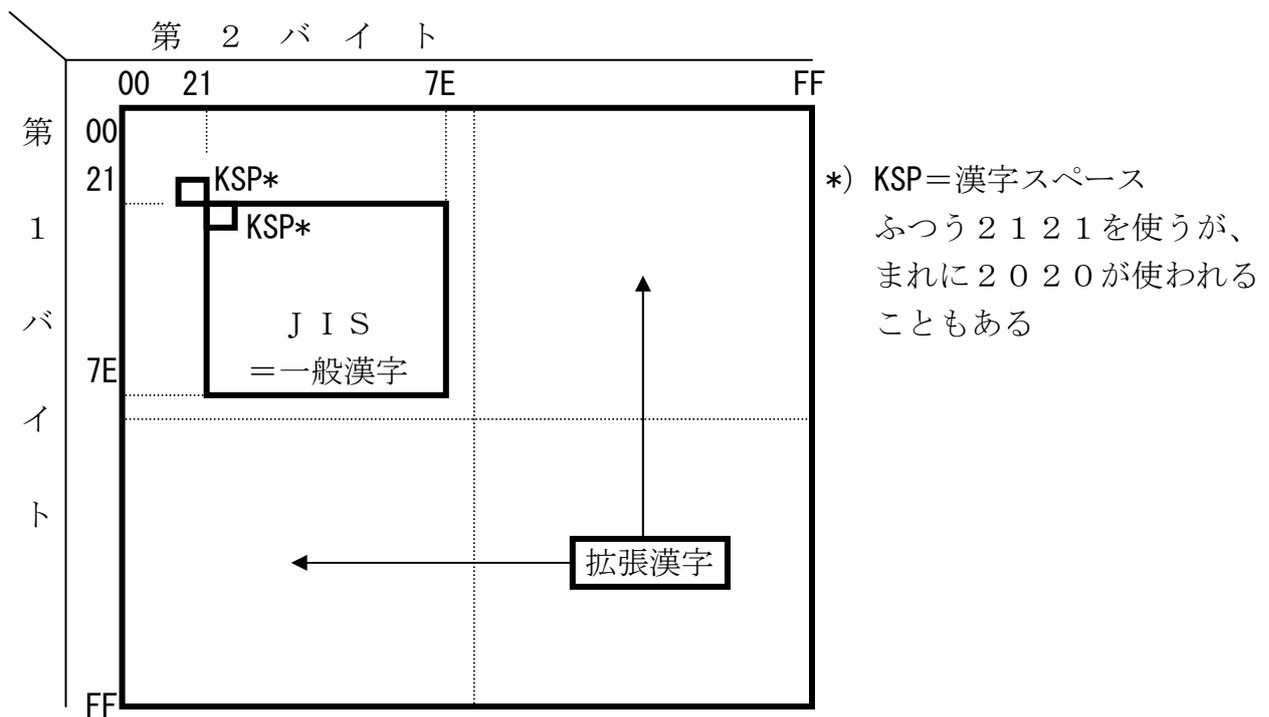
F * TRAN+がホストファイル側の一般漢字と拡張漢字をどのように区別しているかを述べます。

「一般漢字」とは、J I S句点コードで1区1点～94区94点の範囲内の漢字すべてを意味します。「拡張漢字」とは、一般漢字以外のすべてのコード、すなわち、1区1点～94区94点の範囲以外のコードを持つ漢字すべてです。

J I Sタイプ、NEC漢字、J I S+8080タイプ、EUC、I B M漢字、に分けて説明します。

● J I Sタイプ

J I Sタイプの1区1点～94区94点内の漢字とは、16進コードでいい替えると第1バイトが21H～7EHかつ第2バイトも21H～7EHの範囲に収まる漢字を指します。そして、この範囲内に収まらないコードの漢字を拡張漢字と呼びます。



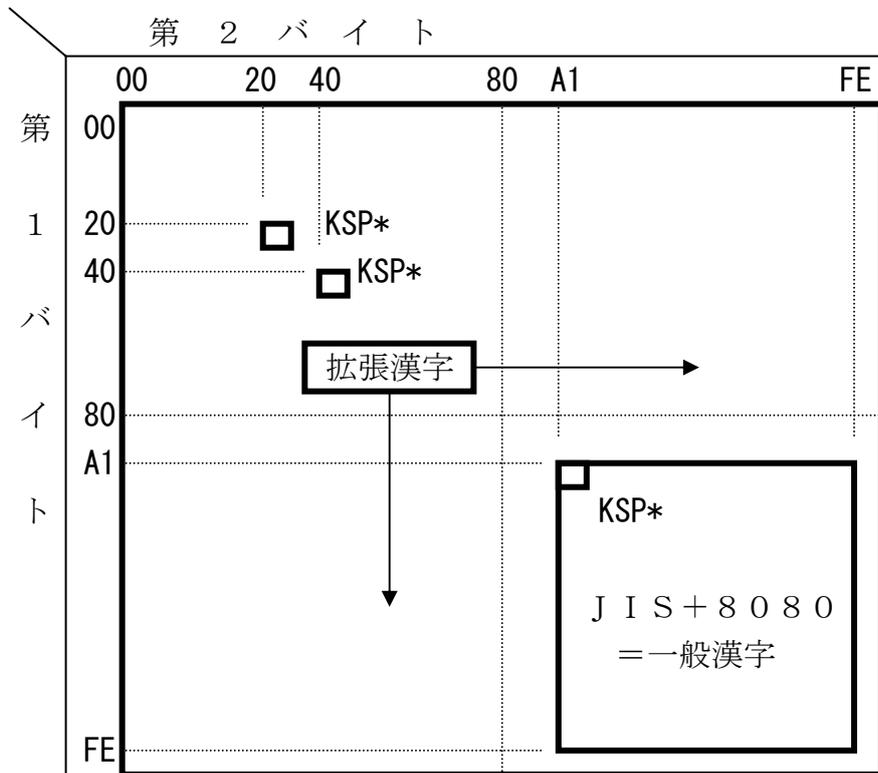
このタイプでは、一般漢字の領域内にそのメーカー・システム固有の特殊漢字を含めていることが多いものです。ですが、図からわかるようにF * TRAN+ではそれは拡張漢字としては扱いません。

● NEC漢字

NEC漢字の場合は単純な判定はできないので、いったんJ I Sタイプに変換してから一般漢字・拡張漢字の判定をします。

● J I S + 8 0 8 0 タイプ

J I S + 8 0 8 0 タイプの1区1点～94区94点内の漢字とは、16進コードでいい替えると、第1バイトがA1H～FEHかつ第2バイトもA1H～FEHの範囲に収まる漢字を指します。そして、この範囲内に収まらないコードの漢字を拡張漢字と呼びます。ただし、ホスト側の漢字スペースを半角2個に設定しているときは、それはこの範囲には収まりません。



- *) KSP=漢字スペース。2020、4040、A1A1がある
- 2020：J I S 8 / A S C I I 系のシステムで、まれに使われることがある
- 4040：E B C D I C 系のシステムで多く使われる
- A1A1：J I S + 8 0 8 0 タイプの全角スペース

このタイプの場合、一般漢字の領域内にそのメーカー・システム固有の特殊漢字を含めることはあまりありません。

● E U C

E U C の一般漢字は J I S + 8 0 8 0 と同じですが、拡張漢字は 8 F H + (J I S + 8 0 8 0) の3バイトコードが標準です。

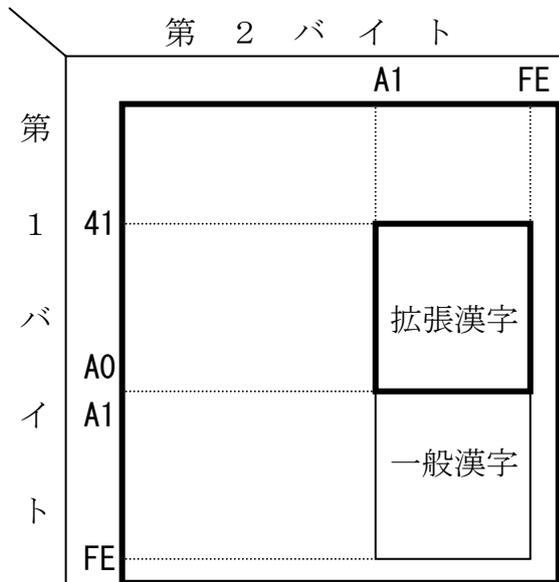
● I B M 漢字

I B M 漢字の場合は、16進コードで第1バイトが41H～7FHかつ第2バイトが41H～FEHの範囲を漢字（ただし、漢字スペースは4040H）として扱い、内部テーブルによってコード変換を行います。漢字エリアの中には、拡張漢字/ユーザー選定文字の領域も含まれています。

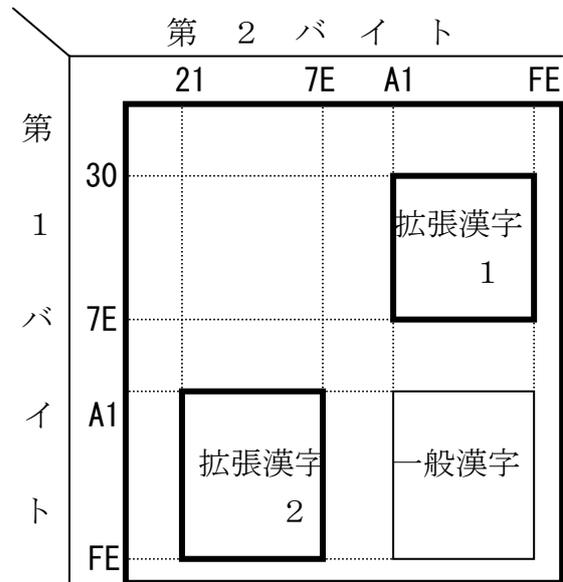
2. 19 ホストの拡張漢字の領域

F*TRAN+では、拡張漢字の領域をつぎのように認識しています。

● J I S + 8 0 8 0 系

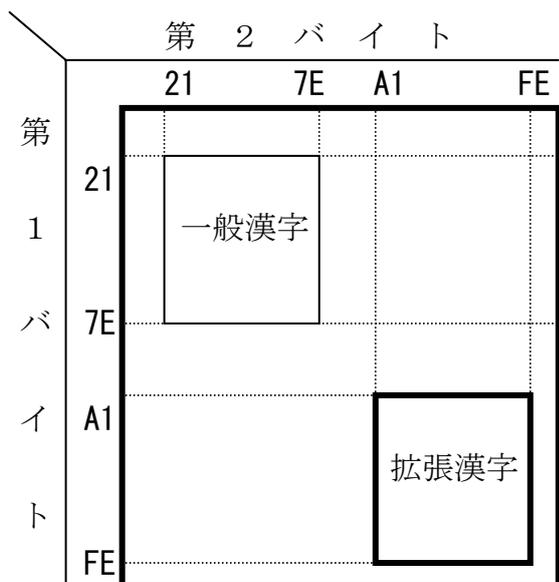


富士通 JEF、日立 KEIS、三菱 MELCOM



日本ユニシス LETS-J

● J I S 系



東芝漢字、NEC漢字 JIPS

● E U C

一般漢字は、J I S + 8 0 8 0 系と同じ。
拡張漢字は、つぎのタイプがあります。

標準：

8 F H + A 1 A 1 H ~ F E F E H

DEC漢字：

A 1 2 1 H ~ F E 7 E H

Super DEC漢字：

8 F H + A 1 A 1 H ~ F E F E H

8 F H + A 1 2 1 H ~ F E 7 E H

● I B M漢字

4 0 4 0 H、4 1 4 1 H ~ 7 F F E H の
範囲に一般漢字、拡張漢字が含まれます。

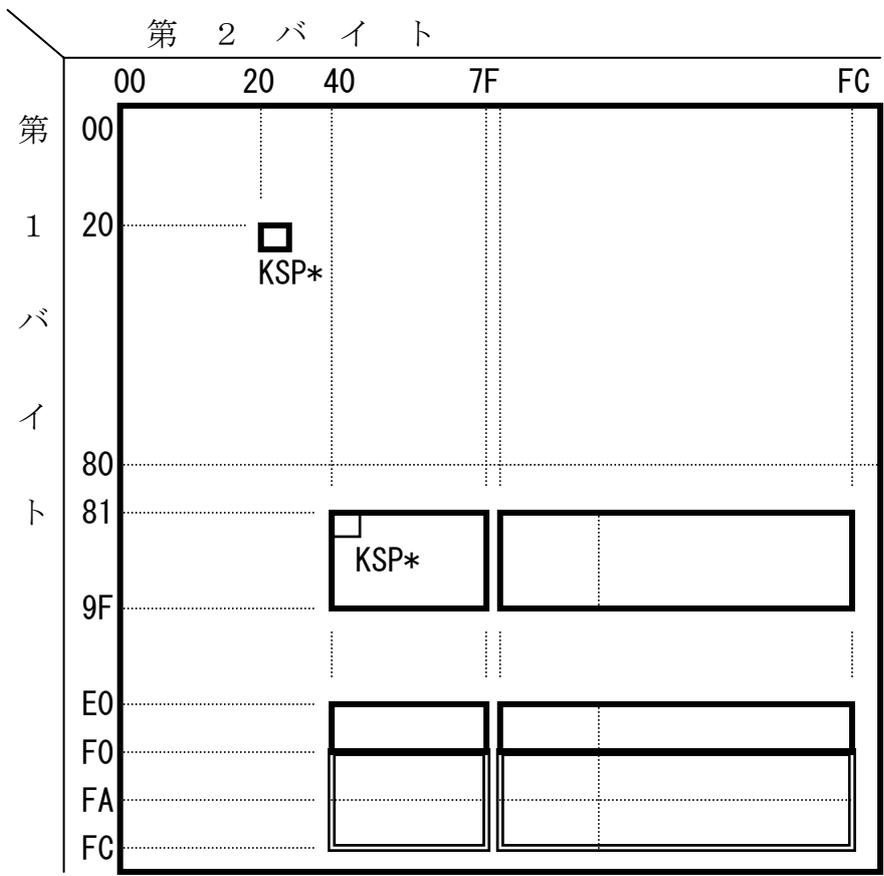
2. 20 変換できない Windows の拡張漢字の扱い

F*TRAN+は、漢字対応表に定義されない Windows の拡張漢字をつぎのように扱います。文字化けを防ぐためです。ここでいう拡張漢字とは、94区を超える漢字コードのことです。なお、以下はシフト JIS の場合です。Unicode にはあてはまりません。

シフト JIS のコード体系では、JIS の1区1点~94区94点に収まらない領域があります。JIS の区点コードを拡張して表現すれば、95区~120区 (シフト JIS では F040 ~ FCFC) の領域です。ここが拡張漢字の領域です (下図の の部分)。Windows 環境では、この領域を

外字 (1880字分) 95区~114区 (F040~F9FC)
IBM社選定拡張漢字 115区~120区 (FA40~FCFC)

に分けて利用しています。IBM社選定拡張漢字は、ホストの拡張漢字の中に割り付けられている漢字が多いので漢字対応表で定義することができます。しかし、それ以外の漢字は JIS コードに対応づけることができないので、ホストの漢字コードに変換するとき F*TRAN+ ではゲタ (=)、あるいはその代わりに全角文字に置き換えます (ゲタ変換に固定)。



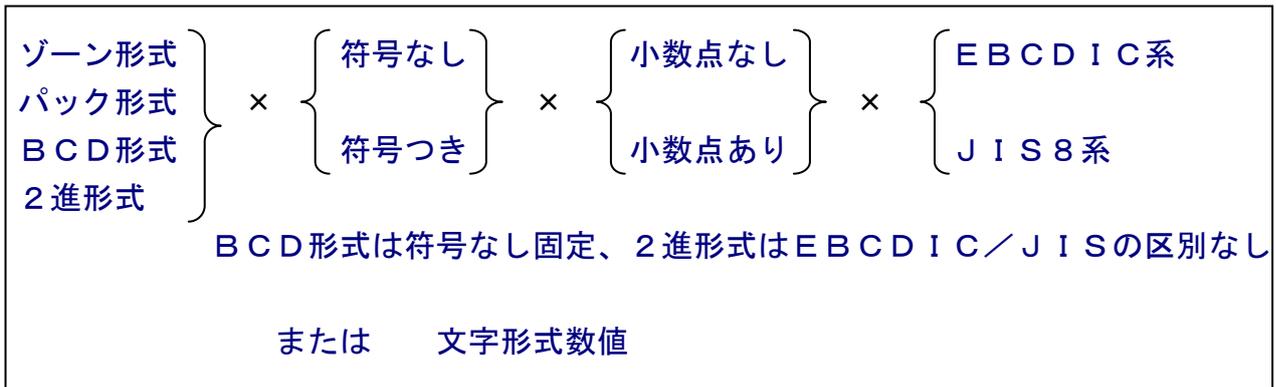
*) KSP=漢字スペース
 目的に応じ、2020と8140を使い分ける

2. 21 数値変換

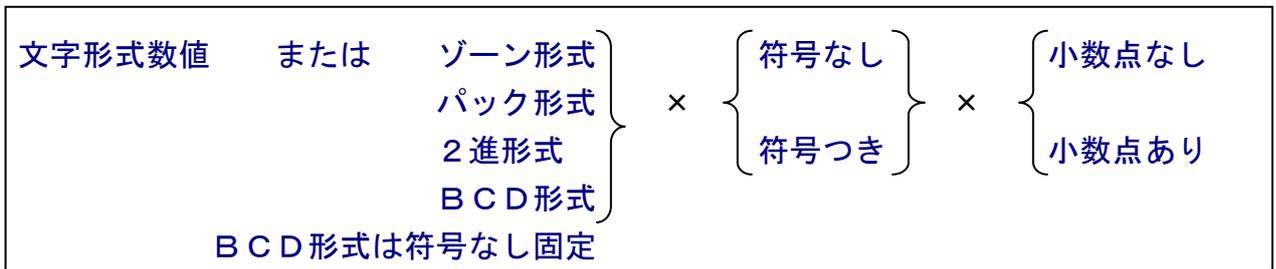
● **ゾーン形式とパック形式、BCD形式、2進形式、文字形式の変換をサポート**

F * TRAN+は、ホストで多く使われるCOBOL数値項目のゾーン形式とパック形式、BCD形式、2進形式、文字形式の数値変換をサポートしています。

ホストファイル側



Windowsファイル側



また、Windows側は、文字形式数値、Windows COBOLのゾーン形式とパック形式、BCD形式、2進形式をサポートしています。ゾーン形式、パック形式、BCD形式、2進形式は、上記のどの組み合わせでもかまいません。

● **ゾーン形式とは**

ゾーン形式は、COBOL数値項目のうち最も多く使われる形式です。1バイトで10進1桁を表します。外部10進数と呼ばれることもあります。文字形式数値に変換するのに都合のよい形式になっているからです。

● **パック形式とは**

パック形式は、COBOL数値項目のうち、ゾーン形式のつぎに多く使われる形式です。1バイトで10進2桁を表す格納効率のよい形式です。内部10進数と呼ばれることもあります。直接、演算の対象にできるからです。

●BCD形式とは

BCD形式は、POS端末等で使われているパック形式に似ている形式です。パック形式同様1バイトで10進2桁を表す格納効率のよい形式ですが、符号情報を持っていません。

●2進形式とは

2進形式は、COBOL数値項目、Visual Basic、C/C++などで使われる形式です。1～8バイトで表現可能な数値を表す最も格納効率のよい形式です。ただし、F*TRAN+は、アプリケーションが採用しているすべての2進形式をサポートしているわけではありません。

●F*TRAN+がサポートしている変換

F*TRAN+は、つぎの組み合わせで数値変換を行うことができます。

ホスト側		Windows側
Disp (文字形式)	⇔	Disp (文字形式)
Zone (ゾーン形式)		Zone (ゾーン形式)
Pack (パック形式)		Pack (パック形式)
Bin (2進形式)		Bin (2進形式)

*) Disp ⇔ Dispは、Numeric変換

*) パック形式にはBCD形式も含む

2. 22 汎用機・オフコン系のゾーン／パック形式

(1) ゾーン形式 (EBCDIC系)

● ゾーン形式 (EBCDIC系) の記録形式

たとえば、ゾーン形式7桁の数値は、下図のような形式で記録されます。

0	1	2	3	4	5	6
F n	F n	F n	F n	F n	F n	s n

1バイトのうち、上位4ビットをゾーン部と呼びます。図のFとsの部分です (Fは16進のFそのものです)。そして、下位4ビットで10進1桁を表します (図のnの部分、n=0~9)。

符号付きの場合、正負の区別はふつう最後の1バイトのゾーン部に記録されます。図のsの部分です。便宜上、sの部分符号部と呼び、Fの部分ゾーン部と呼び分けることにします。

桁数や小数点を表すものは何もありません。この項目をアクセスするプログラムで、それらを指定することになっています。F*TRAN+では「ピクチャ」でそれらを指定します。

● 符号部 (s) の値

下表のように、符号の有無、正負の区別によって符号部 (s) のとる値が違います。

ゾーン形式の符号部 (s) の値 (EBCDIC系)

	符号部 (s)
符号なし	常にF
符号あり	CまたはFなら正 Dなら負

ゾーン形式は、各バイトを“0”~“9”の文字として扱ってもよい記録形式になっています。ただし、符号つきゾーン形式の最後の桁は例外です。EBCDICコードでは、“0”~“9”=F0~F9であることを思い出してください。符号つきゾーン形式の最後の桁だけは文字扱いできない (数字としては読めない) ことがわかります。

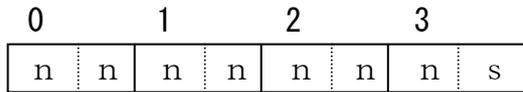
例) ゾーン形式 (EBCDIC系)

16進表現	=	文字表現	
F0 F0 F1 F2 F3	=	1 2 3	符号なし、小数部なし
F0 F0 F1 F2 C3	=	1 2. 3	符号つき、正、小数部あり
F0 F0 F1 F2 D3	=	- 1 2. 3	符号つき、負、小数部あり

(2) パック形式 (EBCDIC系)

●パック形式 (EBCDIC系) の記録形式

たとえば、パック形式7桁の数値は、つぎのような形式で記録されます。



1バイトを上位4ビット+下位4ビットに分け、それぞれが10進1桁を表します (図のnの部分、n=0~9)。ただし、最後の4ビット (図のsの部分) だけは例外で、数値の記録はしません。ここには、符号なしの場合は詰めものがされ、符号付きの場合は正負の区別が記録されます。ここを、符号部と呼ぶことにします。

パック形式でもやはり、桁数や小数点を表すものは何もありません。この項目をアクセスするプログラムで、それらを指定することになっています。F*TRAN+では「ピクチャ」でそれらを指定します。

●符号部 (s) の値

下表のように、符号の有無、正負の区別によって符号部 (s) のとる値が違います。

パック形式の符号部 (s) の値 (EBCDIC系)

	符号部 (s)
符号なし	常にF
符号あり	CまたはFなら正 Dなら負

例) パック形式 (EBCDIC系)

16進表現	=	文字表現	
00 12 3F	=	1 2 3	符号なし、小数部なし
00 12 3C	=	1 2. 3	符号つき、正、小数部あり
00 12 3D	=	- 1 2. 3	符号つき、負、小数部あり

(3) ゾーン形式 (J I S 8系)

● ゾーン形式 (J I S 8系) の記録形式

たとえば、ゾーン形式7桁の数値は、下図のような形式で記録されます。

	0		1		2		3		4		5		6
	3	n	3	n	3	n	3	n	3	n	3	n	s n

1バイトのうち、上位4ビットをゾーン部と呼びます。図の3とsの部分です(3は16進の3そのものです)。そして、下位4ビットで10進1桁を表します(図のnの部分、n=0~9)。

符号付きの場合、正負の区別はふつう最後の1バイトのゾーン部に記録されます。図のsの部分です。便宜上、sの部分符号部と呼び、3の部分ゾーン部と呼び分けることにします。

桁数や小数点を表すものは何もありません。この項目をアクセスするプログラムで、それらを指定することになっています。F*TRAN+では「ピクチャ」でそれらを指定します。

● 符号部 (s) の値

下表のように、符号の有無、正負の区別によって符号部 (s) のとる値が違います。

ゾーン形式の符号部 (s) の値 (J I S 8系)

	符号部 (s)
符号なし	常に3
符号あり	0または3なら正、Bなら負 (正確には0~7なら正、8~Fなら負)

ゾーン形式は、各バイトを“0”~“9”の文字として扱ってもよい記録形式になっています。ただし、符号つきゾーン形式の最後の桁は例外です。J I S 8コードでは、“0”~“9”=30~39であることを思い出してください。符号つきゾーン形式の最後の桁だけは文字扱いできない(負なら数字としては読めない)ことがわかります。

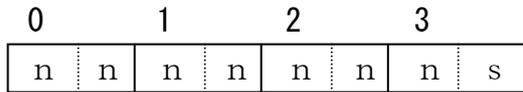
例) ゾーン形式 (J I S 8系)

16進表現		文字表現		
30 30 31 32 33	=	1 2 3		符号なし、小数部なし
30 30 31 32 33	=	1 2. 3		符号つき、正、小数部あり
30 30 31 32 B3	=	- 1 2. 3		符号つき、負、小数部あり

(4) パック形式 (J I S 8系)

● パック形式 (J I S 8系) の記録形式

たとえば、パック形式7桁の数値は、つぎのような形式で記録されます。



1バイトを上位4ビット+下位4ビットに分け、それぞれが10進1桁を表します (図のnの部分、n=0~9)。ただし、最後の4ビット (図のsの部分) だけは例外で、数値の記録はしません。ここには、符号なしの場合は詰めものがされ、符号付きの場合は正負の区別が記録されます。ここを、符号部と呼ぶことにします。

パック形式でもやはり、桁数や小数点を表すものは何もありません。この項目をアクセスするプログラムで、それらを指定することになっています。F*TRAN+では「ピクチャ」でそれらを指定します。

● 符号部 (s) の値

下表のように、符号の有無、正負の区別によって符号部 (s) のとる値が違います。

パック形式の符号部 (s) の値 (J I S 8系)

	符号部 (s)
符号なし	0 または 3 (正確には0~Fのどれでもよい)
符号あり	0 または 3 なら正、B なら負 (正確には0~7なら正、8~Fなら負)

例) パック形式 (J I S 8系)

16進表現		文字表現		
00 12 30	=	1 2 3	=	符号なし、小数部なし
00 12 33	=	1 2. 3	=	符号つき、正、小数部あり
00 12 3B	=	- 1 2. 3	=	符号つき、負、小数部あり

2. 23 オープン系 (Windows・Unix) のゾーン/パック形式

●ゾーン形式（オープン系）の記録形式

たとえば、ゾーン形式7桁の数値は、下図のような形式で記録されます。

	0	1	2	3	4	5	6	
符号なし	3	n	3	n	3	n	3	n

	0	1	2	3	4	5	6			
符号つき	3	n	3	n	3	n	3	n	s	s

数値は、基本的に“0”～“9”の文字が使われますが、符号つきの場合、正負の区別は最後の1バイトに数値の情報とともに記録されます。図のs sの部分です。

桁数や小数点を表すものは何もありません。この項目をアクセスするプログラムで、それらを指定することになっています。F*TRAN+では「ピクチャ」でそれらを指定します。

●符号つきバイト（s s）の値

以下の2つの表に示すように、正負の区別、数値、ベンダによって符号つき（s s）のとり値が決まります。

ゾーン形式の符号つきバイト（s s）の値（数値が正の場合）

数値	富士通	日立/MF/Am	NEC/Ai	Acucorp -Dca
+0	@ (4 0)	0 (3 0)	{ (7 B)	0 (3 0)
+1	A (4 1)	1 (3 1)	A (4 1)	1 (3 1)
+2	B (4 2)	2 (3 2)	B (4 2)	2 (3 2)
+3	C (4 3)	3 (3 3)	C (4 3)	3 (3 3)
+4	D (4 4)	4 (3 4)	D (4 4)	4 (3 4)
+5	E (4 5)	5 (3 5)	E (4 5)	5 (3 5)
+6	F (4 6)	6 (3 6)	F (4 6)	6 (3 6)
+7	G (4 7)	7 (3 7)	G (4 7)	7 (3 7)
+8	H (4 8)	8 (3 8)	H (4 8)	8 (3 8)
+9	I (4 9)	9 (3 9)	I (4 9)	9 (3 9)

MF = マイクロフォーカス、Am = Acucorp -Dcm、Ai = Acucorp -Dci

ゾーン形式の符号つきバイト (s s) の値 (数値が負の場合)

数値	富士通	日立/MF/Am	NEC/Ai	Acucorp -Dca
-0	P (5 0)	p (7 0)	} (7 D)	} (7 D)
-1	Q (5 1)	q (7 1)	J (4 A)	J (4 A)
-2	R (5 2)	r (7 2)	K (4 B)	K (4 B)
-3	S (5 3)	s (7 3)	L (4 C)	L (4 C)
-4	T (5 4)	t (7 4)	M (4 D)	M (4 D)
-5	U (5 5)	u (7 5)	N (4 E)	N (4 E)
-6	V (5 6)	v (7 6)	O (4 F)	O (4 F)
-7	W (5 7)	w (7 7)	P (5 0)	P (5 0)
-8	X (5 8)	x (7 8)	Q (5 1)	Q (5 1)
-9	Y (5 9)	y (7 9)	R (5 2)	R (5 2)

MF = マイクロフォーカス、Am = Acucorp -Dcm、Ai = Acucorp -Dci

●パック形式 (オープン系) の記録形式

オープン系のパック形式は、汎用機・オフコン系のパック形式 (EBCDIC系) と同じです。

2. 2.2 汎用機・オフコン系のゾーン/パック形式、(2) パック形式 (EBCDIC系) を参照してください。

2. 24 BCD形式

●BCD形式の記録形式

たとえば、BCD形式8桁の数値は、つぎのような形式で記録されます。

0		1		2		3	
n	n	n	n	n	n	n	n

1バイトを上位4ビット+下位4ビットに分け、それぞれが10進1桁を表します(図のnの部分、n=0~9)。

BCD形式はパック形式と似ていますが、符号情報を持たない形式です。

BCD形式でもやはり、桁数や小数点を表すものは何もあります。この項目をアクセスするプログラムで、それらを指定することになっています。F*TRAN+では「ピクチャ」でそれらを指定します。

例) BCD形式

<u>16進表現</u>	=	<u>文字表現</u>	
00 01 23	=	1 2 3	小数部なし
00 01 23	=	1 2 . 3	小数部あり

2. 25 2進形式

● **2進形式の記録形式**

たとえば、2進形式9桁の数値は、下図のような形式で記録されます。

	0	1	2	3					
符号なし	0	7	5	B	C	D	1	5	値：1 2 3 4 5 6 7 8 9

	0	1	2	3					
符号つき	F	8	A	4	3	2	E	B	値：- 1 2 3 4 5 6 7 8 9

2進形式は、少ないバイト数で大きな数値を表現可能な最も格納効率のよい形式です。そのため、格納された数値を確認するには専門的な知識が必要となり、判断が難しいという側面があります。一般には、アプリケーションの内部で処理されることが多い形式です。

F*TRAN+でサポートする2進形式は、

1～8バイトの範囲で表現できる整数／小数（マイナス値を含む）

です。具体的には、

<p style="color: blue;">COBOL の</p> <p style="color: blue;">Visual Basic の</p> <p style="color: blue;">C/C++ の</p>	<p style="color: blue;">BINARY、COMP、COMP-4、COMP-5</p> <p style="color: blue;">Integer、Long</p> <p style="color: blue;">short、int、long</p>
--	---

などが該当します。2進形式は、桁数や小数点を表すものは何もありません。この項目をアクセスするプログラムで、それらを指定することになっています。F*TRAN+では「2進ピクチャ」でそれらを指定します。

● **エンディアン（格納順）**

2進形式の場合、気をつけなければならないのはエンディアン（格納順）がある事です。エンディアンの違いを図にすると、つぎのようになります。

	0	1	2	3					
ビッグ（正順）	0	7	5	B	C	D	1	5	値：1 2 3 4 5 6 7 8 9

	0	1	2	3					
リトル（逆順）	1	5	C	D	5	B	0	7	値：1 2 3 4 5 6 7 8 9

一般には、ホスト側がビッグ（正順）で、Win側がリトル（逆順）です。

2. 26 パック/BCD/2進形式の桁数とバイト数の換算

●パック形式の桁数とバイト数の換算

パック形式の数値の桁数（整数部桁数＋小数部桁数）とバイト数の換算は、下表を参考にしてください。

パック形式の桁数とバイト数の換算表

桁 数	バイト数
1	1
2～3	2
4～5	3
6～7	4
8～9	5
10～11	6
12～13	7
14～15	8
16～17	9
18	10

●BCD形式の桁数とバイト数の換算

BCD形式の数値の桁数（整数部桁数＋小数部桁数）とバイト数の換算は、下表を参考にしてください。

BCD形式の桁数とバイト数の換算表

桁 数	バイト数
1～2	1
3～4	2
5～6	3
7～8	4
9～10	5
11～12	6
13～14	7
15～16	8
17～18	9

● 2進形式で表現可能な数値の範囲

2進形式で表現可能な数値の範囲は、つぎのとおりです。

符号なし2進形式の場合

バイト数	値の範囲	桁数
1	0~255	3
2	0~65,535	5
3	0~16,777,215	8
4	0~4,294,967,295	10
5	0~1,099,511,627,775	13
6	0~281,474,976,710,655	15
7	0~72,057,594,037,927,935	17
8	0~18,446,744,073,709,551,615	18*

符号つき2進形式の場合

バイト数	値の範囲	桁数
1	-128~127	3
2	-32,768~32,767	5
3	-8,388,608~8,388,607	7
4	-2,147,483,648~2,147,483,647	10
5	-549,755,813,888~549,755,813,887	12
6	-140,737,488,355,328~140,737,488,355,327	15
7	-36,028,797,018,963,968~36,028,797,018,963,967	17
8	-9,223,372,036,854,775,808~9,223,372,036,854,775,807	18*

*) 18桁ではオーバーフローするが、COBOL規格/F*TRAN+の上限です。

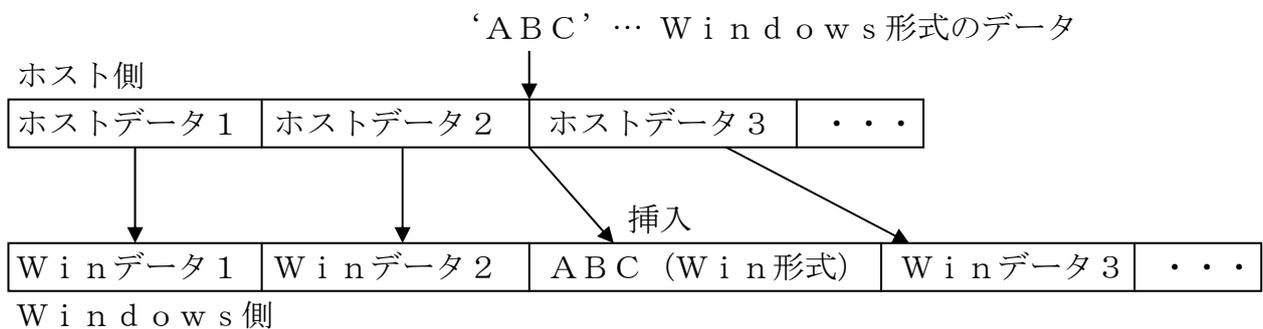
2. 27 定数挿入

データファイル変換とランダムファイル変換では、項目別に分けて変換できますが、その際に定数や変数を挿入することができます。

テキストファイル変換の場合は、これに相当する機能はありません。

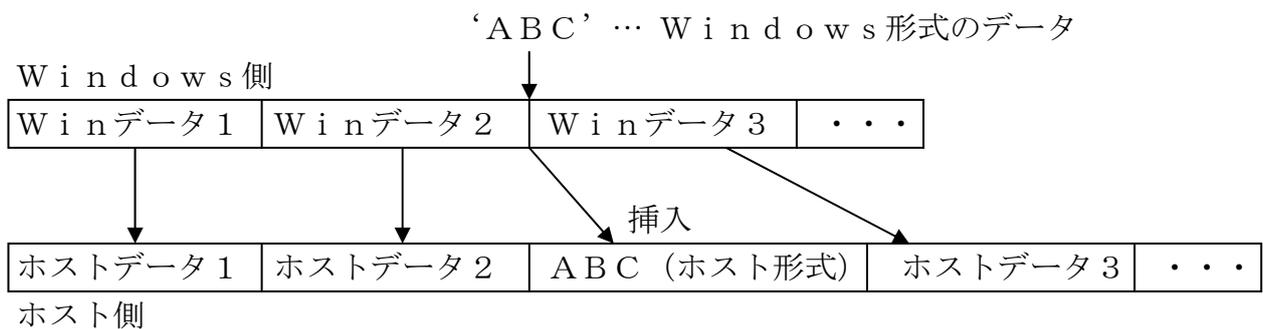
●ホスト→Win変換時

指定した定数は、Windows側に指定されたコード（シフトJIS、Unicode（UCS-2、UTF-8））に変換されてWindows側に挿入・出力されます。



●Win→ホスト変換時

指定した定数は、ホスト形式のデータに変換してから、ホスト側に挿入・出力されます。ホストがWindowsの場合も指定されたコード（シフトJIS、Unicode（UCS-2、UTF-8））に変換されて挿入・出力されます。



●列定数の種類

文字列定数

‘文字列’	文字列は最大256文字で、半角(?)でくくる。 ※ただし(‘), (*), (?)は(¥), (¥*), (¥?)で表し、(¥)自身は(¥¥)で表す。 (例) ‘Ank’, ‘漢字’, ‘Ank&漢字’, ”
--------------	--

16進列定数

{X x} ‘hh…’	hは0~9、A~F (a~f)、2桁で1バイト、 任意のバイト境界に半角スペースを挿入できる。 最大256バイトで、半角(?)でくくる。 (例) X‘1234’, x‘00 FF a1a1 ff’, x”
----------------------	---

汎用定数

Space	空白文字列 (半角、全角の空白からなる文字列)
LowValue	00H…の文字列、または00H…のバイナリデータ
HighValue	FFH…の文字列、またはFFH…のバイナリデータ
※ANK変換表をFF→FFにする必要がある。	

●数値定数の種類

整数定数

[−] n	(-)はマイナス符号、nは最大18桁 (例) 0、123、-99
--------------	-------------------------------------

小数定数

[−] n.m	(-)はマイナス符号、nとmは合わせて最大18桁 (例) 0.0、1.23、-99.9
----------------	--

16進定数

0 {X x} h…	hは0~9、A~F (a~f) で最大16桁 (8バイト) 途中の空白は不可 (例) 0x00、0XFF、0xee、0X1、0xffff
---------------------	--

汎用定数

Zero	0または0. 0
Min	最小値
Max	最大値

●指定できる定数

文字列変換

変換方法	文字列定数	16進列定数	Space LowValue HighValue
Ank変換	○	×	○
漢字変換	○	×	△ *1
Ank・漢字混在変換	○	×	○
バイナリ変換	×	○	△ *2
バイナリ反転変換	×	○	×

*1) Spaceのみ可

*2) Spaceのみ不可

数値変換

変換方法	数値定数	システム変数	Zero Min Max
文字形式へ変換	○	○	×
ゾーン形式へ変換	○	○	○
パック形式へ変換	○	○	○
2進形式へ変換	○	○	○

2. 28 ホスト→Winテキストファイル変換のしくみ

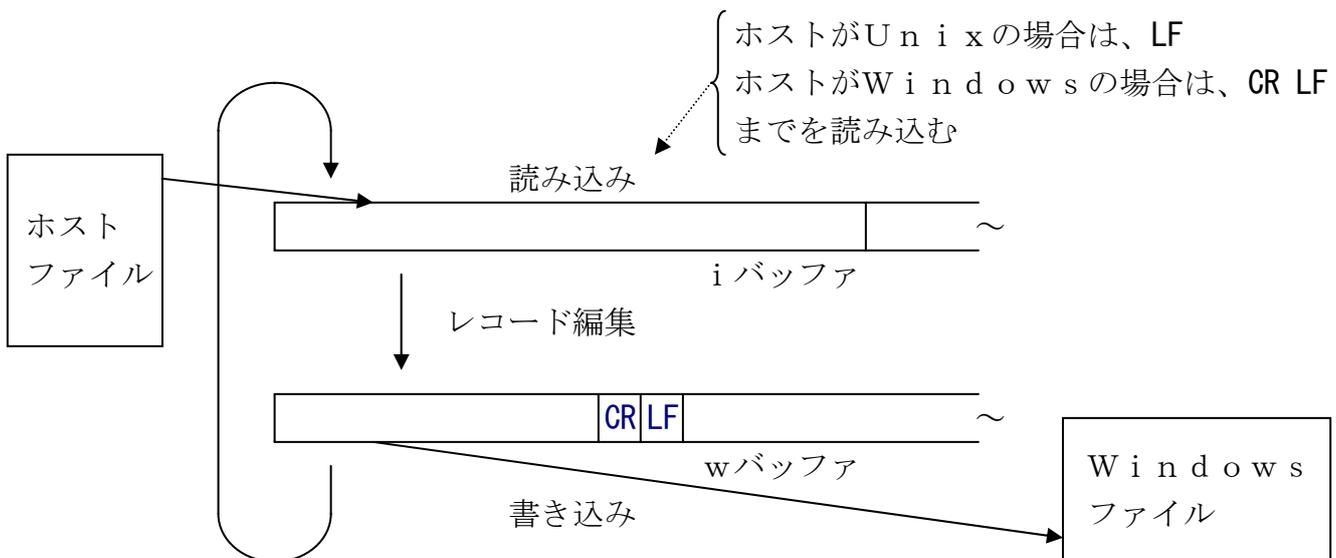
ホスト→Winテキストファイル変換は、文字データだけからなるホストファイルをWindowsのテキストファイルに変換します。おもに、ソースプログラムの変換に使用します。ホストが汎用機・オフコンの場合、ホストファイル側に漢字があるときはKI/KOがついていることが前提となり、KI/KOなしのファイルを項目別に変換するときは、ホスト→Winデータファイル変換を使います。

■解説

●テキストファイル変換の考え方

まず、ホストファイルの1レコードを保持するiバッファと、Windowsファイルの1レコードを保持するwバッファを考えます。iバッファ、wバッファは十分な大きさ（約32Kバイト）があると考えてください。

下図のようなサイクルでファイル変換が行われます。



●レコード編集

ホストファイルの1レコードがWindowsファイルの1行に対応し、レコードごとに、

ホストがUnixの場合、行末のLFを取る

ホストがWindowsの場合、行末のCR、LFを取る

ホストがUnix、Windowsの場合、

TABをタブ位置の前までのスペースに変換する（オプション指定：タブ拡張）

コード変換する（ホストが汎用機・オフコンの場合、

オプション指定：ANKのみ／ANK・漢字（KI/KO付）混在）

レコード末尾の空白類を削除する（行末圧縮）

空白類：スペース、NUL、TAB、CR、LF

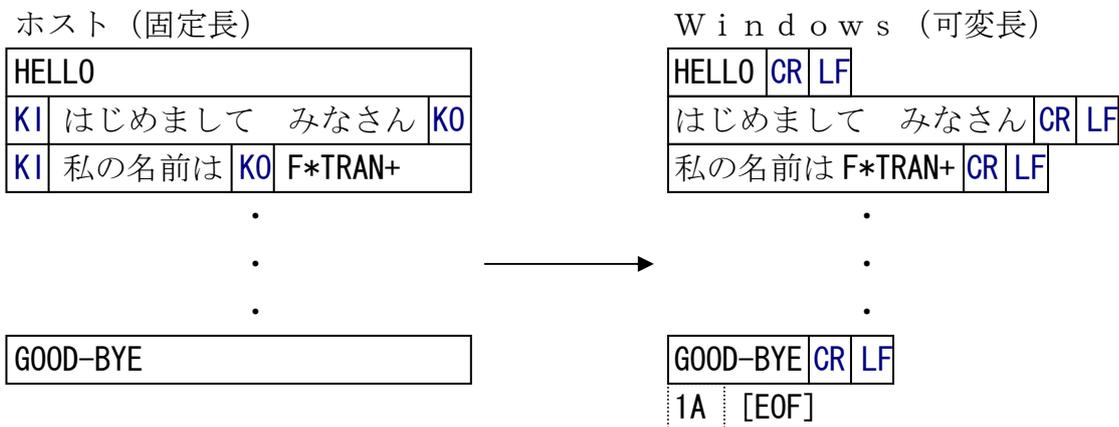
途中のNULを除去する（NULサプレス）

2個以上連続するスペースをTABに置き替える（オプション指定：タブ圧縮）

最後に、改行コード（CR/LF）を付加する

の順で加工・編集されます。

実際には、コード変換が最初に行われ、その後残りの作業が行われるので、ANK変換表を修正して制御コードや未定義コードなどを取り除くことは可能です（それらがNULに変換されるようにします）。



このように、ホスト→Winテキストファイル変換では基本的には圧縮がかかるので、レコード長はふぞろい（可変長）になります。固定長のテキストファイルが必要なときは、ホスト→Winデータファイル変換を使ってください。

ここではホストが汎用機・オフコンの例をあげていますが、ホストがUnix、Windowsの場合は、ホストファイルが可変長となり、Unixの場合はLF、Windowsの場合はCR LFまでを1レコードとしたKI/KOのないデータが対象になります。

2. 29 ホスト→Winデータファイル変換のしくみ

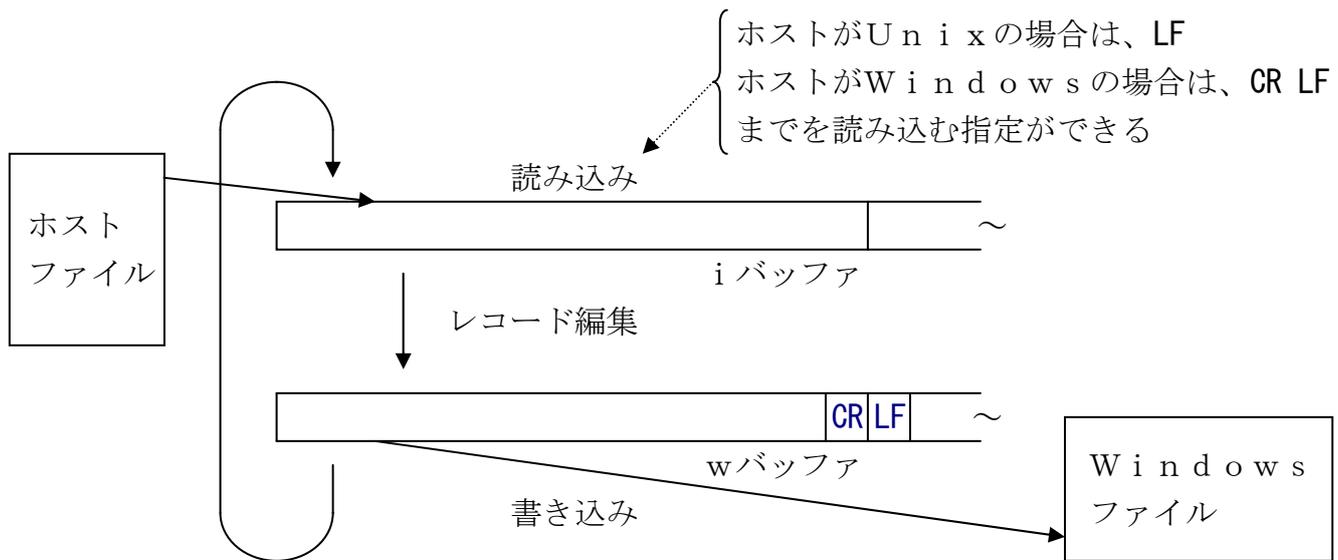
ホスト→Winデータファイル変換は、ホストのデータファイル（おもにCOBOLデータ）をWindowsのデータファイル（テキスト形式）に変換します。項目別に分けて変換できます。Windowsファイルは常に改行コード（CR/LF=0D0AH）付きのテキストファイルになります。プリント形式への変換とデリミタ形式への変換の、2つの変換方法があります。

■解説

●データファイル変換の考え方

まず、ホストファイルの1レコードを保持するiバッファと、Windowsファイルの1レコードを保持するwバッファを考えます。iバッファ、wバッファは十分な大きさ（約32Kバイト）があると考えてください。

下図のようなサイクルでファイル変換が行われます。

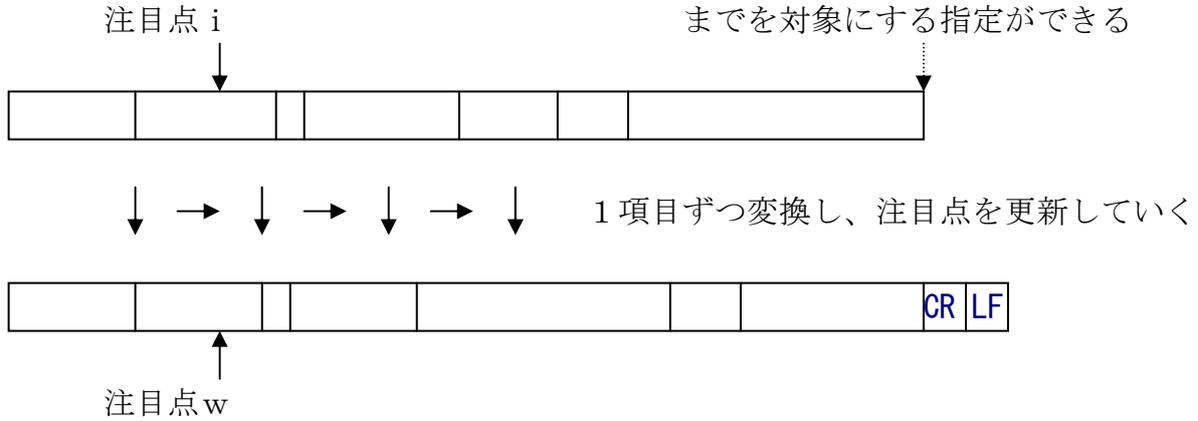


●レコード編集の考え方

F*TRAN+は、マップ設定オプションの指定を頭から順番に見ていき、それを中間コードに翻訳して「マップバッファ」と呼ぶ領域に収めます。1件ずつのレコード編集は、この中間コードを見ながら行われるので、高速なレコード編集ができるようになっています。

右図のように、iバッファの上には注目点iがあり、wバッファの上には注目点wがあると考えます。レコード編集とは、1項目ずつコード変換などを行い、指定に応じて注目点を更新していく作業なのです。

ホストがU n i xの場合は、LF
 ホストがW i n d o w sの場合は、CR LF
 までを対象にする指定ができる



ホスト→W i nデータファイル変換は、レコード編集の最初の作業として、注目点 i と注目点 w を、それぞれ i バッファ、w バッファの先頭に位置づけます。そして、w バッファをスペースでクリアします。

すべての項目の変換がおわると、w バッファの末尾（最終項目）に改行コード（CR / LF）を付加します。そして、デリミタ形式への変換で圧縮の指定があれば、さらに不要スペースの圧縮を行います。

2. 30 ホスト→Winランダムファイル変換のしくみ

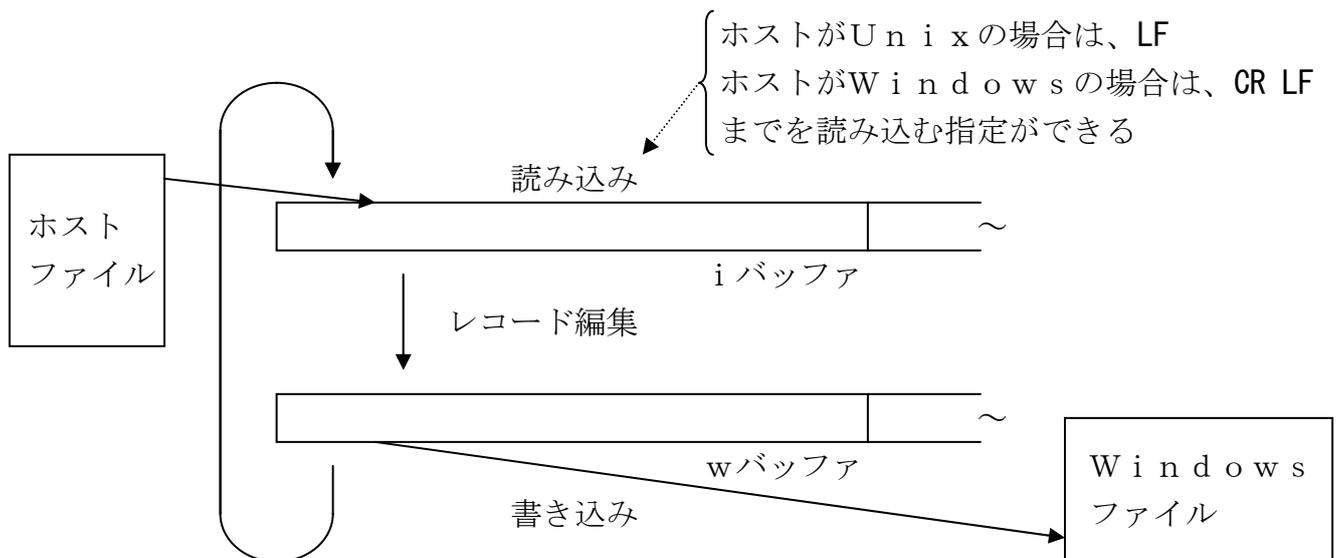
ホスト→Winランダムファイル変換は、ホストのデータファイル（おもにCOBOLのデータ）をWindowsのランダムファイル（固定長ファイル）に変換します。おもにWindows COBOLの順ファイルやBASICのランダムファイルに変換するのに使います。項目別変換ができます。作成されるWindowsファイルは固定長で、改行コードもデリミタもないファイルになります。ただし、改行コードの挿入はできます。

■解説

●ランダムファイル変換の考え方

まず、ホストファイルの1レコードを保持するiバッファと、Windowsファイルの1レコードを保持するwバッファを考えます。iバッファ、wバッファは十分な大きさ（約32Kバイト）があると考えてください。

下図のようなサイクルでファイル変換が行われます。

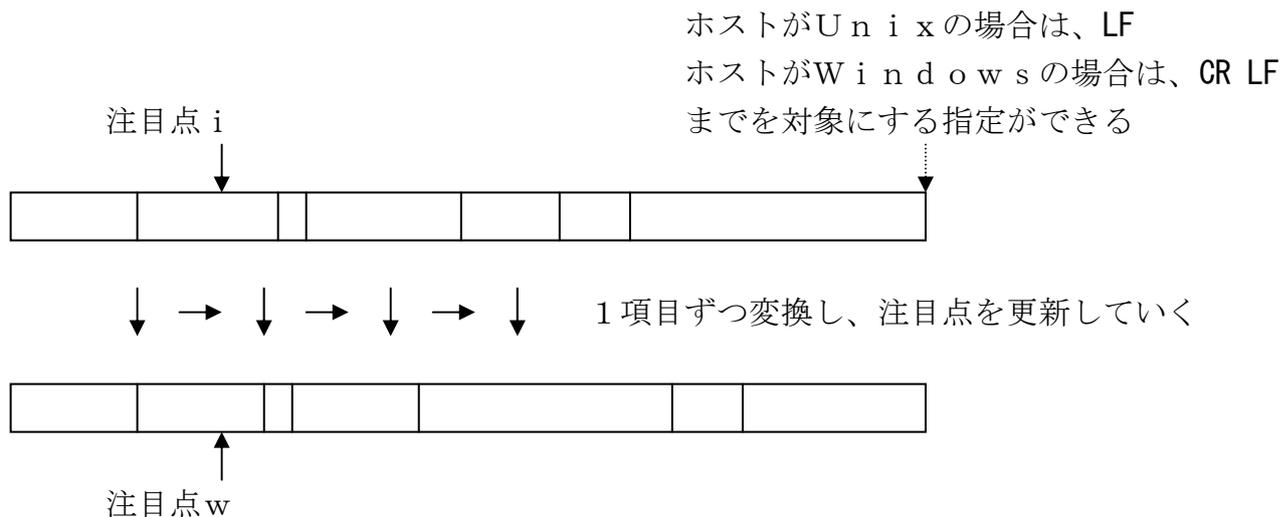


●レコード編集の考え方

F*TRAN+は、マップ設定オプションの指定を頭から順番に見ていき、それを中間コードに翻訳して「マップバッファ」と呼ぶ領域に収めます。

1件ずつのレコード編集は、この中間コードを見ながら行われるので、高速なレコード編集ができるようになっています。

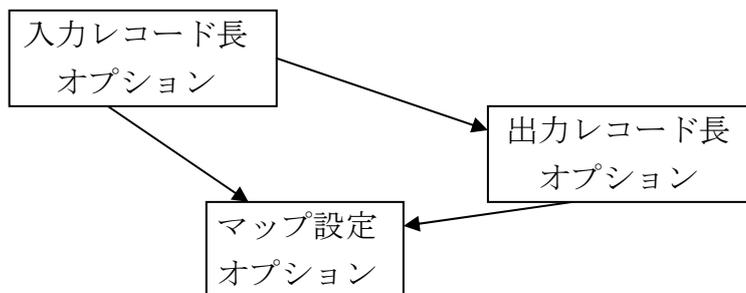
右図のように、iバッファの上には注目点iがあり、wバッファの上には注目点wがあると考えます。レコード編集とは、1項目ずつコード変換などを行い、指定に応じて注目点を更新していく作業なのです。



ホスト→W i n ランダムファイル変換は、レコード編集の最初に注目点 i と注目点 w をそれぞれ i バッファ、w バッファの先頭に位置づけます。そして、w バッファを N U L でクリアします。すべての項目の変換がおわると、それ以上の加工はせずに、W i n d o w s ファイルに出力します。

●レコード長の参照関係

ホスト→W i n ランダムファイル変換は、レコード長の参照関係が少し複雑です。それを、下図に示します。



2. 31 Win→ホストテキストファイル変換のしくみ

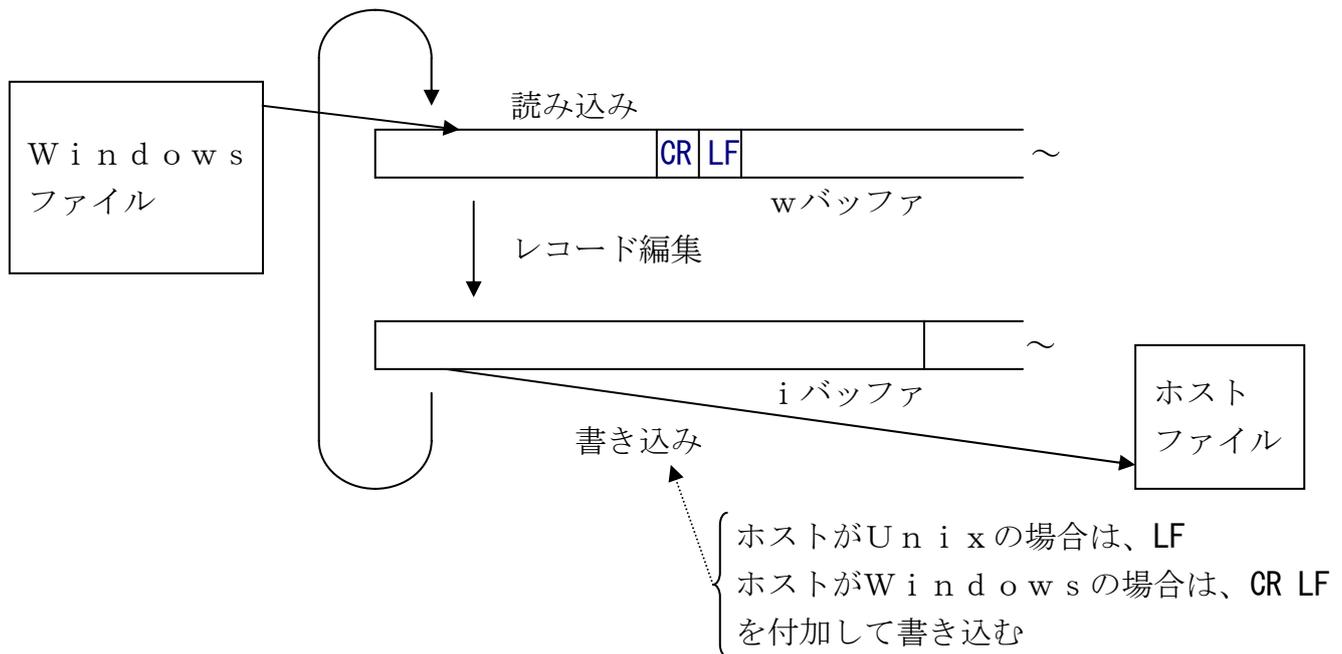
Win→ホストテキストファイル変換は、ソースプログラムのようなWindowsファイルをホストファイルに変換します。Windowsファイルはテキストファイル(改行コードつき)でなければなりません。改行コードでレコードのおわりを検出するので、可変長でかまいません。ホストが汎用機・オフコンの場合は、ホストファイルは固定長になり、ホストがUnix、Windowsの場合は、ホストファイルは可変長になります。

■解説

●テキストファイル変換の考え方

まず、Windowsファイルの1レコードを保持するwバッファと、ホストファイルの1レコードを保持するiバッファを考えます。wバッファ、iバッファは十分な大きさ(約32Kバイト)があると考えてください。

下図のようなサイクルでファイル変換が行われます。



●レコード編集

Windowsファイルの1行(CR/LFまで)がホストファイルの1レコードに対応し、レコードごとに、

行末のCR、LFを取る

TABをタブ位置の前までのスペースに変換する（オプション指定：タブ拡張）

コード変換する（ホストが汎用機・オフコンの場合、

オプション指定：ANKのみ／ANK・漢字（KI／KO付）混在）

ホストがUnix、Windowsの場合、コード末尾の空白類を削除する（行末圧縮）

空白類：スペース、NUL、TAB、CR、LF

NULを除去する（NULサプレス）

ホストがUnix、Windowsの場合、

2個以上連続するスペースをTABに置き替える（オプション指定：タブ圧縮）

ホストが汎用機・オフコンの場合、レコード末尾にスペースを詰める（行末パディング）

ホストがUnixの場合、レコード末尾にLFを付ける

ホストがWindowsの場合、レコード末尾にCR、LFを付ける

の順で加工・編集されます。

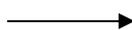
実際には、コード変換が最初に行われ、その後残りの作業が行われるので、ANK変換表を修正して制御文字や未定義文字などを取り除くことは可能です(それらがNULに変換されるようにします)。

Windows（可変長）

HELLO	CR	LF
はじめまして	みなさん	CR LF
私の名前は	F*TRAN+	CR LF
.		
.		
.		
GOOD BYE	CR	LF
1A	[EOF]	

ホスト（固定長）

HELLO		
KI	はじめまして	みなさん KO
KI	私の名前は	KO F*TRAN+
.		
.		
.		
GOOD BYE		



このように、Win→ホストテキストファイル変換では基本的には可変長のテキストファイルを扱いますが、単純な固定長テキストファイルの変換に使うこともできます（Win→ホストデータファイル変換を使うほうが適切ですが）。

ここではホストが汎用機・オフコンの例をあげていますが、ホストがUnix、Windowsの場合は、ホストファイルが可変長となり、Unixの場合はLF、Windowsの場合はCR LFを付加したKI／KOのないデータになります。

2. 32 Win→ホストデータファイル変換のしくみ

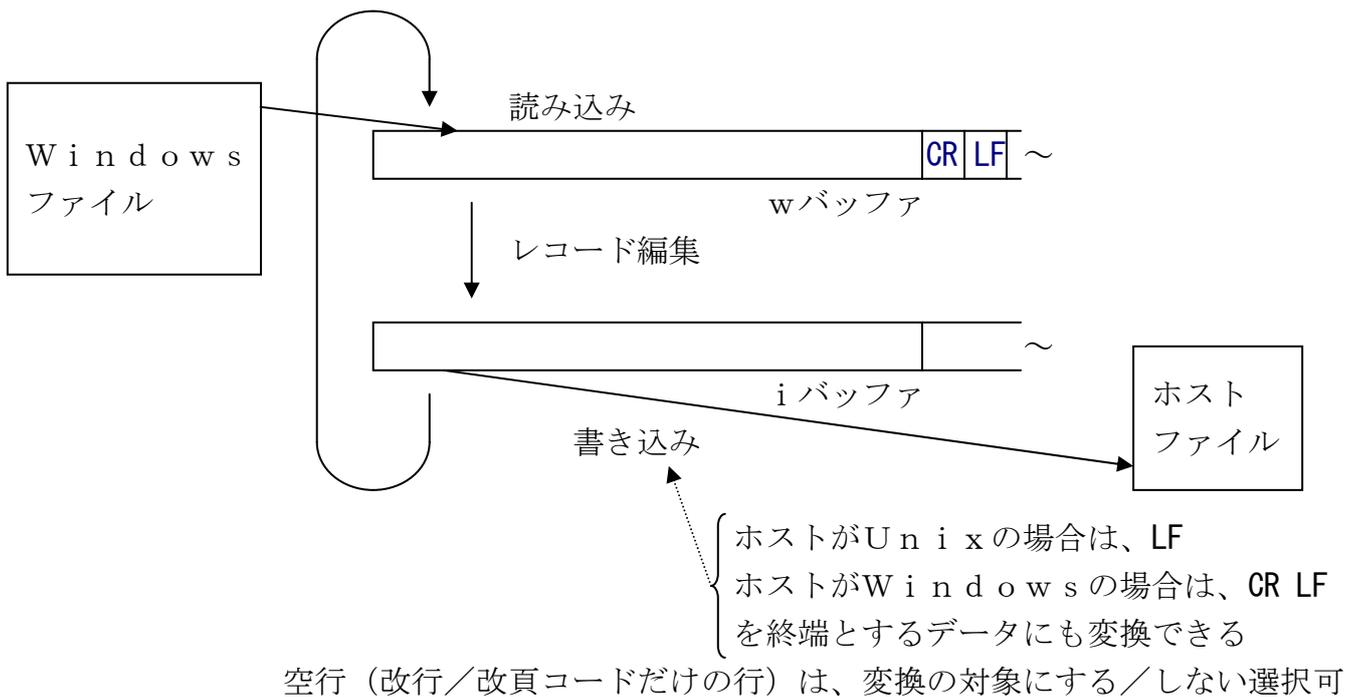
Win→ホストデータファイル変換は、Windowsのデータファイルをホストファイルに変換します。Windowsファイルはプリント形式のファイルかデリミタ形式のファイルでなければいけません。ANKデータだけの単純な変換、項目別の変換などが行えます。なお、ホストが汎用機・オフコンの場合、ホストファイルは必ず固定長・固定欄になります。

■解説

●データファイル変換の考え方

まず、Windowsファイルの1レコードを保持するwバッファと、ホストファイルの1レコードを保持するiバッファを考えます。wバッファ、iバッファは十分な大きさ（約32Kバイト）があると考えてください。

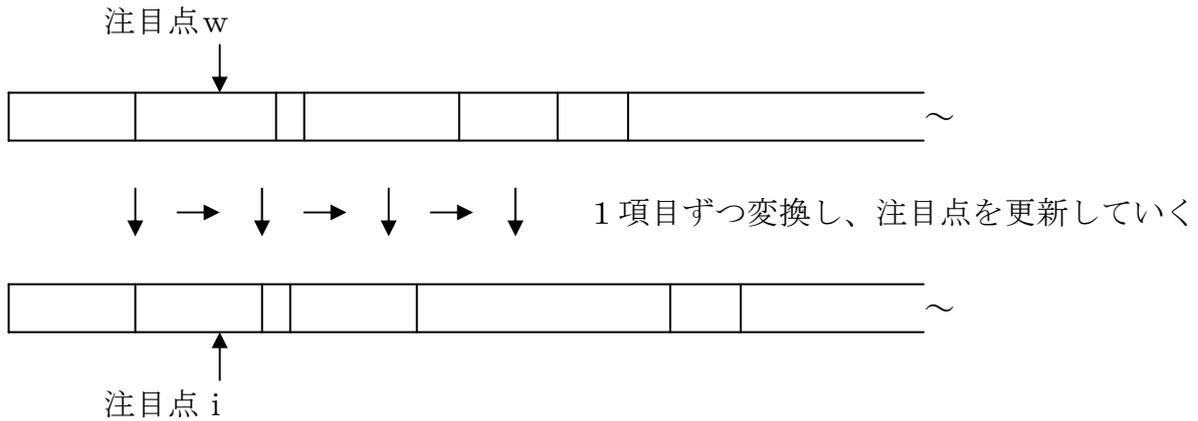
下図のようなサイクルでファイル変換が行われます。



●レコード編集の考え方

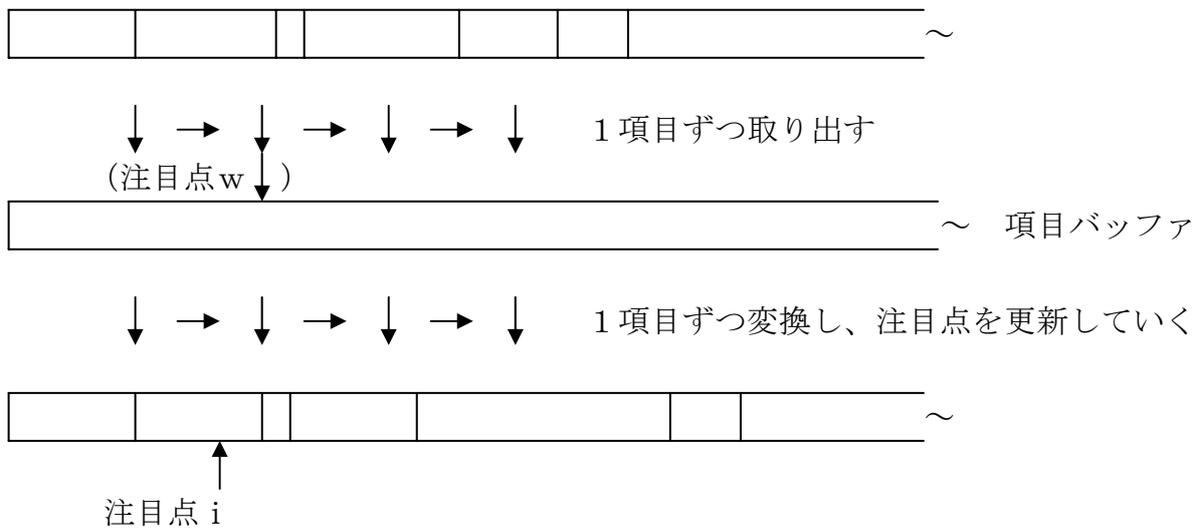
F*TRAN+は、マップ設定オプションの指定を頭から順番に見ていき、それを中間コードに翻訳して「マップバッファ」と呼ぶ領域に収めます。1件ずつのレコード編集は、この中間コードを見ながら行われるので、高速なレコード編集ができるようになっています。

まず、プリント形式からの変換を考えます。右図のように、wバッファの上には注目点wがあり、iバッファの上には注目点iがあると考えます。レコード編集とは、1項目ずつコード変換などを行い、指定に応じて注目点を更新していく作業なのです。



Win→ホストデータファイル変換は、レコード編集の最初の作業として、注目点wと注目点iを、それぞれwバッファ、iバッファの先頭に位置づけます。そして、iバッファをスペースでクリアします。すべての項目の変換がおわると、iバッファの内容をホストファイルに書き込みます。

デリミタ形式からの変換の場合は、wバッファとiバッファの間に「項目バッファ」があると考えるとわかりやすくなります。



各項目の変換のたびに項目を項目バッファに取り込み、うしろはスペースでクリアします。そして、それに対して変換をかけます。注目点wは、あまり重要な働きはしません。一方、注目点iのほうは、1項目ずつ変換するたびに更新していきます。

2. 33 Win→ホストランダムファイル変換のしくみ

Win→ホストランダムファイル変換は、Windowsのランダムファイル（固定長ファイル）をホストファイルに変換します。固定長レコードの単純な変換、固定長・固定欄形式のファイルの項目別変換などが行えます。バイナリ変換もできます。

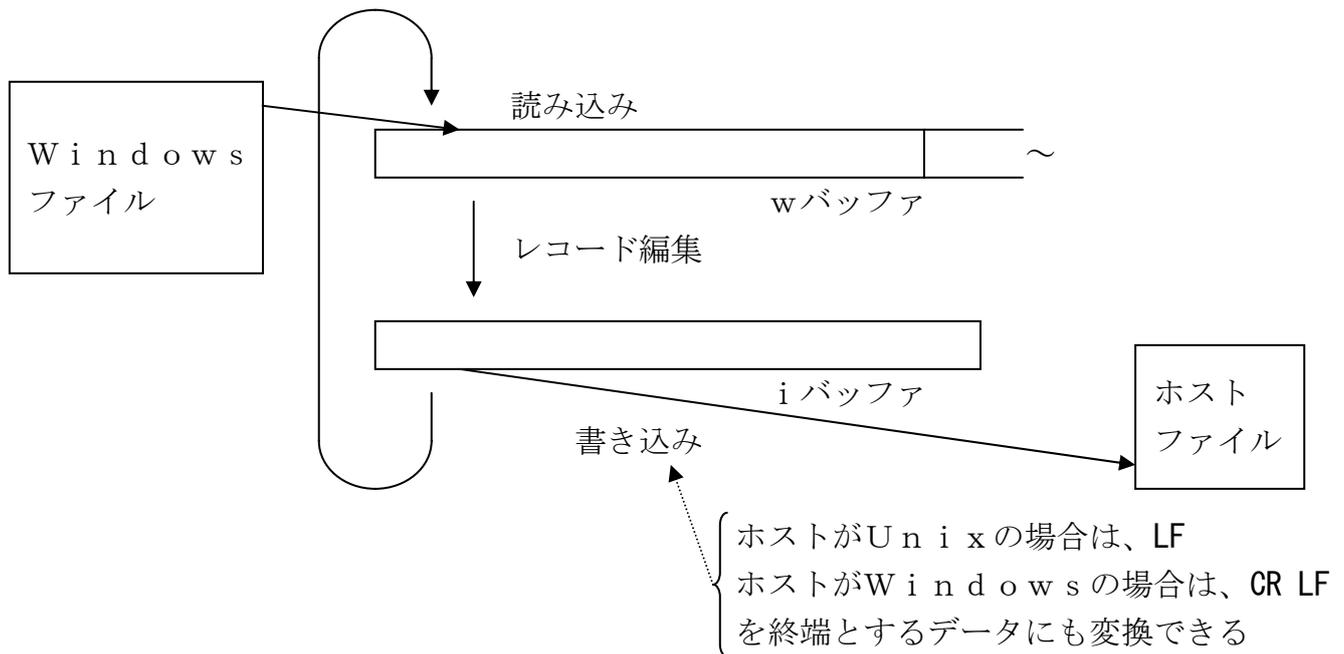
固定長・固定欄なら、改行コードのついていないテキストファイルも扱えますが、その場合にはWin→ホストデータファイル変換のプリント形式からの変換機能を使ったほうがずっと簡単です。

■解説

●ランダムファイル変換の考え方

まず、Windowsファイルの1レコードを保持するwバッファと、ホストファイルの1レコードを保持するiバッファを考えます。wバッファ、iバッファは十分な大きさ（約32Kバイト）があると考えてください。

下図のようなサイクルでファイル変換が行われます。

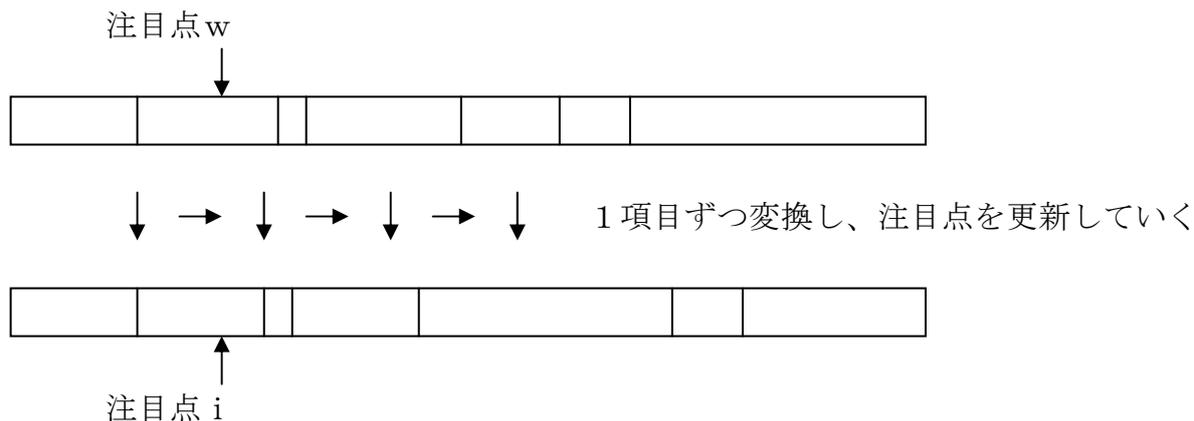


●レコード編集の考え方

F*TRAN+は、マップ設定オプションの指定を頭から順番に見ていき、それを中間コードに翻訳して「マップバッファ」と呼ぶ領域に収めます。

1件ずつのレコード編集は、この中間コードを見ながら行われるので、高速なレコード編集ができるようになっています。

右図のように、wバッファの上には注目点wがあり、iバッファの上には注目点iがあると考えます。レコード編集とは、1項目ずつコード変換などを行い、指定に応じて注目点を更新していく作業なのです。

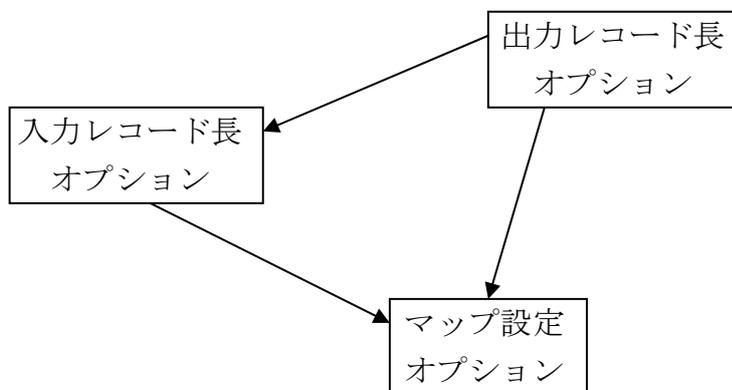


Win→ホストランダムファイル変換は、レコード編集の最初の作業として、注目点wと注目点iを、それぞれwバッファ、iバッファの先頭に位置づけます。そして、iバッファをNULでクリアします。

すべての項目を変換すると、それ以上は加工しないでホストファイルに出力します。

●レコード長の参照関係

Win→ホストランダムファイル変換は、レコード長の参照関係が少し複雑です。それを、以下に示します。



第3章

操作の基礎

3. 1 ファイルの指定

●指定形式

ファイルはつぎの形式で指定します。

```
[d:] [パス名指定] [基本ファイル名 [. 拡張子]]
```

d : はドライブ名

●ドライブ名 (d :)

ドライブ名 (d :) には、

A : ~ Z : 実ドライブ名を指定します。

また、F*TRAN+内でのみ通用するドライブ名として、

@ : カレントドライブのカレントディレクトリを表す

? : インストールディレクトリを表す

も指定できます。

ドライブ名は省略可能です。省略するとカレントドライブを指定したものとみなされます。

●パス名指定 (¥ディレクトリ名¥サブディレクトリ名¥・・・)

パス名指定 (¥やディレクトリ名を使う指定) ができます。指定したディレクトリ配下のファイルを扱うことができます。

●基本ファイル名と拡張子

大部分のコマンドでは、入力側のファイル名を指定する場合、基本ファイル名と拡張子にワイルドカード文字（*、?）が使用できます。

出力側のファイル名を指定する場合、基本ファイル名と拡張子にはふつうの基本ファイル名や拡張子以外に、*も指定できます。*は「入力側の基本ファイル名や拡張子を引き継げ」という意味です。ふつうは、

`d : *. dat` のように、拡張子を指定します。

こうすると、指定ドライブに

元の基本ファイル名. 拡張子 というファイルができます。

なお、できる限り適切な拡張子をつけるように心がけてください。

●拡張子の省略値

ファイル名の指定ができるところでも、拡張子の省略値が、

`. CCT` **コード変換表ファイルの、省略時の拡張子**
`. P` **パラメータファイルの、省略時の拡張子**

のように決まっているものもあります。

3.2 式の指定

F*TRAN+では、オプションデータなどのパラメータを10進数で指定できるところで、ほとんどの場合、10進数の代わりに値を「式」で指定することができます。計算を省いたり、レコード長のようなその都度変わるデータに対しても同じ指定ですむようにしたり、相対的な指定を可能にしたりするためです。

●式とは

式といっても、複雑な数式のようなものを指定できるわけではありません。10進数を、四則演算子とカッコで組み合わせられるだけの単純なものです。一部のオプションではさらに特殊変数を使うこともできます。

●四則演算子

四則演算子はずぎの5つです。

+	加算
-	減算
* または x (小文字の x)	乗算
/ または %	除算
¥ ¥	剰余 (割算の余り)

演算子間の優先順位はありません。*を+-より先に計算したりしないということです。カッコを使って計算の順番を明示してください。カッコは何重にも入れ子にできます。

●特殊変数

特殊変数にはづぎの6つがあります。

\$	最大値を表す	(ふつうはレコード長を意味する)
. (ピリオド)	現在値を表す	(ふつうはレコードの現在の桁位置を意味する)
*	残りを表す	(ふつうはレコードの残りの長さを意味する)
~SysPhase	フェーズ	(ふつうはマルチレコードの指定時に使用する)
~SysRecNum	レコード番号	(ふつうはマルチレコードの指定時に使用する)
~SysReturn	リターン値	(ふつうはマルチレコードの指定時に使用する)
~SysBreak	ブレーク値	(ふつうはマルチレコードの指定時に使用する)
~SysQuit	クイット値	(ふつうはマルチレコードの指定時に使用する)

式の例を示します。つぎに示すのはいずれも正しい式です。

7	.	\$
(1 5)	. - 1 5	\$ x 4
4 + (6 + 8)	. + 2	\$ + 2
2 5 6 * 2 6	*	\$ % 3
1 0 2 4 * 4	* - 2	\$ x ((\$ % 8 0) + 1)
* * 3	* / / 3	* ¥ ¥ 3
~SysRecNum	~SysRecNum¥¥3	

◆注意 ---- 式に空白を入れてはならない

演算子やカッコなどの前後に空白を入れてはいけません。空白はパラメータ類の区切りを意味するからです。

3. 3 ピクチャの指定

●ピクチャとは

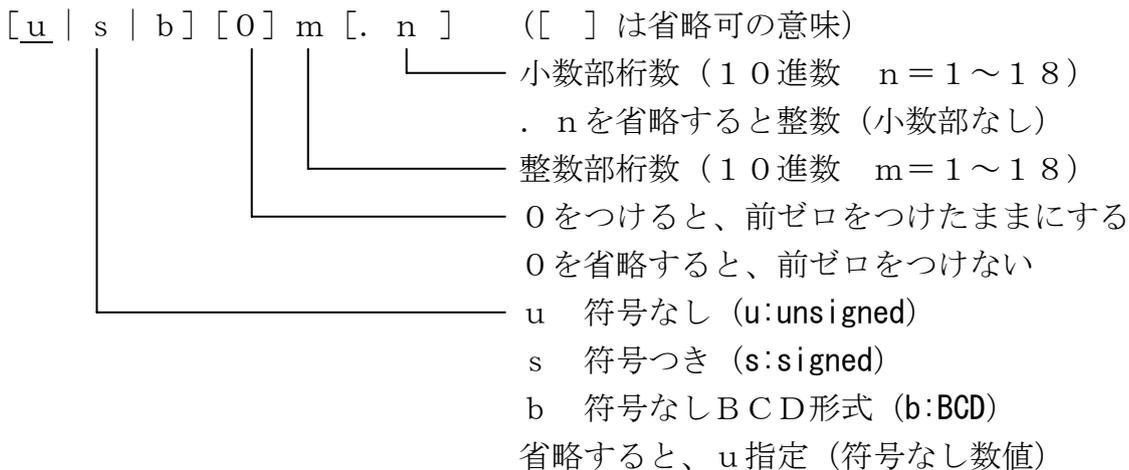
ピクチャとは、COBOLのゾーン/パック形式や、BCD形式の数値項目を変換するとき、

- 符号の有無
- 数値の桁数
- 前ゼロの有無
- 小数部の有無と桁数

を指定するためのものです。これらは、データ自体には記録されていないので、外からこれらの情報を与える必要があります。

●ピクチャの指定形式

ピクチャはつぎの形式で指定します。



●COBOLをまねた

F*TRAN+のピクチャは、COBOLのピクチャ指定をまねた上で、大幅に簡略化したものです。たとえば、

- 1 2 . 3

という数字があつて、5バイトのゾーン形式の項目に記録してあるとします。COBOLのピクチャなら

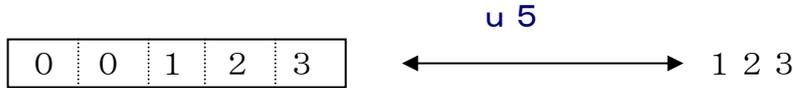
P I C S 9 (4) V 9 (1)

のように指定します。

F*TRAN+のピクチャではこれを、 **s 4 . 1** と指定します。

ピクチャの指定例を示します。以下の図の左側がゾーン形式の項目、右側が文字形式数値、そして矢印の上がピクチャです。

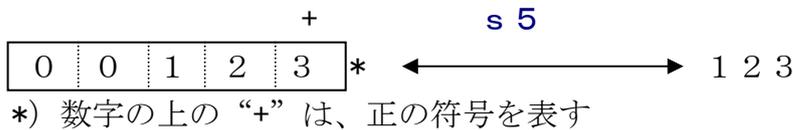
例1) 符号なし整数



例2) 符号付きの、負の整数



例3) 符号付きの、正の整数



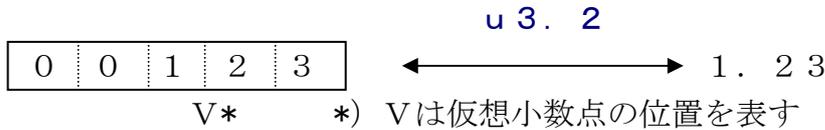
例4) 符号なし整数、前ゼロをつけたままにする



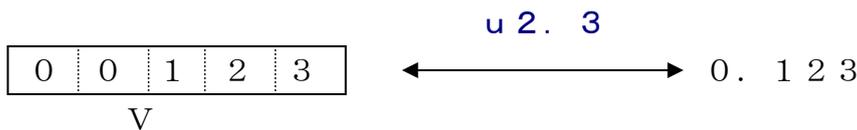
例5) 符号付き整数、前ゼロをつけたままにする



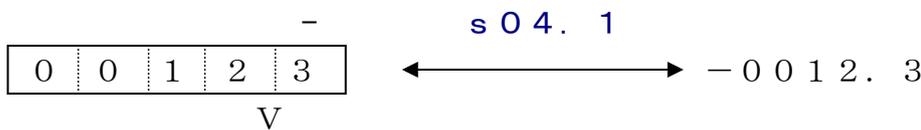
例6) 小数部（小数点）がある／その1



例7) 小数部（小数点）がある／その2



例8) 符号も小数部（小数点）もあり、前ゼロもつけたままにする



3. 4 2進ピクチャの指定

● 2進ピクチャとは

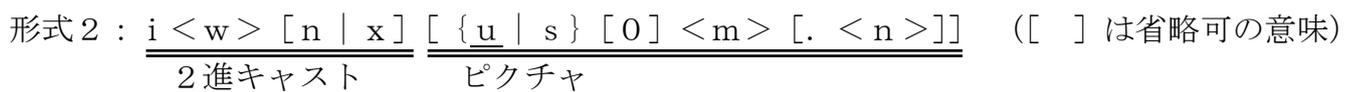
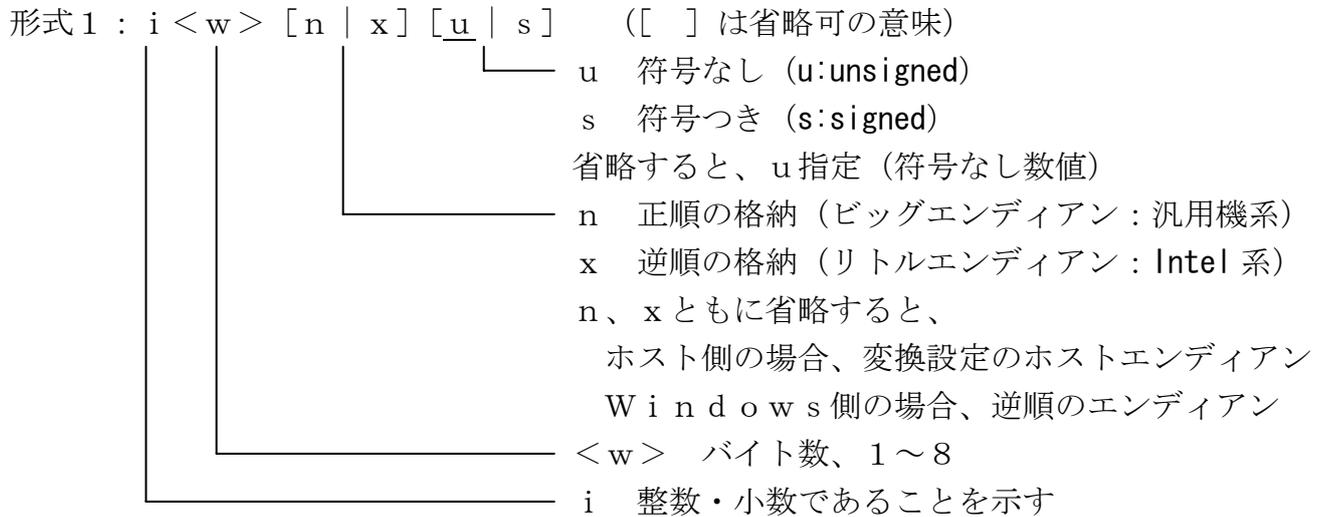
2進ピクチャとは、COBOL、Visual Basic、C/C++などの2進数値項目を変換するとき、

- バイト数
- 格納順
- 符号の有無
- 数値の桁数
- 前ゼロの有無
- 小数部の有無と桁数

を指定するためのものです。これらは、データ自体には記録されていないので、外からこれらの情報を与える必要があります。

● 2進ピクチャの指定形式

2進ピクチャはつぎのどちらかの形式で指定します。



- 0 前ゼロ付加
- <m> 整数部桁数、1～18
- <n> 小数部桁数、1～18、省略すると0
- <m>+<n>が1～18になること
- <w>に応じて、<m>+<n>の省略値が定まる

●各言語の型と2進ピクチャの対応

言語	型	2進ピクチャ
COBOL (例)	B I N A R Y	i 1 n u ~ i 8 n u
	SつきB I N A R Y	i 1 n s ~ i 8 n s
	C O M P	i 1 n u ~ i 8 n u
	SつきC O M P	i 1 n s ~ i 8 n s
	C O M P - 4	i 1 n u ~ i 8 n u
	SつきC O M P - 4	i 1 n s ~ i 8 n s
	C O M P - 5	i 1 x u ~ i 8 x u
	SつきC O M P - 5	i 1 x s ~ i 8 x s
V i s u a l B a s i c	B y t e	i 1 u
	I n t e g e r	i 2 s
	L o n g	i 4 s
	C u r r e n c y	i 8 s 1 4 . 4
C / C ++	s i g n e d c h a r	i 1 s
	u n s i g n e d c h a r	i 1 u
	s i g n e d s h o r t	i 2 s
	u n s i g n e d s h o r t	i 2 u
	s i g n e d i n t	i 4 s
	u n s i g n e d i n t	i 4 u
	s i g n e d l o n g	i 4 s
	u n s i g n e d l o n g	i 4 u
	s i g n e d ___ i n t 6 4	i 8 s
u n s i g n e d ___ i n t 6 4	i 8 u	

3. 5 日付データの指定

●日付マスク

F*TRAN+で使用できる日付データの編集指定はつぎのとおりです。

日付マスク	データ例	日付マスク	データ例
yyyy-mm-dd *1	1998/12/31 1998-12-31 1998.12.31	yyyy-mm *1	1998/12 1998-12 1998.12
yy-mm-dd	98/12/31 98-12-31 98.12.31	yy-mm	98/12 98-12 98.12
nyy-mm-dd	H10/12/31 H10-12-31 H10.12.31	nyy-mm	H10/12 H10-12 H10.12
yyyymmdd	19981231	yyyymm	199812
yyymmdd	981231	yyymm	9812
mm-dd-yyyy *1	12/31/1998 12-31-1998 12.31.1998	mm-yyyy *1	12/1998 12-1998 12.1998
mm-dd-yy	12/31/98 12-31-98 12.31.98	mm-yy	12/98 12-98 12.98
mmddyyyy	12311998	mmyyyy	121998
mmddyy	123198	mmyy	1298
dd-mm-yyyy *1	31/12/1998 31-12-1998 31.12.1998	yyyy *1	1998
dd-mm-yy	31/12/98 31-12-98 31.12.98	yy	98
ddmmyyyy	31121998	nyy	H10
ddmmyy	311298	gyymmdd	4101231
		gyymm	41012
		gyy	410

n = 年号
M (明治) 1868-1911
T (大正) 1912-1925
S (昭和) 1926-1988
H (平成) 1989-

g = 元号
1 (明治) 1868-1911
2 (大正) 1912-1925
3 (昭和) 1926-1988
4 (平成) 1989-

◆注意 ---- 出力時、和暦の年号/元号の最終年は、次年号/元号の元年(01)になる

実際には、日付マスク分の長さが編集対象になります。たとえば“y y y y m m d d”と指定すれば、8文字分のデータの編集を行います。

入力側に“y y - m m - d d”のような日付区切りのある指定をした場合は、8文字の内容が“98 12, 31”であっても、“98-12-31”と同等のデータとして扱います。つまり、数字（0～9）以外の文字を日付区切り記号とみなします。

●日付マスクのオプション

日付マスク（y y y y m m d dなど）には以下のオプションが指定可能です。

日付マスク	$\left[\begin{array}{l} -D i s p \\ -A l p h a \end{array} \right]$
-------	--

入力、または出力コードがUCS-2の時、-D i s p 指定では1桁2バイト単位で処理します。それ以外の場合は1桁1バイト単位で処理します。これが省略値です。

-A l p h a 指定ではコードに関係なく1桁1バイト（ゾーン形式）で処理します。

●ウインドウ方式とシフト方式

日付データの年の2桁（y y）と4桁（y y y y）の変換を行う場合、F*TRAN+ではウインドウ方式とシフト方式の概念を採用しています。

ウインドウ方式とは、19xx年（基準年）から100年として扱う方式です。ウインドウ方式で“30”と指定すれば、実際のデータはつぎのようになります。

	1930年		2000年		2030年
データ	30	99	00	29	

シフト方式とは、西暦からnn引いた値の下2桁のデータを扱う方式です。一般には、nn=25（昭和通年方式）、nn=88（平成通年方式）、nn=28（暦一巡方式）などがあります。シフト方式で“30”と指定すれば、実際のデータはつぎのようになります。

	1930年		2000年		2030年
データ	00	69	70	99	

入力側に*1の日付マスク指定（日付区切りがある4桁の年指定）をし、実際の日付データの年が2桁以下であった場合は、無条件にウインドウ方式による拡張を行います。

●日付区切り記号

日付データを出力する際に指定できる日付区切り記号はつぎのとおりです。

<u>日付区切り記号</u>	<u>デ ー タ 例</u>
／（スラッシュ）	1 9 9 8 / 1 2 / 3 1
－（ハイフン）	1 9 9 8 - 1 2 - 3 1
．（ピリオド）	1 9 9 8 . 1 2 . 3 1

●マップ設定オプション指定

日付データを変換する手順はつぎのとおりです。

- ①年設定（日付データ2桁の年の扱い、ウインドウ方式またはシフト方式の指定）
- ②日付区切り設定（日付データ出力時の日付区切り記号の指定）
- ③日付項目変換（日付データ変換時の日付マスク指定）

①、②を省略すると、年設定はウインドウ方式で1930年より（入力、出力とも）、“／”で日付区切りとなります。

第4章



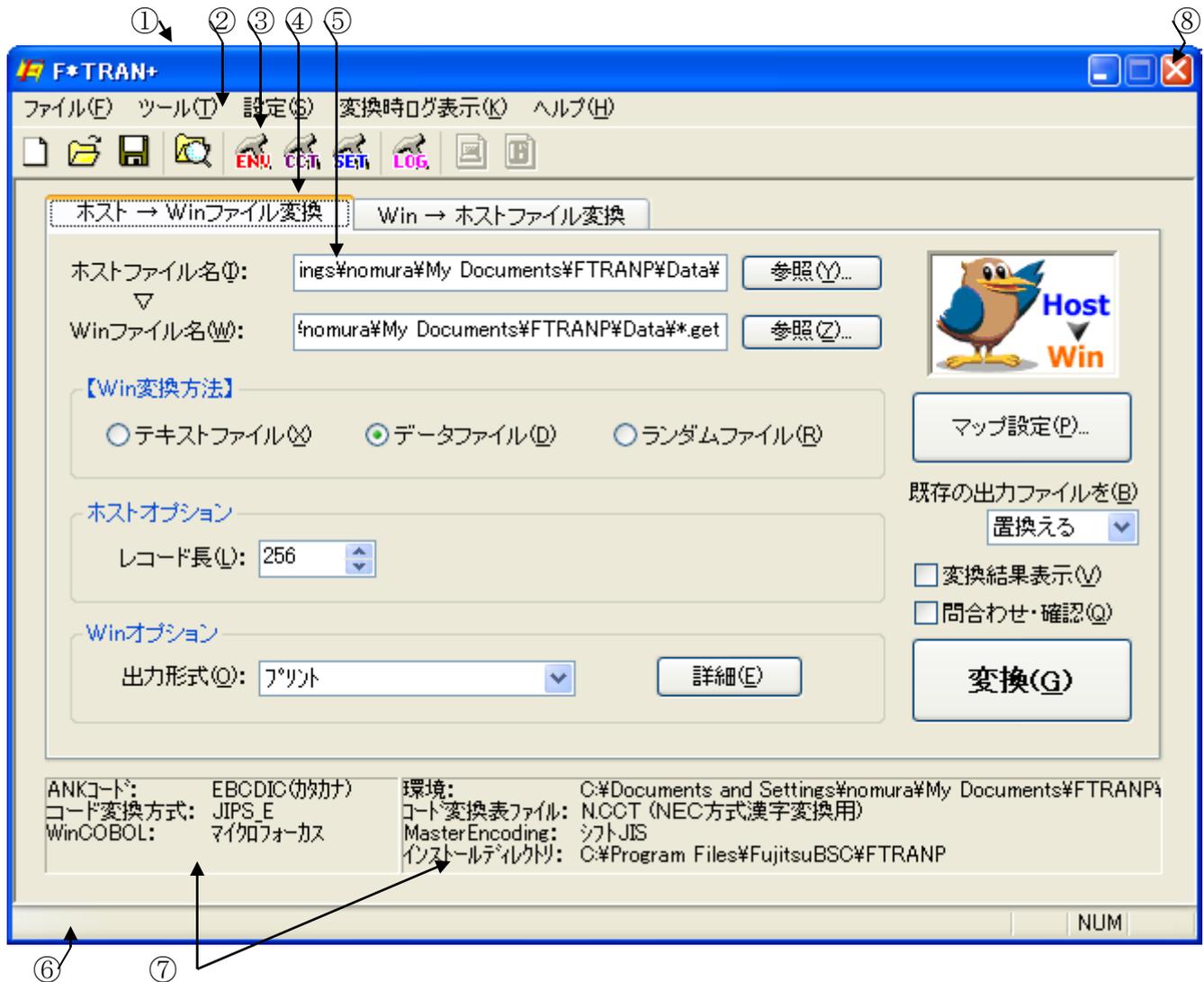
操作

4.1 F*TRAN+ 起動と終了

■ F*TRAN+の起動

F*TRAN+をウインドウ設定方式で使う場合の起動方法について説明します。

- 通常、スタートメニューより、プログラム (P) → F*TRAN+をクリックします。
起動を行うと、つぎのメインウインドウが開きます。<ホストが汎用機・オフコンの例>



- ①タイトルバー ----- ファイルメニューで開いたファイル名が表示されます。
- ②メニューバー ----- 目的のメニューを選択して、機能を実行します。
- ③ツールバー ----- ②の中で、よく使われる機能がボタン化されています。
- ④タブコントロール ----- 2通りのメイン処理部の切り替えを行います。
- ⑤メイン処理部 ----- ④の切り替えによって、メイン処理部の内容が変わります。
- ⑥ステータスバー ----- 操作のガイダンスやメッセージが表示されます。
- ⑦設定表示バー ----- おもにコード変換に関する現在の設定が表示されています。

“設定表示バー”は、メインウインドウおよび実行ウインドウ（変換実行時のウインドウ）の下方にあり、コード変換に関する現在の設定等をいつでも確認できるようになっています。内容はつぎのとおりです。

ANKコード : EBCDIC (かか) <ホストが汎用機・オフコンの場合>
 (「変換設定」により設定されたホストのANKコード)

漢字変換方式 : JEF
 (「変換設定」により設定された漢字変換方式)

WinCOBOL : 富士通
 (「変換設定」により設定されたWindows COBOLのベンダ名)

ホストCOBOL : 富士通 <ホストがUnix、Windowsの場合>
 (「変換設定」により設定されたホストCOBOLのベンダ名)

環境 : <ユーザデータフォルダ>%EnvGroup%Env%
 (「環境設定」により設定された環境フォルダ名)

コード変換表ファイル : F-TRAN.CCT (標準設定)
 (「ホスト選択」により選択されたコード変換表ファイル名とコメント)

MasterEncoding : シフトJIS
 (「環境設定」により設定された基準になるエンコーディング)

インストールディレクトリ : C:%Program Files%FujitsuBSC\FTRANP
 (F*TRAN+のインストールディレクトリ)

◆参考 ---- <ユーザデータフォルダ>とは

<ユーザデータフォルダ>は、ログオンユーザの「Documents」にある“FTRANP”フォルダがデフォルトで指定されています。

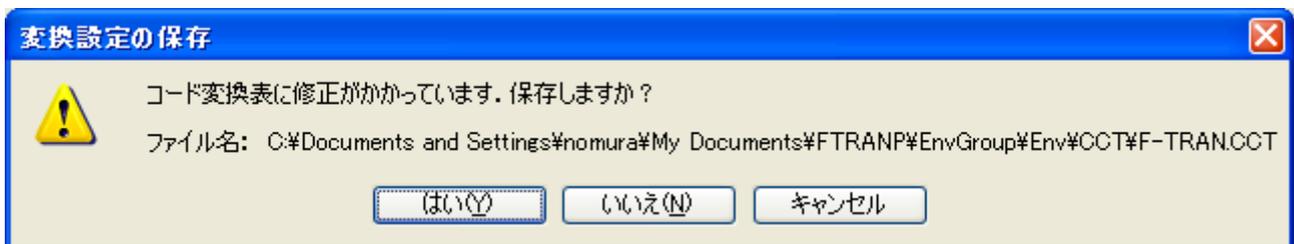
■ F*TRAN+の終了

F*TRAN+をウインドウ設定方式で使う場合の終了の方法について説明します。

- ふつう、②メニューバーのファイル (F) → 終了 (X) を実行します。
または⑧の  をクリックしても終了します。

●コード変換表の保存

コード変換表に何らかの修正がかかった状態で、F*TRAN+を終了しようとする、修正のかかったコード変換表を保存するか否かをたずねるメッセージウインドウが開きます。



はい (Y) をクリックすると、修正のかかったコード変換表を保存します。

いいえ (N) をクリックすると、修正のかかったコード変換表を保存しません。

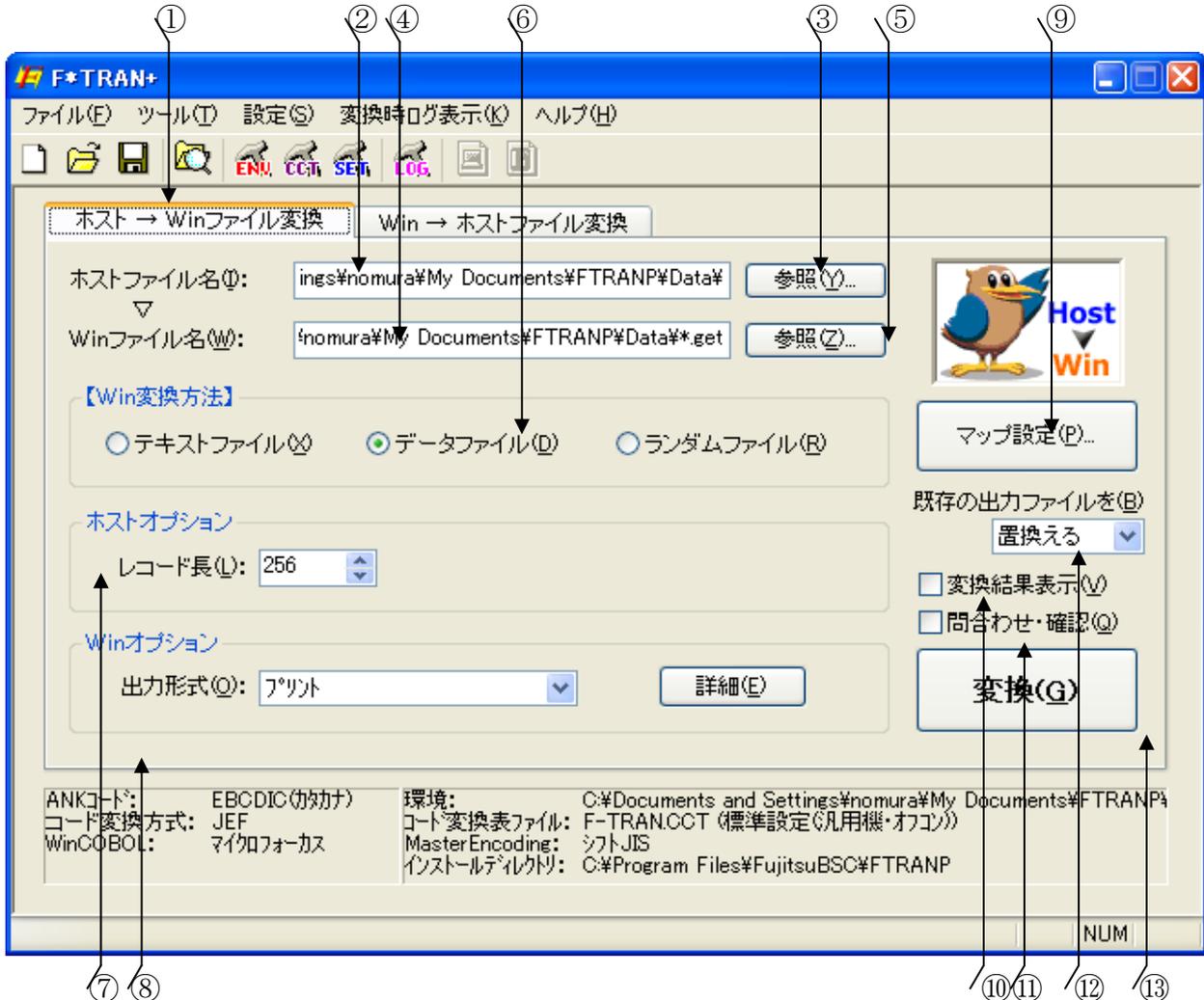
キャンセルをクリックすると、F*TRAN+の終了をキャンセルします。

動作環境設定ファイル“FP.SYS”にて、無条件にコード変換表ファイルを保存する指定も可能です。

4.2 ホスト→Winファイル変換のファイル指定と共通オプション

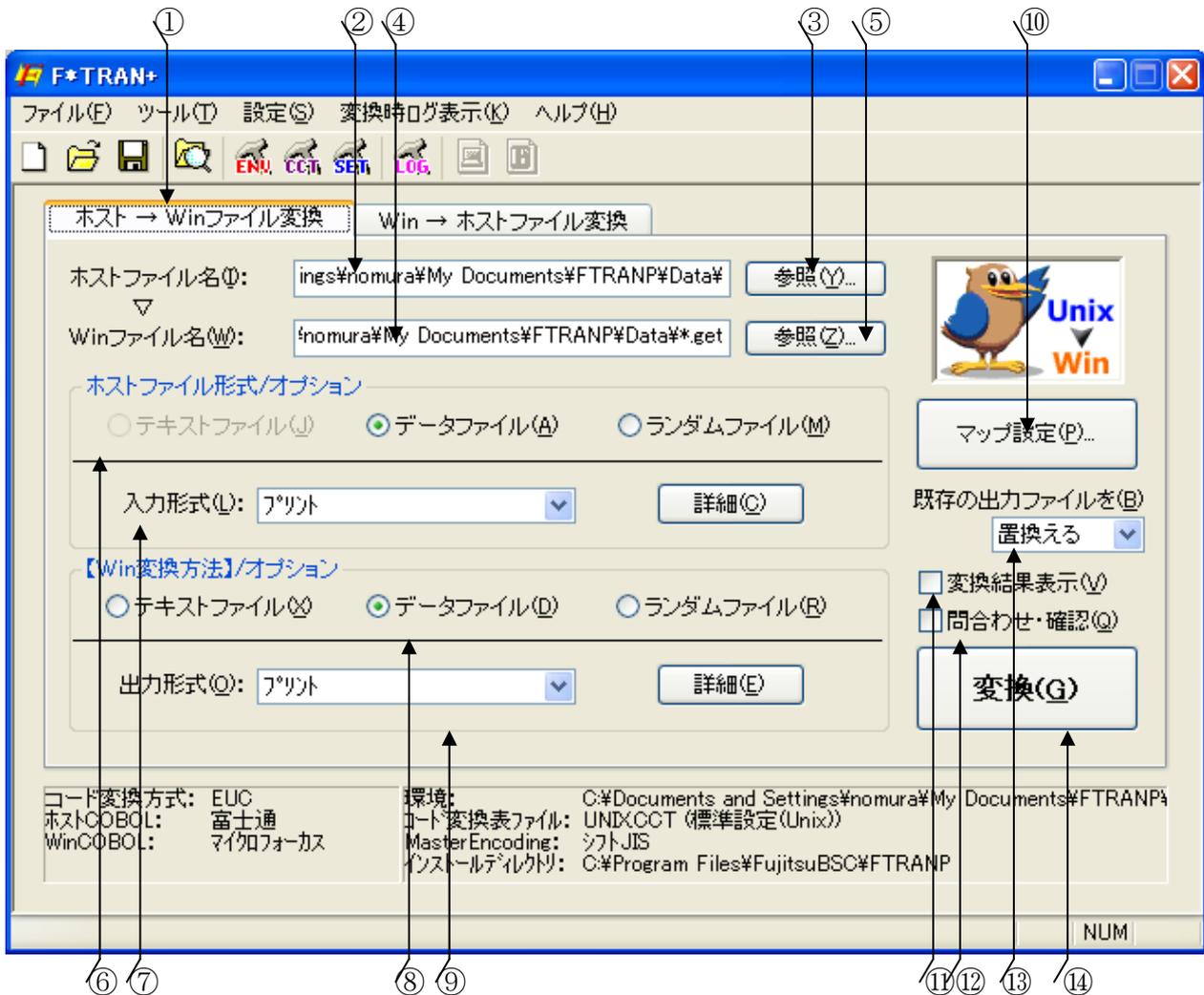
ここでは、ホスト→Winファイル変換の共通事項を説明します。

●ホストが汎用機・オフコンの場合



- ①ホスト→Winのタブをクリックし、メイン処理部をホスト→Winファイル変換にします。
- ②変換元のホストファイル名を指定します。
- ③変換元のホストファイル名を指定する参照ウインドウを開くボタンです。
- ④変換先のWindowsファイル名を指定します。
- ⑤変換先のWindowsファイル名を指定する参照ウインドウを開くボタンです。
- ⑥変換方法を選択するラジオボタンです。
- ⑦ホストファイルのオプション設定です。⑥の選択によって、オプションの内容が変わります。
- ⑧Winファイルのオプション設定です。⑥の選択によって、オプションの内容が変わります。
- ⑨マップ設定（レコードレイアウト設定）をするオプションボタンです。
- ⑩変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑪変換実行時の問い合わせ・確認ウインドウ表示の選択をするチェックボックスです。
- ⑫変換先ファイルが既に存在した場合の動作を指定するコントロールです。
- ⑬変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

●ホストがUnix、Windowsの場合



- ①ホスト→Winのタブをクリックし、メイン処理部をホスト→Winファイル変換にします。
- ②変換元のホストファイル名を指定します。
- ③変換元のホストファイル名を指定する参照ウインドウを開くボタンです。
- ④変換先のWindowsファイル名を指定します。
- ⑤変換先のWindowsファイル名を指定する参照ウインドウを開くボタンです。
- ⑥ホストファイル（入力ファイル）の形式を選択するラジオボタンです。
- ⑦ホストファイルのオプション設定です。
- ⑧Winファイルの変換方法を選択するラジオボタンです。変換の基準となる項目です。
- ⑨Winファイルのオプション設定です。
- ⑩マップ設定（レコードレイアウト設定）をするオプションボタンです。
- ⑪変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑫変換実行時の問い合わせ・確認ウインドウ表示の選択をするチェックボックスです。
- ⑬変換先ファイルが既に存在した場合の動作を指定するコントロールです。
- ⑭変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

■ホストファイル指定

[d:] [パス名指定] 基本ファイル名 [. 拡張子]

d : はドライブ名

入力側のホストファイルを指定します。

ドライブ名は、A : ~ Z : 、@ : 、? : のどれかで指定します。ドライブ名を指定すると、そのドライブを検索します。

ドライブ名は省略可能です。ドライブ名を省略すると、カレントドライブを検索します。

パス名指定 (≠ディレクトリ名≠サブディレクトリ名≠・・・) ができます。指定したディレクトリ配下のファイルを検索します。パス名指定を省略すると、カレントディレクトリを検索します。

基本ファイル名と拡張子には、ワイルドカード文字 (* と ?) を使うことができます。ワイルドカード文字を使うと、一致するファイルをすべて検索し変換の対象にします。

まとめると、あるディレクトリのファイルをすべて変換したいなら、

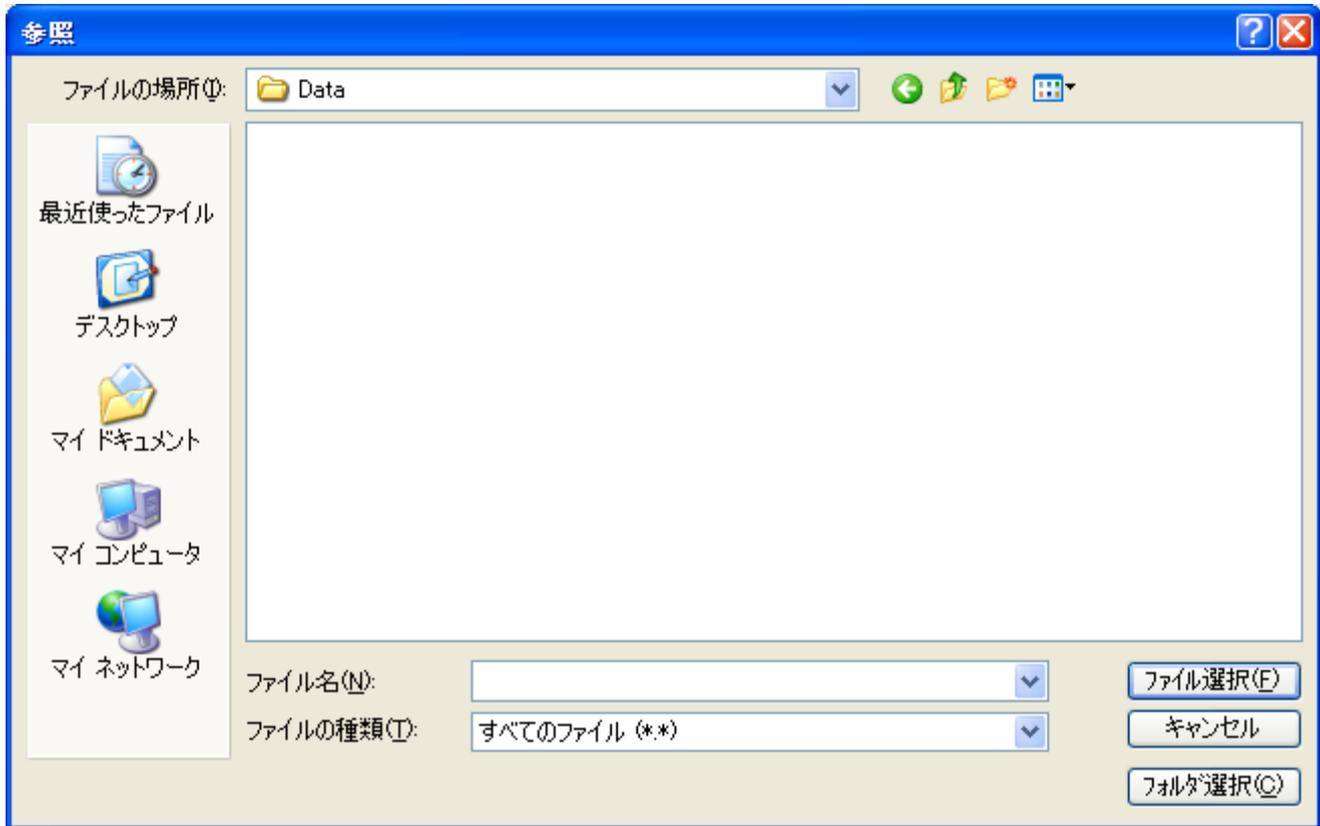
C : * . * のように指定し、

拡張子が . DAT のファイルをすべて変換したいなら、

C : * . DAT のような指定になります。

●ホストファイル参照ウインドウ

ホストファイルの参照ボタンをクリックすると、つぎのウインドウが開きます。



変換元のファイルを直接指定する場合は、ファイル名を選択して、ファイル選択 (F) ボタンをクリックします。

変換元の場所 (どこのフォルダのファイルを変換の対象にするか) を指定する場合は、フォルダ選択 (C) ボタンをクリックします。ファイル名にフォルダまでのフルパスが入力され、確定します。

■Windowsファイル指定

[d:] [パス名指定] 基本ファイル名 [. 拡張子]

d : はドライブ名

出力側のWindowsファイルを指定します。

ドライブ名はA : ~Z : 、@ : 、? : のどれかで指定します。ドライブ名を指定すると、そのドライブにファイルを作ります。

ドライブ名は省略可能です。ドライブ名を省略すると、カレントドライブにファイルを作ります。

パス名指定 (¥ディレクトリ名¥サブディレクトリ名¥・・・) ができます。指定したディレクトリ配下にファイルを作ります。パス名指定を省略すると、カレントディレクトリにファイルを作ります。

基本ファイル名の部分には、ふつうの基本ファイル名、または*を指定します。*を指定するとホストファイルの基本ファイル名がWindowsファイルの基本ファイル名になります。

拡張子を省略すると、拡張子なしのファイルになります。しかし、拡張子をつけたほうがファイルの管理が容易になるので、なるべく適当な拡張子を指定してください。拡張子に*を指定するとホストファイルの拡張子がWindowsファイルの拡張子になります。

以上がディスクファイルの指定方法です。まとめると、基本ファイル名を引き継ぐときは

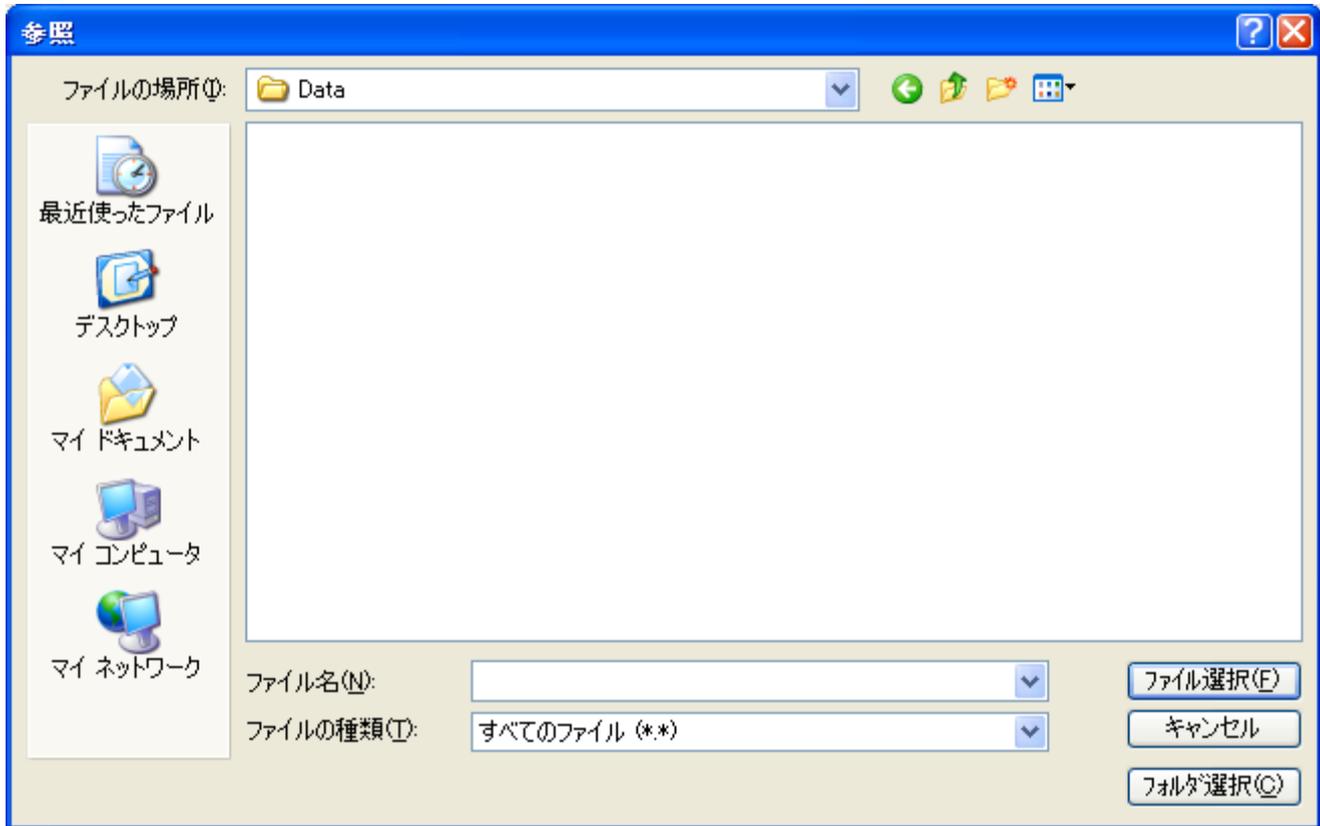
C : *. DAT のような指定になり、

ファイル名をつけ替えるときは

C : NEWNAME. DAT のような指定になります。

●Windowsファイル参照ウインドウ

Windowsファイルの参照ボタンをクリックすると、つぎのウインドウが開きます。



変換先のファイルを直接指定する場合は、ファイル名を選択して、ファイル選択 (F) ボタンをクリックします。

変換先の場所 (どこのフォルダに変換後のファイルを作成するか) を指定する場合は、フォルダ選択 (C) ボタンをクリックします。ファイル名にフォルダまでのフルパスが入力され、確定します。

■その他の指定

●変換結果表示指定

変換後に、変換先のWindowsファイルの中身を表示ウインドウに出力するかどうかを指定します。出力する場合は、変換結果表示のチェックボックスをONにします。

●問合わせ・確認指定

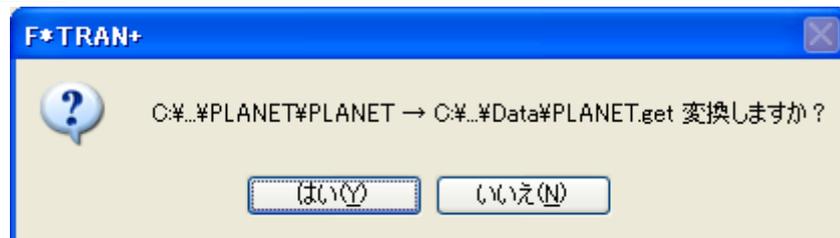
1ファイルごとに処理を問合わせるか否かを指定します。
問合わせ・確認のチェックボックスを、

ONにすると、変換するか否かを問合わせる指定

OFFにすると、ファイル名の確認なしで自動変換する指定 となります。

問合わせ・確認指定がONであれば、ファイル名を確認しながら変換できます。
F*TRAN+は、1ファイルごとに変換を実行するか問わせてきます。
つぎのどれかで応答してください。この機能は、比較的小さいファイルが多数あって、そのうちいくつかを選んで変換したいときなどに、便利です。

1ファイルの変換



はい (Y)

表示中のファイルを変換する

いいえ (N)

表示中のファイルは変換しない

2ファイル以上の変換



はい (Y)

表示中のファイルを変換する

すべて変換 (A)

全ファイル変換に切り替え、以降のファイルをすべて変換する

いいえ (N)

表示中のファイルは変換しない

キャンセル

これ以降の変換処理を中断する

問合わせ・確認指定がOFFであれば、ファイル名の確認なしで、自動的に指定のファイルをすべて変換します。こちらがデフォルトです。

●既存の出力ファイルの置き換え指定

変換先ファイルと同名のファイルが同一フォルダ上に存在した場合の動作を以下の3通りから指定します。

◇「置換える」指定時（省略値）

出力先ファイルと同名のファイルが同一フォルダ上に存在した場合は、自動で置き換えます。

◇「置換えない」指定時

出力先ファイルと同名のファイルが同一フォルダ上に存在した場合は以下のメッセージをログに出力し、プログラムは正常終了します。（異常終了ではありません）

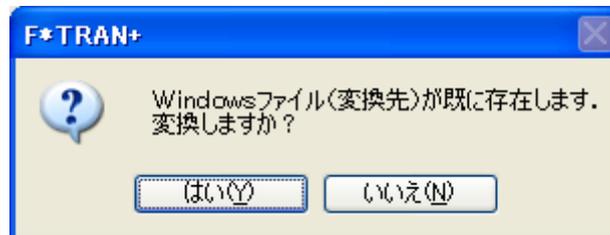
- ・ホスト→Win変換時のメッセージ

「Windowsファイルが既に存在する為、変換できません。」

◇「問合わせる」指定時

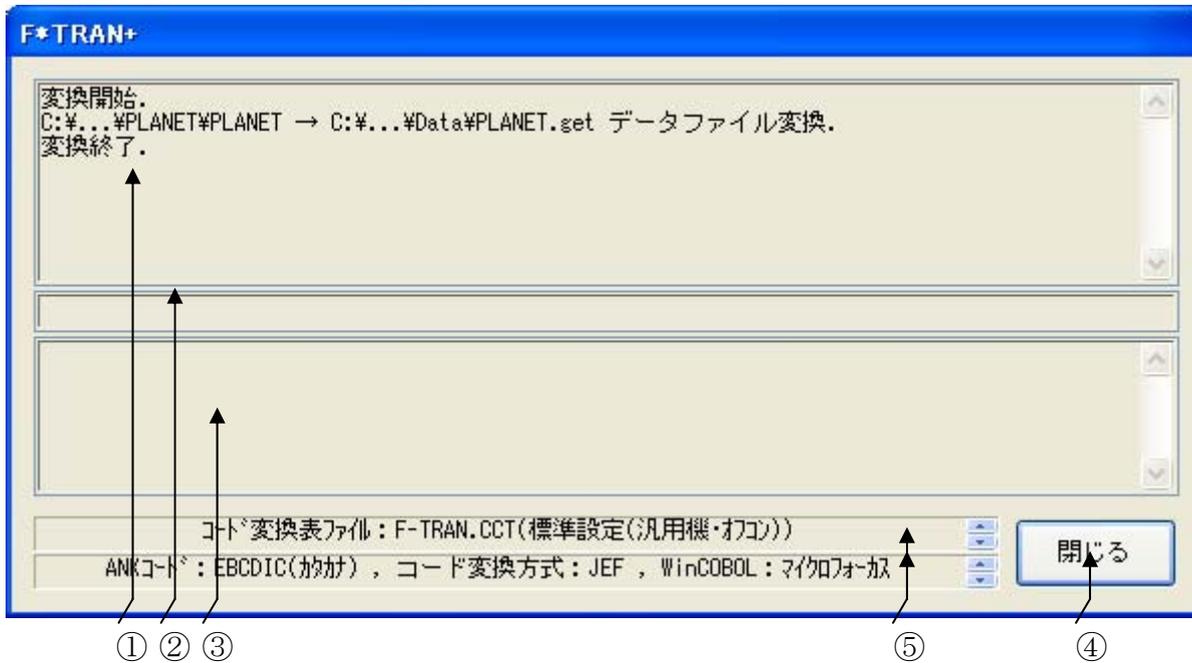
出力先ファイルと同名のファイルが同一フォルダ上に存在した場合は、以下の問い合わせダイアログを表示して、ユーザの指示を待ちます。

【ホスト→Win変換時の問い合わせ画面】



■変換時の実行ウィンドウ

変換ボタンをクリックすると、ホスト→Winファイル変換が始まり、つぎの実行ウィンドウが開きます。



①変換中のメッセージを表示するメッセージフィールドです。

変換が始まると、“**変換開始.**”というメッセージ表示につづいて、“**ホストファイル名 → Windowsファイル名 ~ファイル変換.**”が表示され、変換が正常終了すると、“**変換終了.**”と表示されます。

②変換中の進行状況（進行度%出力）を表示するフィールドです。

③変換中のエラーメッセージを表示するフィールドです。

④変換中は「中止 (Q)」と表示されています。

変換中にこのボタンをクリックすると、変換を途中で中止します。

変換が終了したら、その変換結果を確認して、“閉じる”ボタンをクリックします。

⑤おもにコード変換に関する現在の設定が表示されています。

■注意事項

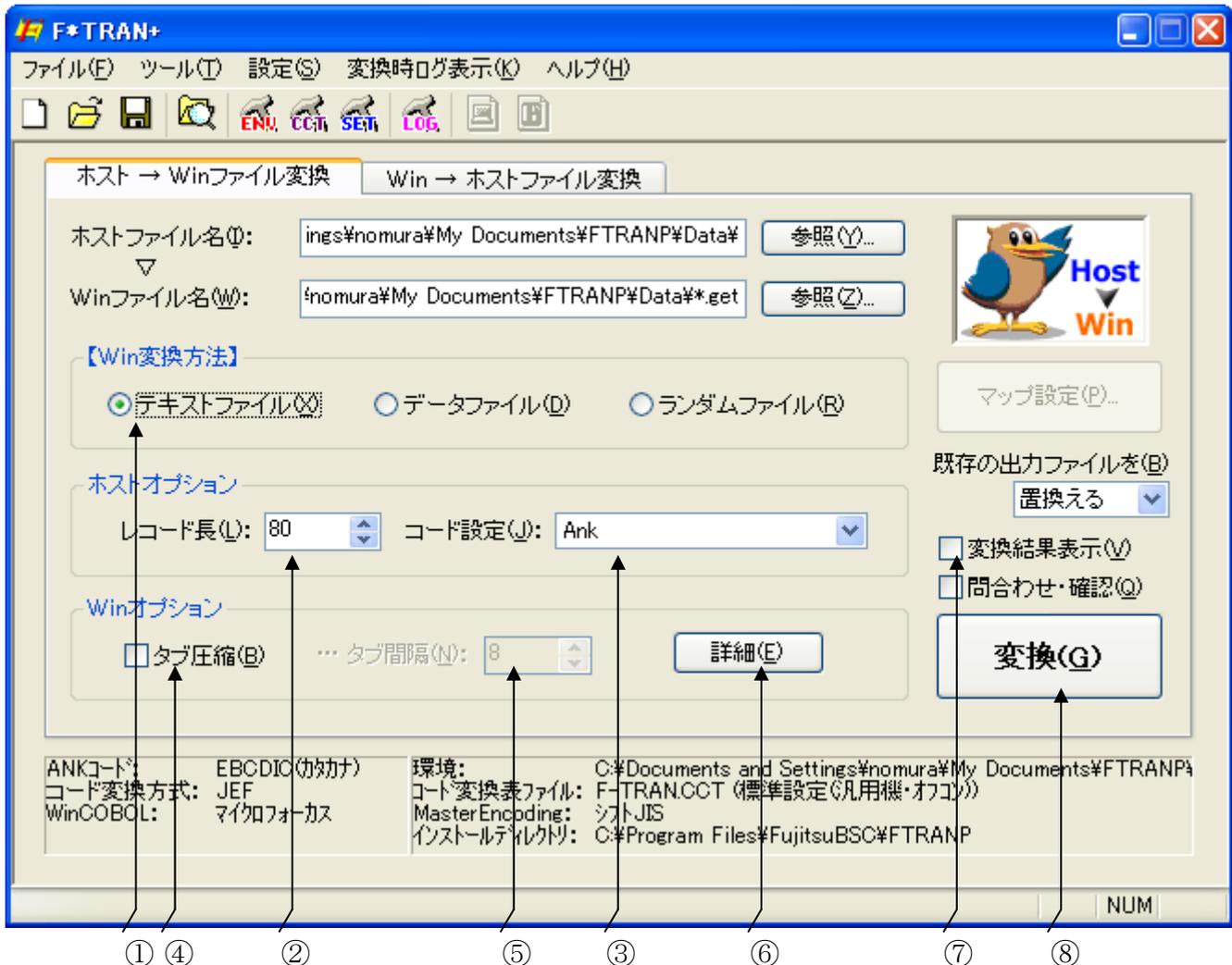
同名ファイルは置換する（デフォルト）

すでに同じ名前のWindowsファイルがある場合、デフォルトでは自動的に元のファイルを削除し、新たに変換したファイルで置き替えます。この動作は画面上からの指定により変更可能です。詳細は「●既存の出力ファイルの置き換え指定」の説明を参照してください。

4.3 ホスト→Winテキストファイル変換

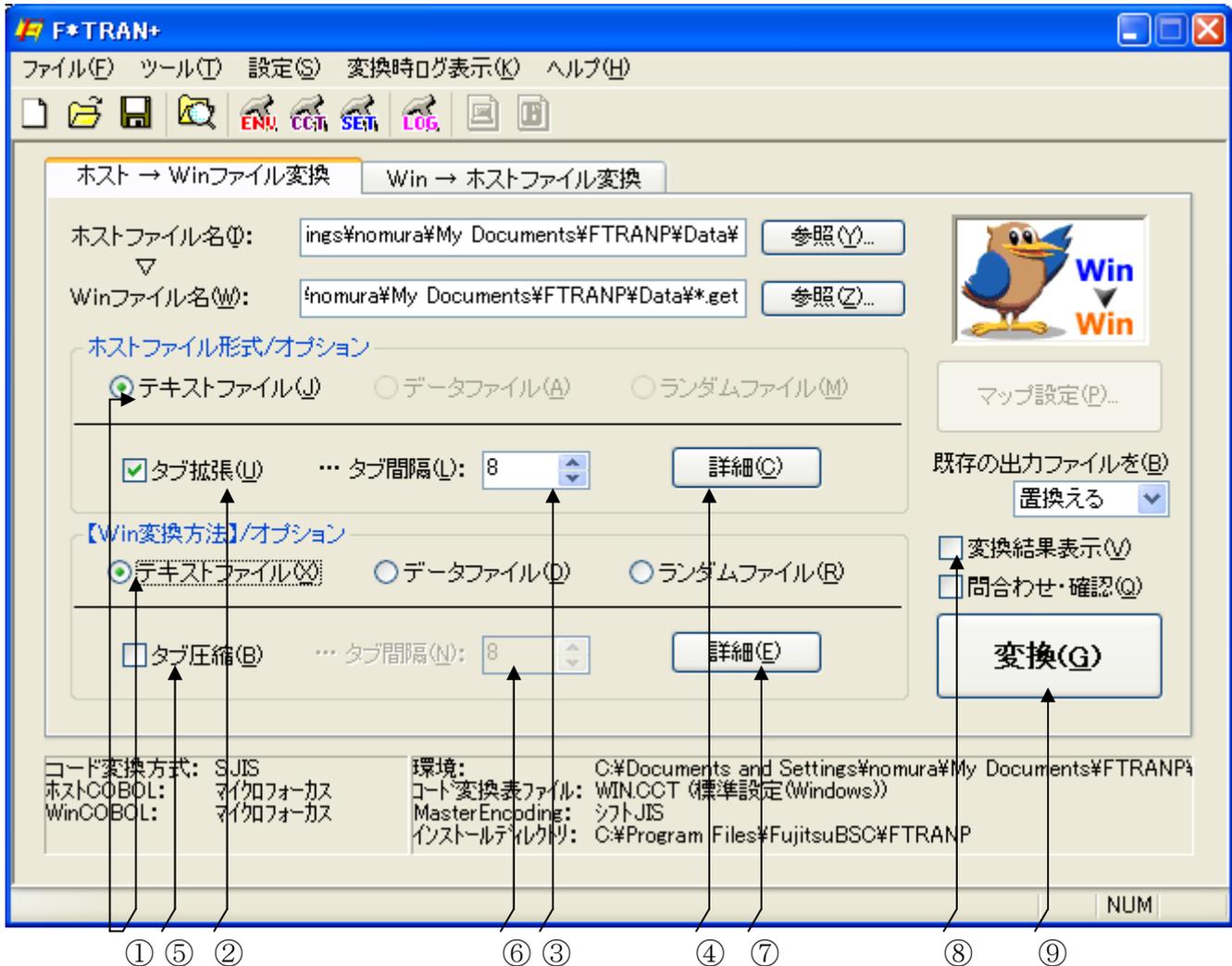
ホスト→Winテキストファイル変換の操作方法について説明します。

●ホストが汎用機・オフコンの場合



- ①変換方法のテキストファイル (X) をクリックし、ホスト→Winテキストファイル変換を選択します。
- ②ホストファイルのレコード長を指定します。
- ③ANK変換かAnk・漢字 (KI/KO付) 混在変換かを指定するオプションです。
- ④タブ圧縮の有無を指定するオプションのチェックボックスです。
- ⑤④でタブ圧縮のチェックをONにしたとき、タブ間隔を指定できます。
- ⑥Winファイルの詳細オプションボタンです。EOF付加の選択、Windows側エンコーディングの設定ができます。
- ⑦変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑧変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

●ホストがUnix、Windowsの場合



- ① Win変換方法のテキストファイル (X) をクリックし、ホスト→Winテキストファイル変換を選択します。ホストファイル形式は、テキストファイル (J) 固定になります。
- ② タブ拡張の有無を指定するオプションのチェックボックスです。
- ③ ②でタブ拡張のチェックをONにしたとき、タブ間隔を指定できます。
- ④ ホストファイルの詳細オプションボタンです。EOF検査の選択ができます。
ホストがWindowsの場合のみ有効です。ホストがUnixの場合はありません。
- ⑤ タブ圧縮の有無を指定するオプションのチェックボックスです。
- ⑥ ⑤でタブ圧縮のチェックをONにしたとき、タブ間隔を指定できます。
- ⑦ Winファイルの詳細オプションボタンです。EOF付加の選択、Windows側エンコーディングの設定ができます。
- ⑧ 変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑨ 変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

■オプションの指定

●ホストファイル・レコード長指定 <ホストが汎用機・オフコンの場合のみ>

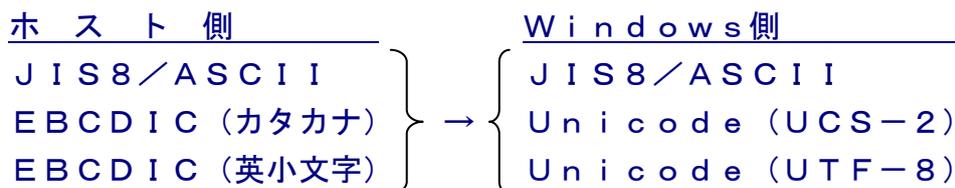
ホストファイルのレコード長を1～32767の範囲の10進数で指定します。デフォルトは、**80バイト**です。レコード長の指定が間違っていると正しいデータ変換が行われませんので、この指定は極めて重要です。

●コード設定オプション <ホストが汎用機・オフコンの場合のみ>

コード変換の方法（ANK変換かAnk・漢字（KI/KO付）混在変換か）を指定します。

Ank指定

Ank指定すると、すべてANKデータとして変換します。これがデフォルトです。以下に示す、



の9通りの変換が可能です。あらかじめ、変換設定のANKコード設定でホストファイル側のコード系を設定しておかなければいけません（ふつう、セットアップ時に1回だけ行います）。漢字がまじっているときは、つぎのAnk・漢字（KI/KO付）混在指定を使ってください。

Ank・漢字（KI/KO付）混在指定

ANK・漢字混在で、KI/KOもついているとき、この指定をします。あらかじめ、変換設定の漢字変換方式設定で適切な漢字変換方式の割り当てをする必要があります（ふつう、セットアップ時に1回だけ行います）。

●タブ拡張のオプション <ホストがUnix、Windowsの場合のみ>

タブ拡張の有無と、タブ拡張するときのタブ間隔を指定します。タブ拡張とは、TAB (09H) をつぎのタブ位置の直前までの連続スペースに展開することです。

タブ拡張のチェックボックスを、

ONにすると、タブ拡張する指定

OFFにすると、タブ拡張しない指定 となります。

タブ拡張指定がONであれば、タブ拡張します。タブ間隔は、2～255の範囲で指定し、それがレコードのおわりまで繰り返し適用されます。**タブ間隔のデフォルトは標準のタブ間隔(8桁きざみ)になります。**これがデフォルトです。

タブ拡張指定がOFFであれば、タブ拡張はしません。

●EOF検査指定 <ホストがWindowsの場合のみ>

EOFコード(1AH)を検査するか否かを指定します。

EOF検査のチェックボックスを、

ONにすると、EOFコードを検査する指定

OFFにすると、EOFコードを検査しない指定 となります。

EOF検査指定がONであれば、EOFコードを検査し、EOFコードが現れたら変換を終了します。こちらがデフォルトです。

EOF検査指定がOFFであれば、EOFコードを検査しません。単なるデータとして扱います。

●タブ圧縮のオプション

タブ圧縮の有無と、タブ圧縮するときのタブ間隔を指定します。タブ圧縮とは、タブ位置の直前までつづく2個以上の連続スペースを、TAB (09H) に置き替えることです。

タブ圧縮のチェックボックスを、

ONにすると、タブ圧縮する指定

OFFにすると、タブ圧縮しない指定 となります。

タブ圧縮指定がONであれば、タブ圧縮します。タブ間隔は、2～255の範囲で指定し、それがレコードのおわりまで繰り返し適用されます。**タブ間隔のデフォルトは標準のタブ間隔(8桁きざみ)**になります。

タブ圧縮は、ソースプログラムなど空白部分が多いファイルの、変換後のファイル容量を減らすのに効果的です。

なお、文字列定数中のスペースまでTABに変換してしまうことを避けるため、アポストロフィ (') か引用符 (") が見つかったら、その行についてはそこでタブ圧縮を打ち切ります。

タブ圧縮指定がOFFであれば、タブ圧縮はしません。これがデフォルトです。

●EOF付加指定

EOFコード (1AH) の扱いを指定します。

EOF付加のチェックボックスを、

ONにすると、EOFコードをつける指定

OFFにすると、EOFコードをつけない指定 となります。

EOF付加指定がONであれば、WindowsファイルのおわりにEOFコードをつけます。現在では少なくなりましたが、テキストファイルのおわりにEOFコードがついていないとエラーにするソフトがあります。その場合にも対処するための機能です。

EOF付加指定がOFFであれば、EOFコードはつけません。これがふつうだと思ってください。こちらがデフォルトです。

●エンコーディング

Windows側のエンコーディングを以下の4通りから選択します。

Master Encodingに従う (詳細は「4.12 環境設定」を参照)
シフトJIS
Unicode (UCS-2)
Unicode (UTF-8)

Unicodeの場合は以下の選択も可能です。

BOMつき
BOMなし
BOM自動

『BOM自動』の指定ではUCS-2ではBOMを付け、UTF-8ではBOMは付けません。

さらにUCS-2の場合は以下の選択も可能です。

Little Endian
Big Endian

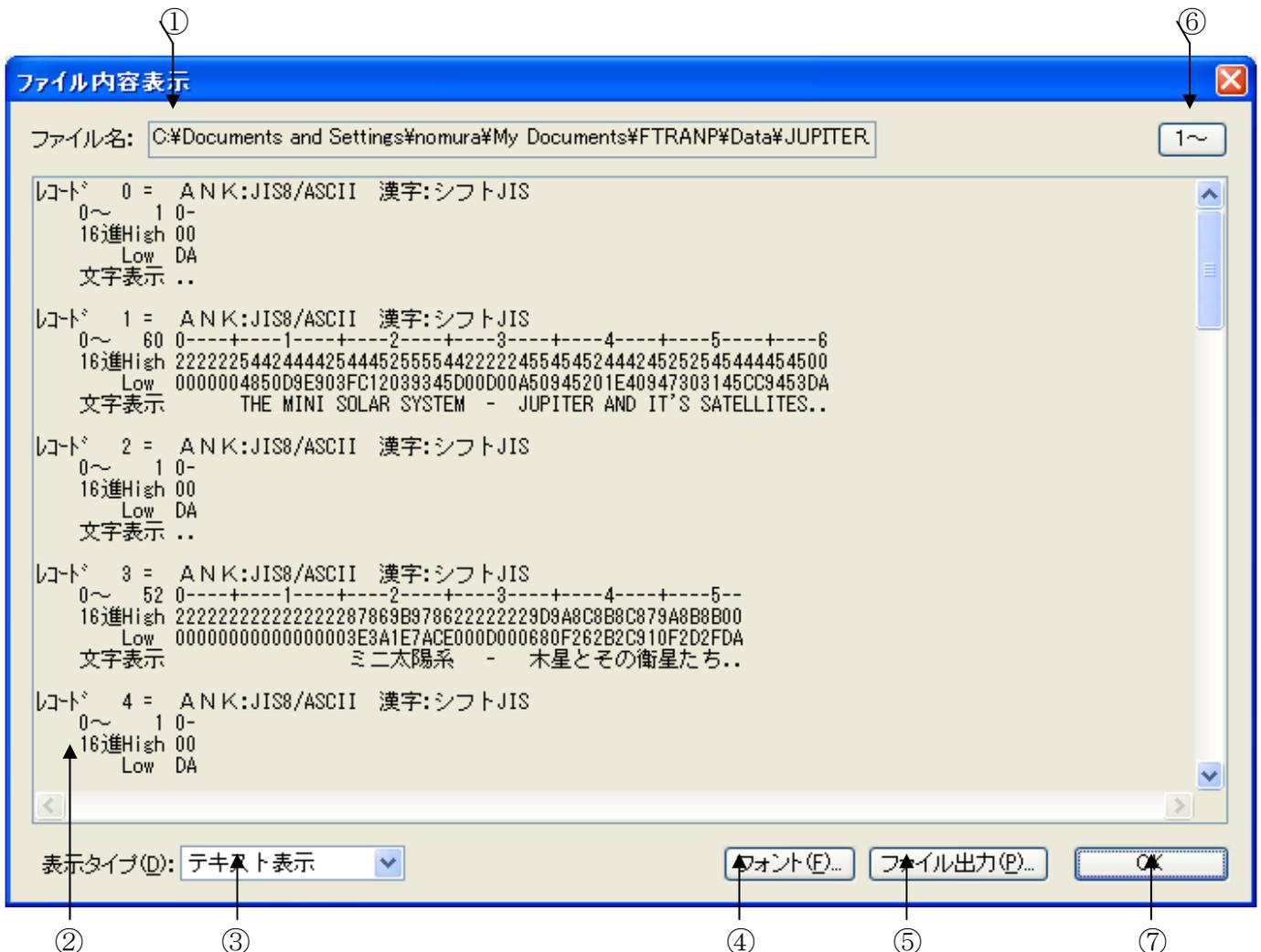
■ホスト→Winテキストファイル変換の実行

変換ボタンをクリックすると、ホスト→Winテキストファイル変換が始まります。

行末の空白類の削除（行末圧縮）と改行コード（CR/LF）の付加は無条件に行われます。

●ファイル内容表示ウインドウ

変換結果表示 (V) のチェックボックスをONにし、ホスト→Winテキストファイル変換を実行すると、実行ウインドウを閉じた後につぎのファイル内容表示ウインドウが開き、変換先のファイルの中身を確認することができます。



①変換先のファイル名が表示されます。

②変換先のファイルの中身が、③で選択されている表示タイプで出力されます。

- ③②の表示タイプをランダム表示、テキスト表示、テキストのみ表示に切り替えます。
- ④フォント (F) ボタンをクリックして、②の出力文字フォントを切り替えることができます。
- ⑤ファイル出力 (P) ボタンをクリックして、②の内容をファイルに保存できます。
- ⑥桁位置のガイド表示を、0 起点 (先頭を0として数える：デフォルト)、
1 起点 (先頭を1として数える) のどちらかに切り替えることができます。
- ⑦OKボタンをクリックすると、ファイル内容表示ウインドウを閉じます。

ファイル内容表示は変換結果のプレビュー表示機能のため、表示できる容量に制限があります。全ての内容が表示されている場合はデータの最後に、

----- ファイルの最後まで表示しました。 -----

全ての内容が表示できなかった場合には、

----- ファイルの途中で表示を切りました。 -----

と表示されます。

■注意事項

漢字があるときはコード設定オプションをAnk・漢字(KI/KO付)混在にすることを忘れずに

漢字が入っているときは、あらかじめ

変換設定の漢字変換方式設定で、適当な漢字変換方式を割り当てておく

のを忘れないでください。また、ホストが汎用機・オフコンの場合、変換のとき、

コード設定オプションをAnk・漢字(KI/KO付)混在にするのを忘れがち

なので注意してください。

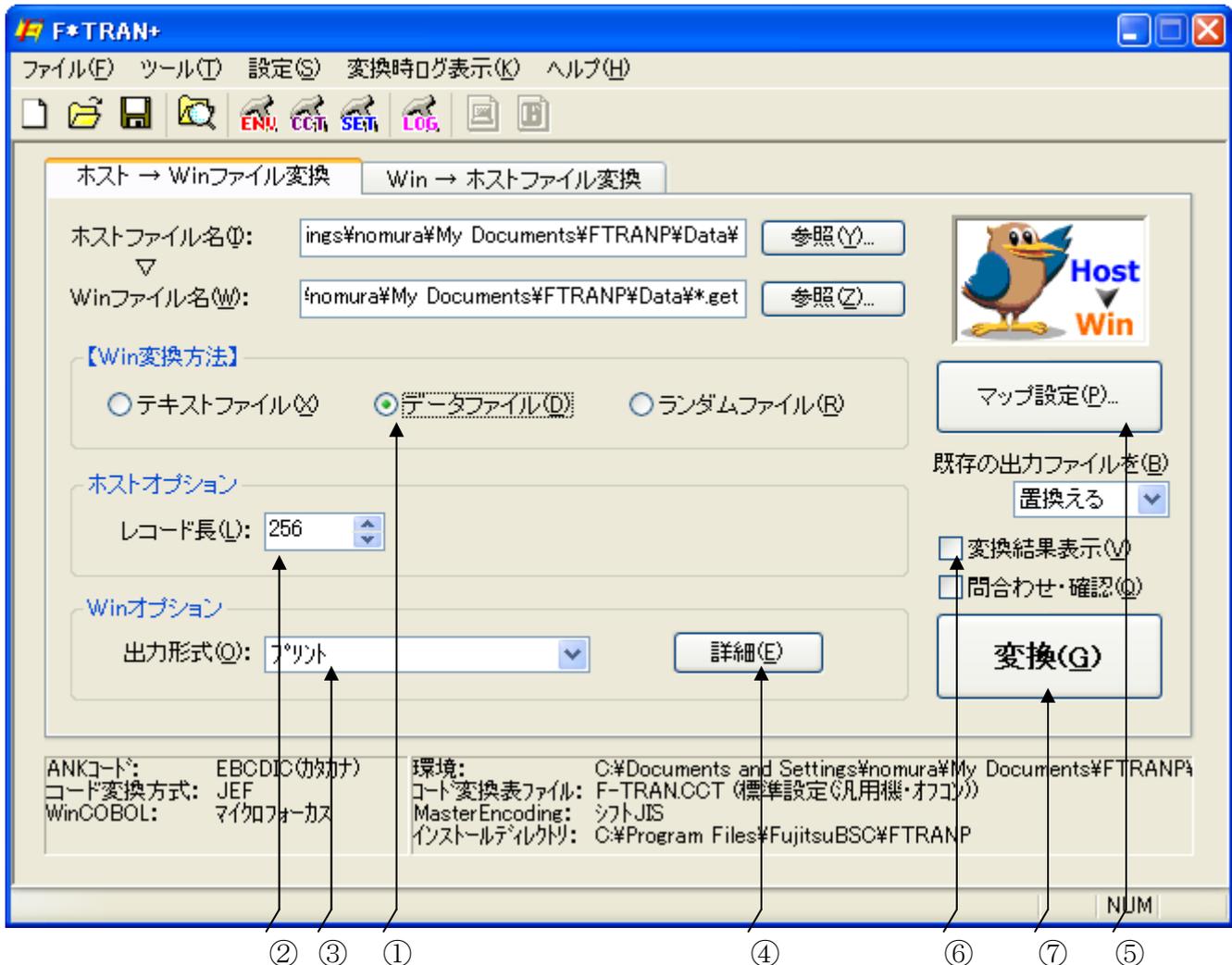
その他の注意事項

「ホスト→Winファイル変換」の節を参照してください。

4.4 ホスト→Winデータファイル変換

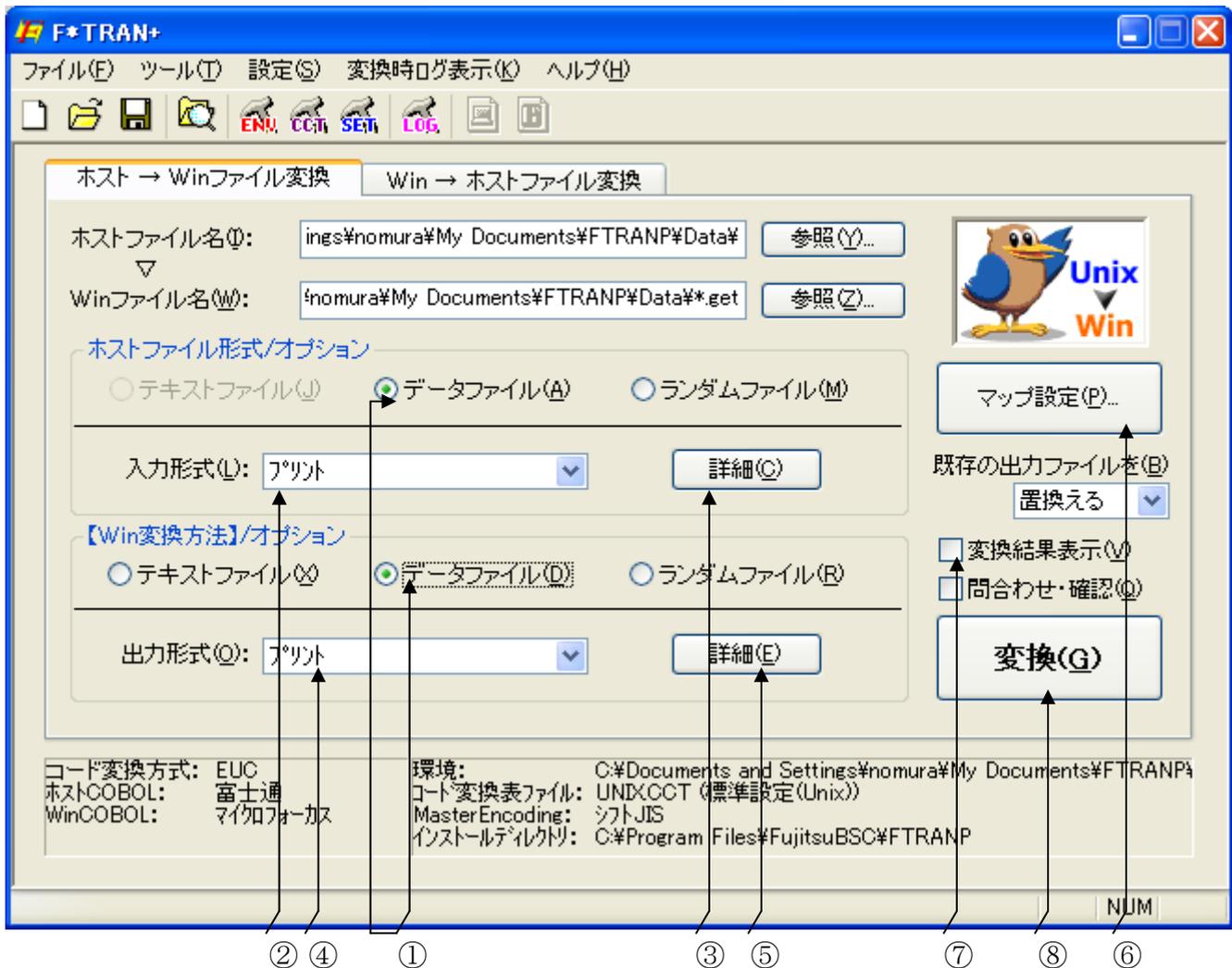
ホスト→Winデータファイル変換の操作方法について説明します。

●ホストが汎用機・オフコンの場合

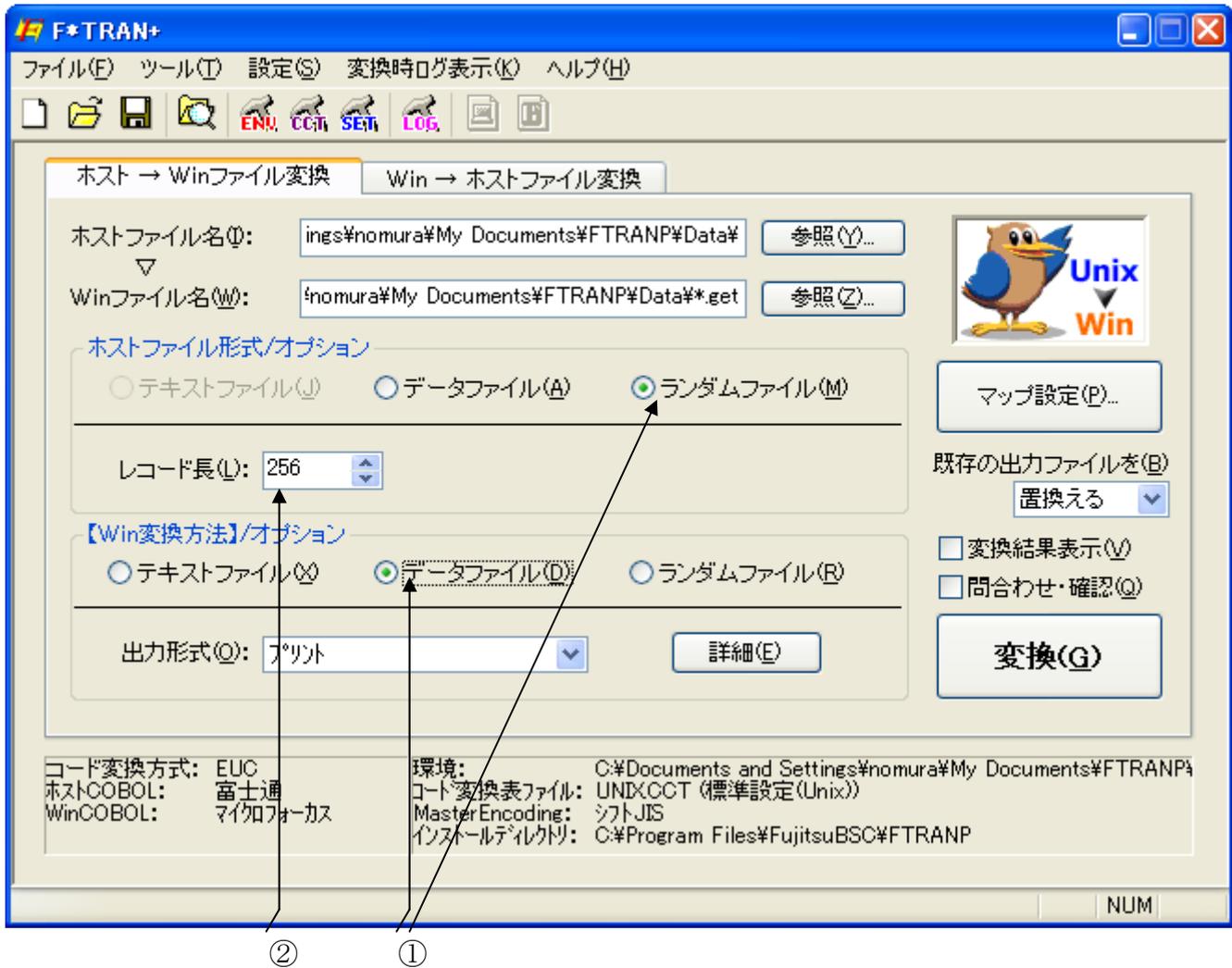


- ①変換方法のデータファイル (D) をクリックし、ホスト→Winデータファイル変換を選択します。
- ②ホストファイルのレコード長を指定します。
- ③出力ファイル形式を指定するオプションです。
- ④Winファイルの詳細オプションボタンです。EOF付加、Windows側エンコーディングの設定ができます。
- ⑤マップ設定 (レコードレイアウト設定) をするオプションボタンです。
- ⑥変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑦変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

●ホストがUnix、Windowsの場合



- ① Win変換方法のデータファイル (D) をクリックし、ホスト→Winデータファイル変換を選択します。ホストファイル形式は、データファイル (A)、ランダムファイル (M) の選択ができますが、上の例はホストファイル形式がデータファイル (A) です。
- ② 入力ファイル形式を指定するオプションです。
- ③ ホストファイルの詳細オプションボタンです。タブ拡張、空行無視等の指定ができます。
- ④ 出力ファイル形式を指定するオプションです。
- ⑤ Winファイルの詳細オプションボタンです。EOF付加、Windows側エンコーディングの設定ができます。
- ⑥ マップ設定 (レコードレイアウト設定) をするオプションボタンです。
- ⑦ 変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑧ 変換を実行するボタンです。このボタンをクリックすると、変換が始まります。



① Win変換方法のデータファイル (D) をクリックし、ホスト→Winデータファイル変換を選択します。ホストファイル形式は、データファイル (A)、ランダムファイル (M) の選択ができますが、上の例はホストファイル形式がランダムファイル (M) です。

② ホストファイルのレコード長を指定します。

■オプションの指定

●ホストファイル・レコード長指定 <ホストファイルが固定長の場合のみ>

ホストファイルのレコード長を1～32767の範囲の10進数で指定します。デフォルトは、**256バイト**です。レコード長の指定が間違っていると正しいデータ変換が行われませんので、この指定は極めて重要です。

●入力形式オプション <ホストファイルが可変長の場合のみ>

変換するホストファイルがプリント形式かデリミタ形式か、デリミタ形式ならコンマ区切り・タブ区切り・スペース区切りのどれなのかを指定します。

プリント ----- プリント形式を変換する
 デリミタ (コンマ) ----- デリミタ形式 (コンマ) を変換
 デリミタ (タブ) ----- デリミタ形式 (タブ) を変換
 デリミタ (スペース) ---- デリミタ形式 (スペース) を変換

プリントを指定すると、デリミタ (区切り文字) なしのプリント形式のファイル (固定長テキストファイル [SDF形式])。ただし、最終項目だけ可変長でもよい) を変換できます。

デリミタを指定すると、デリミタ形式 (区切り文字つき) のテキストファイルを変換できます。区切り文字の種類によって、さらに細かい形式が決まります。デリミタは、

コンマ区切り形式 (CSV形式、K3形式)

タブ区切り形式 (TAB=09H)

スペース区切り形式 (SP=20H)

の3つのなかから指定できます。

後述のマップ設定オプションで、デリミタ検出=コンマ (,) を指定したところに、上記の区切り文字があるとみなされます。

● E O F 検査指定 <ホストがWindowsで、可変長ファイルの場合のみ>

E O F コード (1 A H) を検査するか否かを指定します。

E O F 検査のチェックボックスを、

ONにすると、E O F コードを検査する指定

OFFにすると、E O F コードを検査しない指定 となります。

E O F 検査指定がONであれば、E O F コードを検査し、E O F コードが現れたら変換を終了します。こちらがデフォルトです。

E O F 検査指定がOFFであれば、E O F コードを検査しません。単なるデータとして扱います。

● タブ拡張のオプション <ホストファイルが可変長の場合のみ>

タブ拡張の有無と、タブ拡張するときのタブ間隔を指定します。タブ拡張とは、TAB (0 9 H) をつぎのタブ位置の直前までの連続スペースに展開することです。

タブ拡張のチェックボックスを、

ONにすると、タブ拡張する指定

OFFにすると、タブ拡張しない指定 となります。

タブ拡張指定がONであれば、タブ拡張します。タブ間隔は、2～255の範囲で指定し、それがレコードのおわりまで繰り返し適用されます。**タブ間隔のデフォルトは標準のタブ間隔(8桁きざみ)になります。**これがデフォルトです。

タブ拡張指定がOFFであれば、タブ拡張はしません。

● 空行無視のオプション <ホストファイルが可変長の場合のみ>

空行を無視するかどうかを指定します。

ONにすると、空行を無視する指定

OFFにすると、空行を無視しない指定 となります。

●引用符はずし <ホストがUnix/Windowsで、デリミタ形式の場合のみ>

変換するホストファイルがデリミタ形式の場合には、項目が引用符でくくられていることがあります。その引用符の種類や扱い方を指定します。なお、この設定はデリミタ形式からの変換の場合のみ有効です。

(1) 引用符の種類

変換元のホストファイルで使われている引用符の種類を指定します。

なし 引用符が使われていない時の指定。(もしくは引用符をはずさない指定)

単一引用符 単一引用符「' (半角)」をはずす指定。

2重引用符 2重引用符「" (半角)」をはずす指定。

◆注意 ----- 引用符の種類指定は一種類のみ

ホストファイルに単一引用符(')でくくられた項目と2重引用符(")でくくられた項目が混在していても、どちらか片方の引用符しかはずすことができません。

(2) 引用符項目の取得方法

データの中に引用符を含むのに全体が引用符でくくられていない

,3.5" FDD,

のような例外的な形式のデータの扱いは、ソフトによって解釈がまちまちです。そこで、F*TRAN+では項目取り出し方式を複数用意しておくことで例外的データに対処しています。

【旧F*TRAN風】

- ・ 引用符は必ず対にして使う
- ・ 引用符対は、項目内のどこに何回現れてもよい
- ・ 一对の引用符の中には区切り文字が使われてもよい
- ・ 対の外では、区切り文字をデータとして使えない
- ・ 引用符が2連化してあってもデータとしては認識しない
- ・ 結果的に、すべての引用符が消える

例) ,AAA" B,B" CCC" " DDD,

→

AAAB, BCCDDDD

【新F*TRAN風】

- ・ 基本的には旧F*TRAN風と同様である
- ・ ただし、2連化した引用符をデータとしての引用符として認識する

例) ,AAA” B,B” CCC” ” DDD, → AAAB, BCCC” DDD

【MS-Excel風】

- ◎ 引用符で始まる項目
 - ・ 対になる引用符までは、区切り文字が使われてもよい
 - ・ 対になる引用符までは、2連化した引用符をデータとして扱う
 - ・ 対になる引用符以降は、区切り文字が現れるまで引用符もデータとして扱う
 - ・ 項目内改行はサポートしない
- ◎ 普通の文字で始まる項目
 - ・ 2文字目以降の引用符は、単なるデータとして扱う
 - ・ 区切り文字をデータとして使うことはできない

例) ,3.5” FDD, → 3.5” FDD

【MS-Access風】

- ◎ 引用符で始まる項目
 - ・ 対になる引用符までは、区切り文字が使われてもよい
 - ・ 対になる引用符までは、2連化した引用符をデータとして扱う
 - ・ 対になる引用符以降は、区切り文字が現れるまで削除する
 - ・ 項目内改行はサポートしない
- ◎ 普通の文字で始まる項目
 - ・ 区切り文字をデータとして使うことはできない
 - ・ 2文字目以降に引用符が現れると、それ以降区切り文字までを削除する

例) ,3.5” FDD, → 3.5

◆注意 ---- MS-Excel風とMS-Access風について

MS-Excel風とMS-Access風では、引用符は項目先頭と項目の終わりで各1つずつ存在するのを正常系とし、それ以外については例外系として扱う方式です。両者の違いは、例外系データの扱い方に限られます。入力側データが正常系でありさえすれば、両者に違いは生じないわけです。本マニュアルでのサンプルの例はあくまで例外系のものと認識してください。

◆注意 ---- 「引用符項目の取得方法」の存在について

「引用符の項目の取得方法」はホストがUnix/Windowsで、デリミタ形式の場合に指定可能なオプションです。ホストが汎用機・オフコンの場合には指定できません。

●出力形式オプション

データとしてのテキストファイルにはいくつもの形式があるので、どの形式にするかを指定します。つぎの、7種類の中から選択します。

プリント	-----	プリント形式に変換する
デリミタ (コンマ	・ 圧縮)	---- デリミタ形式 (コンマ) に変換、圧縮あり
デリミタ (タブ	・ 圧縮)	---- デリミタ形式 (タブ) に変換、圧縮あり
デリミタ (スペース	・ 圧縮)	---- デリミタ形式 (スペース) に変換、圧縮あり
デリミタ (コンマ	・ 非圧縮)	---- デリミタ形式 (コンマ) に変換、圧縮なし
デリミタ (タブ	・ 非圧縮)	---- デリミタ形式 (タブ) に変換、圧縮なし
デリミタ (スペース	・ 非圧縮)	---- デリミタ形式 (スペース) に変換、圧縮なし

プリントを指定すると、デリミタ (区切り文字) なしで変換され、プリント形式 (固定長のテキストファイル [SDF形式]) になります。これがデフォルトです。

デリミタを指定すると、デリミタ形式 (区切り文字付きのテキストファイル) に変換されます。デリミタの種類によってさらに細かい形式が決まります。デリミタは、

コンマ区切り形式 (CSV形式、K3形式)

タブ区切り形式 (TAB=09H)

スペース区切り形式 (SP=20H)

の3つのなかから指定できます。

通常は、コンマ区切り (CSV) 形式への変換が一般的です。さらに、

変換後に圧縮をかける (可変長になる。デフォルト)

変換後に圧縮しない (固定長のまま)

を指定できます。

圧縮をかけると本来不要なスペース、つまり、

レコードの先頭・末尾のスペース

デリミタの前後のスペース

引用符の前のスペース

が削除されます。

圧縮をかける機能があるのは、つぎのような理由があるためです。

変換後のWindowsファイルの容量を、大幅に減らすことができる

一部の市販ソフトでは、不要なスペースがあるとうまくデータを読み込めない

● E O F 付加指定

E O F コード (1 A H) の扱いを指定します。

E O F 付加のチェックボックスを、

O N にすると、E O F コードをつける指定

O F F にすると、E O F コードをつけない指定 となります。

E O F 付加指定が O N であれば、W i n d o w s ファイルのおわりに E O F コードをつけます。現在では少なくなりましたが、テキストファイルのおわりに E O F コードがついていないとエラーにするソフトがあります。その場合にも対処するための機能です。

E O F 付加指定が O F F であれば、E O F コードはつけません。これがふつうだと思ってください。こちらがデフォルトです。

● エンコーディング

W i n d o w s 側のエンコーディングを以下の 4 通りから選択します。

M a s t e r E n c o d i n g に従う (詳細は「4. 12 環境設定」を参照)
シフト J I S
U n i c o d e (U C S - 2)
U n i c o d e (U T F - 8)

U n i c o d e の場合は以下の選択も可能です。

B O M つき
B O M なし
B O M 自動

『B O M 自動』の指定では U C S - 2 では B O M を付け、U T F - 8 では B O M は付けません。

さらに U C S - 2 の場合は以下の選択も可能です。

L i t t l e E n d i a n
B i g E n d i a n

●引用符くくり

(1) 引用符の種類

この設定は出力形式がデリミタ形式に指定されているときのみ有効です。引用符の種類それぞれの意味を説明します。

- なし 引用符でくくらずに変換内容を出力します。
- 単一引用符 単一引用符「'（半角）」をくくる。
- 2重引用符 2重引用符「"（半角）」をくくる指定。

「なし」の指定の場合は、Map文で引用符くくりの指定がしてあっても無視します。

◆注意 ---- Map文の引用符記号は、常に二重引用符「"」

「引用符の種類」については「単一引用符」指定時も、Map文において引用符くくりを使う記号は、2重引用符「"（半角）」ですので注意してください。

◆注意 ---- 引用符の種類指定は一種類のみ

単一引用符（'）と2重引用符（"）の両方を混在してくくることはできません。必ずどちらか一方でくくられます。

(2) データとしての引用符の扱い

- 削除 項目のデータとしての引用符は削除する指定。
- 空白化 項目のデータとしての引用符は半角スペースに変換する指定。
- 2連化 項目のデータとしての引用符は、2連化して出力する指定。

2連化を指定した場合、変換後データの項目長が変換前と変わりますが、それについては「2連化時の優先処理」で指定します。

◆注意 ---- 「データとしての引用符の扱い」のデータ内引用符は「引用符の種類」に依存

「データとしての引用符の扱い」ではデータ内容として引用符が存在した時の扱いを指定しますが、その「引用符」というのは「引用符の種類」で指定したものに限りません。

変換処理で使用する引用符として、2重引用符を指定した場合は、単一引用符は空白化などの処理の対象になることはありません。

(3) 2連化時の優先順位

引用符の2連化によって入力データ長に対する変換後データ長が膨れ上がった場合の処理を指定します。

出力幅優先 出力幅を優先して、データの後方部分を切り捨てる指定。

データ優先 変換後の長さそのままを出力する指定。

「2連化時の優先順位」は「データとしての引用符の扱い」において「2連化」を指定していない場合には有効になりません。

◆注意 ---- 「2連化時の優先順位」にてデータ優先をした時の出力幅について

「2連化時の優先順位」で「データ優先」を指定すると、Map文で出力幅指定を省略した時だけでなく出力幅指定をしていたとしてもそれを無視して変換します。下は変換処理時の実際の出力幅です。

出力幅指定をした場合	…	Map文指定の出力幅+引用符の長さ
出力幅指定を省略した場合	…	Map文指定の入力幅+引用符の長さ

●マップ設定オプション

このマップ設定オプションで、項目別の細かい変換方法を指示します。本来なら、自動的に項目を認識して変換ができると便利です。しかし、ホストの、とくにCOBOLのデータには、データ自身に桁数や小数点位置の判断に必要な情報が含まれていない、という特性があります。そのため自動変換は原理的に不可能なのです。

マップ設定のボタンをクリックすると、つぎのウインドウが開きます。



マップ設定には、

- 簡易設定 —— 設定ウインドウに従って、必要な値を入力するだけで設定ができます。通常は、こちらの機能だけで間に合う場合がほとんどです。
- 詳細設定 —— フリーフォーマットで、マップ設定を記述するエディタモードです。簡易設定だけでは設定できない特殊な設定をする場合、あるいは、既存のパラメータファイル等を流用したい場合等に使用します。マップ設定の文の書き方を覚えなくては設定できません。

の2通りの方法があります。

◆注意 ---- **簡易設定と詳細設定の組み合わせ**

簡易設定で設定した値は、詳細設定で読み込むことができるテキスト形式に直されます。よって、簡易設定をした後の項目に、詳細設定で特殊設定をつけ足すこともできます。

◆注意 ---- **詳細設定→簡易設定の順で設定した場合の制約**

詳細設定→簡易設定の順で設定した場合は、原形が変わってしまう場合があります。これは、詳細設定で設定した項目が簡易設定で読み込まれることにより、スリム化されてしまうからです。ただし、原形が変わってしまっても、設定内容が変わるわけではありません。

◆注意 ---- **入力幅、出力幅の指定について**

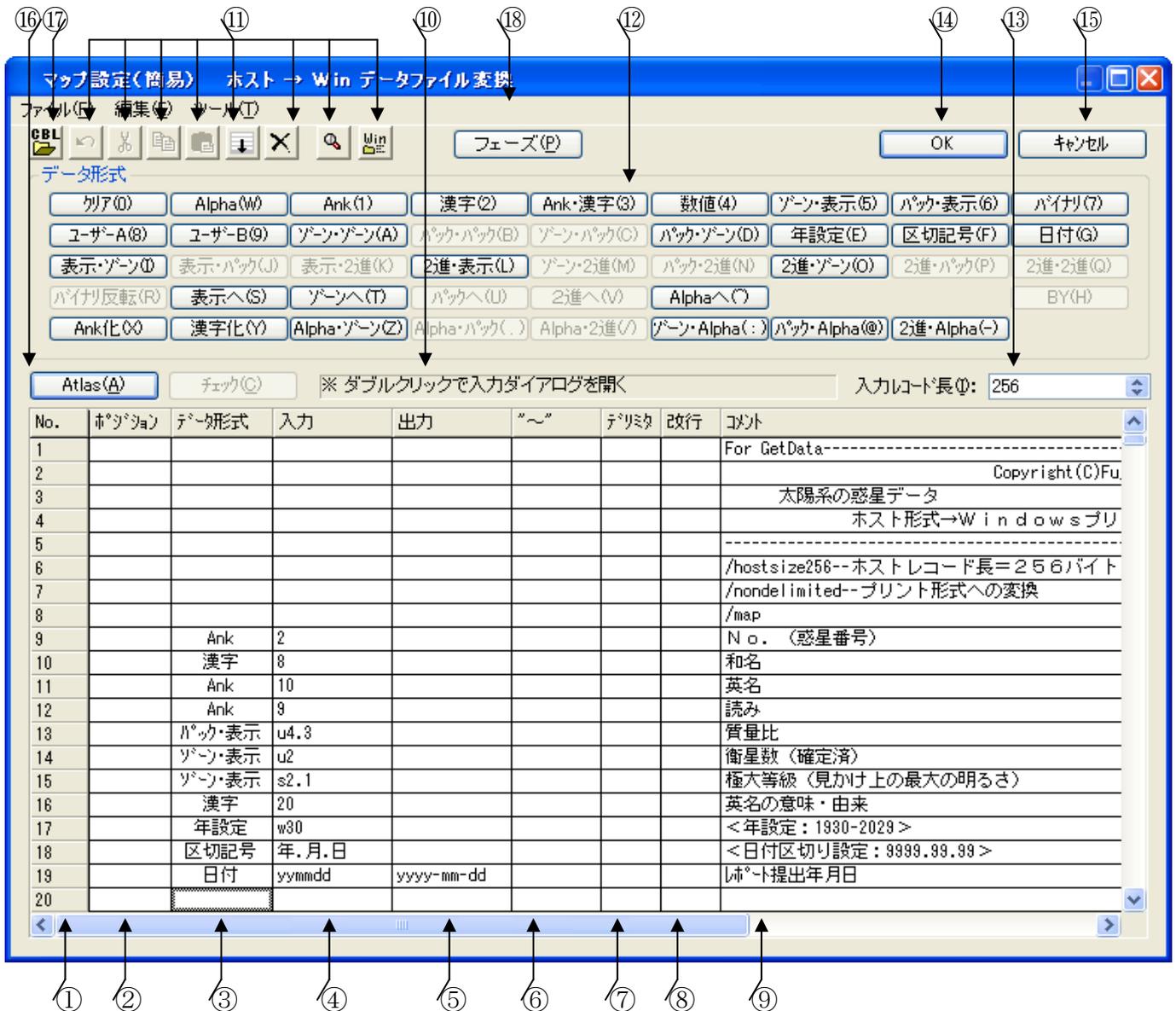
入力側や出力側の文字コードによって、入力幅、および出力幅の指定には注意が必要です。

たとえばANKデータを変換する場合、入力がWindowsでエンコーディングがUCS-2の場合は文字数×2の入力幅を指定する必要があります。

出力幅が省略された場合の省略値は入力コードと出力コードから入力幅に指定された値を最小として自動的に計算して求めています。ただし、漢字とANKが混ざった項目に対して漢字項目として変換した場合（ANK文字が全角化される）や、汎用機・オフコンの半角カナをANK変換でUTF-8に変換する場合など、明確に出力幅を指定しないと出力データが途中で切れてしまう場合もあります。

文字コードとしてUnicodeを指定する場合や半角カナ文字がある場合などは、入力幅や出力幅の指定には十分注意する必要があります。

<マップ設定 (簡易) ウィンドウ>



- ①No. は、1～2000です。ここは非入力項目で、行選択の時にクリックします。
- ②項目の桁位置等を入力します。通常、項目を組み替えるとき以外は入力しません。
ダブルクリックすると、入力サブウィンドウが開きます。
- ③項目のデータ形式を指定します。ダブルクリックすると、入力サブウィンドウが開きます。
- ④項目の入力幅を入力します。省略すると、残りのバイト数を指定したことになります。
データ形式がゾーン/パック/2進指定の場合、ピクチャまたは2進ピクチャで指定します。
Ank/漢字/Ank・漢字/バイナリの場合、定数を指定することもできます。
Alphaへ/表示へ/ゾーンへの場合、数値定数かシステム変数を入力します。
- ⑤項目の出力幅を入力します。
- ⑥デリミタ形式に変換するときに、項目を引用符でくくる指定ができます。ホストがUnix、Windowsの場合、デリミタ形式の項目の引用符をはずす指定にもなります。

- ⑦デリミタ形式に変換するとき、デリミタ挿入を無効にする指定ができます。
ホストがUnix、Windowsの場合、デリミタ形式に変換するときデリミタ検出を無効にする指定にもなります。
- ⑧改行コードを挿入する指定です。2種類（通常、強制）の指定があります。
改行コード挿入を指定すると、変換後の文字列の後に改行コードを挿入します。
- ⑨コメントの入力ができます。
- ⑩セルポインタの入力ガイダンスメッセージが出力されるフィールドです。
誤入力をした場合のエラーメッセージも出力されます。

⑪は編集（E）、ツール（T）のメニューの機能がボタン化されています。



直前の編集作業が無効になり、元に戻ります。



1行または複数行を選択（①の操作）をした後に、このボタンをクリックすると、選択した項目の内容が切り取られ、カットバッファに入ります。



1行または複数行を選択（①の操作）をした後に、このボタンをクリックすると、選択した項目の内容がカットバッファに入ります。



カットバッファに入った内容を、セルポインタがある行へ貼りつけます。
すでに、セルポインタ以降の行に設定項目がある場合は、挿入になります。



セルポインタがある行以降が1行ずつ下がります。通常は行挿入をする場合に使用します。



選択した項目の内容が削除されます。



変換プレビュー（変換前／変換後のデータ確認）ができます。



Winファイルエディタが起動します。
ホストファイルの中身を見ること（編集も可能）ができます。

- ⑫セルポインタが選択項目にある場合に、選択肢のボタンが表示されるエリアです。
- ⑬ホストファイルが固定長の場合はレコード長、可変長の場合はデータ形式を指定できます。
変換プレビューの時の重要項目です。
- ⑭設定が完了したら、OKボタンをクリックします。
- ⑮キャンセルボタンをクリックすると、すべての入力／編集作業が無効になります。
- ⑯マルチレコードの設定ができるAtlasモードに切り替えるボタンです。*1
- ⑰COBOLのソースファイル（Copy句）を読み込むボタンです。*1
- ⑱フェーズボタンをクリックすると、フェーズを設定するウィンドウが開きます。*1

*1) 詳細は、マルチレコード編のマニュアルを参照してください。

◆注意 ----- **マップ未設定時のデータ変換動作**

マップ未設定の場合は、すべてをANKデータとみなしてデータ変換を行います。

◆注意 ----- **デリミタ挿入について**

マップ設定（簡易）で項目を設定し、デリミタ形式に変換する場合、自動的に変換後の項目の後にデリミタ（区切り文字）が挿入されます（ただし、最終項目は除く）。挿入される区切り文字は指定された出力形式により、コンマ、タブ、スペースの3つのうちのいずれかになります。なお、プリント形式への変換のときは、デリミタ挿入は行われません。

また、デリミタに設定したが、ある項目にはデリミタを付加したくない場合などは、「デリミタ」欄を「無効」にします。また、簡易設定後に詳細設定を行い、修正することもできます。

<マップ設定（簡易）サブウィンドウ>

ポジションの欄をダブルクリックすると、つぎのサブウィンドウが開きます。入出力桁位置や入出力スキップの設定を簡便に行うことができます。

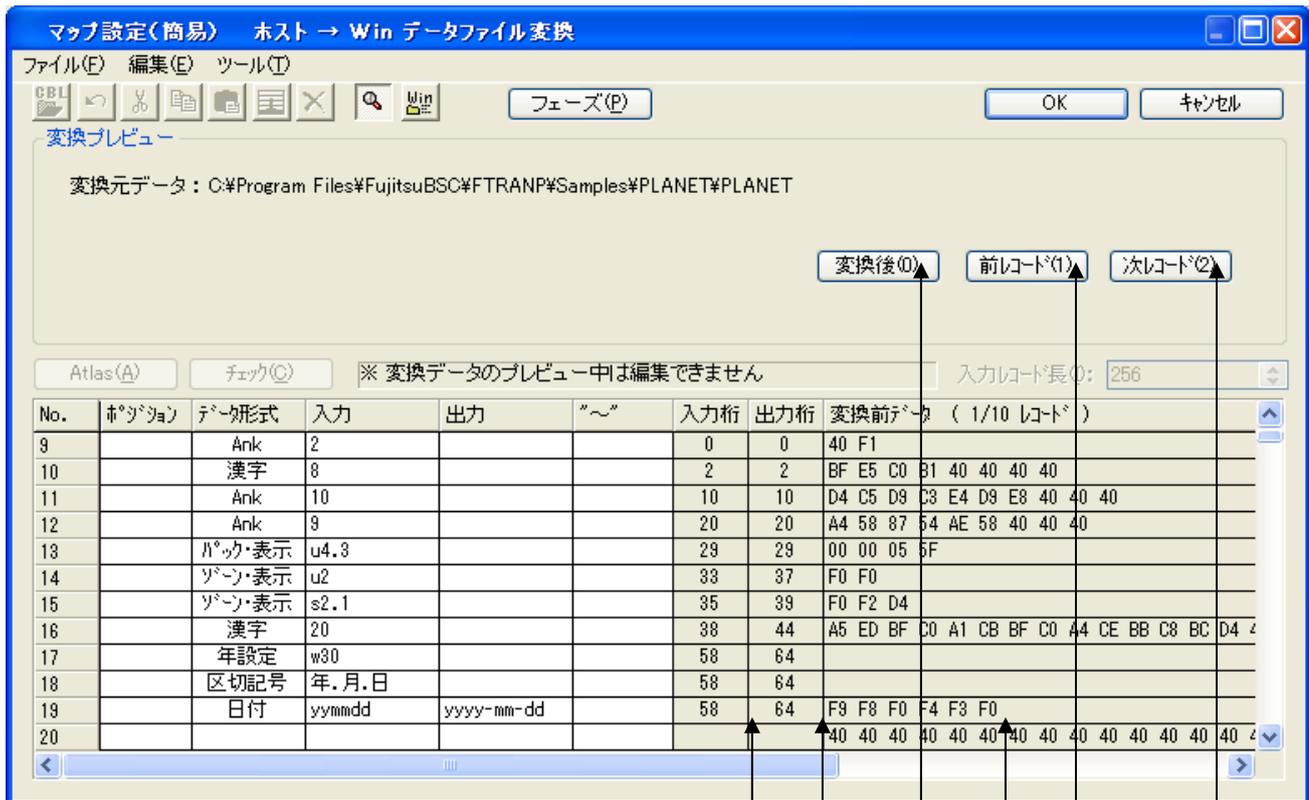


データ形式の欄をダブルクリックすると、つぎのサブウィンドウが開きます。データ形式から引用符くりまでの設定を簡便に行うことができます。

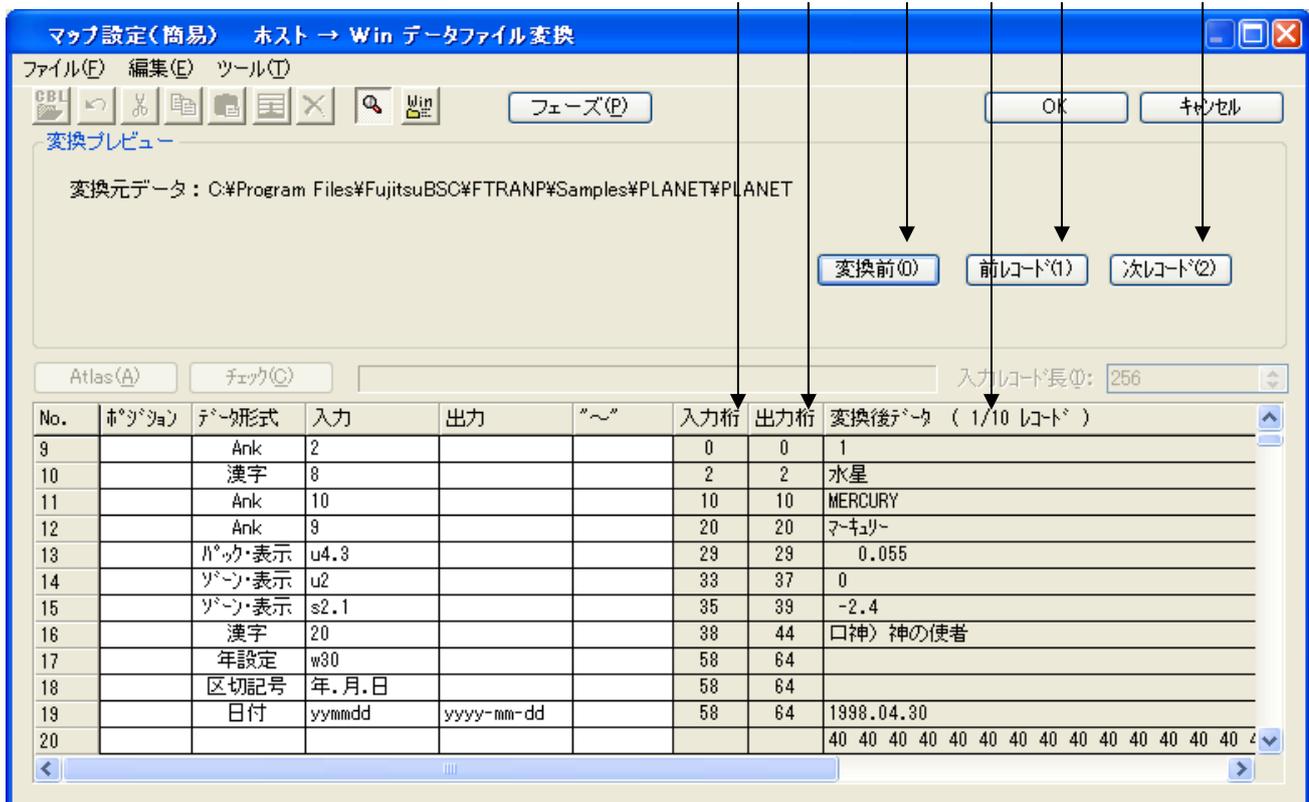


<マップ設定（簡易）・変換プレビュー機能>

変換プレビューボタンをクリックすると、マップ設定（簡易）のウィンドウはつぎのような状態になり、変換前／変換後のデータを確認することができます。



↑変換前データ表示 変換後データ表示↓ ③ ③ ① ④ ② ②



- ①変換前/変換後のデータ表示の切り替えを行うボタンです。④の表示が変わります。
- ②前レコード/次レコード表示のためのボタンです。
- ③ここでは、入力レコード/出力レコードのポジション（桁）が表示されます。
- ④データ表示は、変換前が16進、変換後が文字、未定義個所が16進になります。

(1) ポジション（桁位置）の指定

変換対象にするホスト側（入力）の桁位置や、変換結果を書き込むWindows側（出力）の桁位置を、別の任意の位置に移動できます。現在、処理対象にしている桁位置を、この指定で強制的に変更できます。この機能を利用すると、項目の組み替えなどが簡単に実現できます。

入力桁位置は、ふつう10進数で指定します。式による指定もでき、そのなかでは、

\$ **ホスト側（入力）のレコード長**
. **ホスト側（入力）の現在の桁位置** という特殊変数が使えます。

たとえば、

\$-8 と指定すれば、 **ホスト側（入力）レコード末尾の8バイト前に位置する**
. **-40** と指定すれば、 **現在の入力桁位置から40バイト戻る**

という意味になります。

プリント形式への変換の場合、出力桁位置を移動することもできます。ふつう、

: 10進数 で桁位置を指定します。

10進数の代わりに式による指定もでき、そのなかでは、

. **Windows側（出力）の現在の桁位置** という特殊変数が使えます。

◆注意 ---- デリミタ形式の場合、出力桁位置指定はない

デリミタ形式への変換の場合、入力桁位置の移動だけが有効です。出力桁位置の指定は行わないでください。

◆注意 ---- 先頭を0桁目とする

F*TRAN+では、レコードの先頭を0桁目として数えます。

(2) ポジション (デリミタ位置) の指定

変換対象にするホスト側 (入力) データのデリミタ形式の項目位置を、別の任意の項目位置に移動できます。現在、処理対象にしている項目位置を、この機能で強制的に変更できます。この機能を利用すると、デリミタ形式の項目の組み替えなどが簡単に実現できます。

デリミタ位置は、 **@@入力項目番号** で指定します。式による指定もでき、そのなかでは、

- .** ホスト側 (入力) の現在の項目位置
- \$** ホスト側 (入力) の入力全体の項目数
- *** ホスト側 (入力) の残りの項目数 という特殊変数が使えます。

たとえば、

- @@. - 2** と指定すれば、 **今の入力項目から 2 項目戻る**
- @@\$ - 2** と指定すれば、 **ホスト側 (入力) レコード末尾から 2 項目に位置する**

という意味になります。

◆注意 ---- 先頭を 0 項目目とする

F * TRAN+では、レコードの先頭項目を 0 項目目として数えます。

(3) ポジション（入力スキップ）の指定

ホスト側（入力）レコードに不要な項目があるとき、それをスキップして変換できます。

スキップする幅は、 `^バイト数` で指定します。

たとえば、3バイト分スキップしたいなら、 `^3` と指定します。

バイト数は省略でき、省略すると1バイトとみされるので、

`^3` は `^^^` と指定したのと同じです。

スキップする幅は式による指定もでき、そのなかでは、

* `ホスト側（入力）の残りバイト数` という特殊変数が使えます。

(4) ポジション（出力スキップ）の指定

入力スキップとは逆に、Windows側（出力）に何桁か空きを作ることもできます。プリント形式への変換のとき、空項目を作るのがおもな用途です。

スキップする幅は、 `__バイト数` で指定します。

たとえば、3バイト分スキップしたいなら、 `__3` と指定します。

バイト数は省略でき、省略すると1バイトとみされるので、

`__3` は `_____` と指定したのと同じです。

(5) データ形式

[Alpha]

ホスト側のデータに関係なく1バイトごとにANK項目として変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。出力側のコード体系に関係なく、ANKコードの設定に従って1バイトデータに変換されて出力します。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ **ホスト側（入力）のレコード長**
- * **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のANK項目のおわりにスペース（20H）が詰められます。

[Ank]

ANK項目を変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ **ホスト側（入力）のレコード長**
- * **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅の代わりにANK形式の定数を指定することもできます。指定した定数はWindows側に指定されたコードに変換されて出力されます。入力値として、つぎの定数が使えます。

文字列定数	文字列はANK形式
汎用定数	Space、LowValue、HighValue

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。入力に定数を指定した場合は、出力幅の省略値は「**見た目の文字数×2**」になります。指定された定数は内部的にUCS-2コードなので、出力コードがUCS-2以外の場合は明示的に出力幅を指定する必要があります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のANK項目のおわりにスペースが詰められます。

[漢字]

漢字項目を変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します（漢字の文字数ではありません）。式による指定もでき、そのなかでは、

- \$ **ホスト側（入力）のレコード長**
- * **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅の代わりに漢字形式の定数を指定することもできます。指定した定数はWindows側に指定されたコードに変換されて出力されます。入力値として、つぎの定数が使えます。

文字列定数	文字列は漢字形式
汎用定数	Space

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。入力に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を大きくしなければならない場合があります。拡張漢字のクエスチョン変換を使っていて、オーバーフローの危険があるときなどです（最大で入力幅の3倍）。そのときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、漢字項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、Windows側の漢字項目のおわりに漢字変換方式で設定されている漢字スペースが詰められます。

項目長を大きくしなければならない場合があります。拡張漢字のクエスチョン変換を使っていて、オーバーフローの危険があるときなどです（最大で入力幅の3倍）。そのときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、漢字項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、Windows側の漢字項目のおわりに漢字変換方式で設定されている漢字スペースが詰められます。

[Ank・漢字]

ANK・漢字まじり項目を変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。ホストが汎用機・オフコンの場合、漢字の前後にKI/KOがついていないと、この変換方法は適用できません。変換後、KI/KOが取れて、その分、左詰めされます。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ **ホスト側（入力）のレコード長**
- * **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅の代わりにANK・漢字まじり形式の定数を指定することもできます。指定した定数はWindows側に指定されたコードに変換されて出力されます。入力値としてつぎの定数が使えます。

文字列定数	文字列はANK・漢字まじり形式
汎用定数	Space、LowValue、HighValue

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。入力に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を変更したい場合があります。KI/KOが取れて短くなることや、逆に、拡張漢字のクエスチョン変換を使ってオーバーフローの危険があるときなどです（最大で入力幅の3倍-KI・KOのバイト数の合計）。そのときは、出力幅を指定します。

項目長を縮めると、ANK・漢字まじり項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、Windows側のANK・漢字まじり項目のおわりにスペースが詰められます。

[数値]

文字形式の数値項目どうしの変換をします。

入力幅は、バイト数で指定します。省略すると、入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

数値変換は、Ank項目変換と後述のゾーン・表示変換の中間的なものです。Ank項目変換と比較すると、

文字形式数値しか通さない

入力幅を省略すると15桁とみなす

右詰めになる

などの点が異なります。

「文字形式数値しか通さない」というのは、具体的には、

＋、－、0～9、ピリオド(.)、E、e、D、d

しか変換しないで、これら以外の文字は捨ててしまうということです。たとえば、通貨記号(¥/\$)や位取りのコンマ(,)などは削除されるので、リストファイルから入力データファイルを作るときなどに役立ちます。

[ゾーン・表示]

ホストのCOBOLのゾーン形式数値項目を、文字形式数値項目に変換します。変換結果は右詰めになります。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**－123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする

＋整数部桁数

＋1＋小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。たとえば、ピクチャ=s4.1、出力幅=省略という指定で出力コードがUCS-2以外の場合、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[パック・表示]

ホストのCOBOLのパック形式数値項目、BCD形式数値項目を、文字形式数値項目に変換します。変換結果は右詰めになります。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、ピクチャは **s4.1** と指定します。**123.4** という数字が3バイトのBCD形式の項目に記録されているとすれば、ピクチャは **b5.1** と指定します。なお、ピクチャは省略できません。

パック形式では、整数部桁数+小数部桁数を奇数にしておくのが通例です。

BCD形式では、整数部桁数+小数部桁数を偶数にしておくのが通例です。

整数部の最上位桁に意味があるのかないのかは、半々の割合です。

出力幅は省略できます。出力幅を省略すると、

符号つきパック形式なら1、符号なしパック形式・BCD形式なら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。たとえば、パック形式でピクチャ=s4.1、出力幅=省略という指定で出力コードがUCS-2以外の場合は、出力幅に7を指定したのと同じ意味になります。BCD形式でピクチャ=b5.1、出力幅=省略という指定で出力コードがUCS-2以外の場合は、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[バイナリ]

バイナリ変換（無変換）は、「コード変換を一切しない」という変換方法です。通常、ホスト→Winデータファイル変換では使用しません。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$** **ホスト側（入力）のレコード長**
- *** **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

また、入力幅の代わりにバイナリ形式の定数を指定することもできます。その場合、指定した定数はコード変換されずに、そのまま挿入されます。入力値として、つぎの定数が使えます。

16進列定数、汎用定数（LowValue、HighValue）

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなされます。入力に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、バイナリ項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のバイナリ項目のおわりにスペースが詰められます。

[ユーザーA], [ユーザーB]

ユーザーA/B変換は、利用者独自のバイト単位の変換処理が必要なときに、ユーザーA/B変換表を書き替えて利用します。ユーザーA/B変換には、Alpha項目変換の説明がほとんどそのまま当てはまります。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$** **ホスト側（入力）のレコード長**
- *** **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ユーザーA/B項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のユーザーA/B項目のおわりにスペースが詰められます。

[ゾーン・ゾーン]

ホストのCOBOLのゾーン形式数値項目を、Windows COBOLのゾーン形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**— 1 2 3. 4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、入力ピクチャは **s 4. 1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s 4. 1という指定は、入力ピクチャ=s 4. 1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s 4. 1、出力ピクチャ=省略という指定は、出力ピクチャ=s 4. 1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[パック・ゾーン]

ホストのCOBOLのパック形式数値項目、BCD形式数値項目を、Windows COBOLのゾーン形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**— 1 2 3. 4** という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、入力ピクチャは **s 4. 1** と指定します。**1 2 3. 4** という数字が3バイトのBCD形式の項目に記録されているとすれば、入力ピクチャは **b 5. 1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s 4. 1という指定は、入力ピクチャ=s 4. 1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s 4. 1、出力ピクチャ=省略という指定は、出力ピクチャ=s 4. 1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[年設定]

日付データ項目を変換する際の、年の2桁（yy）と4桁（yyyy）の変換方式を設定します。Wnn、または、Snnの形式で設定します。Wnnの“W”はウインドウ方式を、Snnの“S”はシフト方式を意味し、“nn”は00～99の数字で指定します。

入力側に **W30** と指定すれば、入力データの年を1930～2029年とみなし、出力側に **S25** と指定すれば、出力データの年下2桁を、-25します。

また、シフト方式（“Snn”指定）では、つぎの特殊指定ができます。

SShowa は、“S25”の指定と同じ（昭和通年方式）

SHeisei は、“S88”の指定と同じ（平成通年方式）

年設定は日付項目変換が実行された時に適用になり、複数の年設定がなされている場合は、日付項目変換の直前の年設定が有効になります。

年設定がない場合の日付項目変換のデフォルトは、入力／出力ともに **W30** となり、入出力データの年を1930～2029年とみなします。

[区切記号]

日付データ項目を出力する際の日付区切り記号を、つぎの3つの中から設定します。入力に、SLASH、または、HYPHEN、または、PERIODと入力します。

<u>指 定 文 字</u>	<u>日付区切り記号</u>	<u>デ ー タ 例</u>
SLASH	／（スラッシュ）	1998／12／31
HYPHEN	－（ハイフン）	1998－12－31
PERIOD	．（ピリオド）	1998．12．31

日付区切り設定は日付項目変換が実行された時に適用され、複数の日付区切り設定がなされている場合は、日付項目変換の直前の日付区切り設定が有効になります。

日付区切り設定がない場合の日付項目変換のデフォルトは、**SLASH** となり、日付区切り記号を“／”にします。

[日付]

日付データ項目を変換します。

入力と出力は、必ず日付マスクの形式で指定し、省略はできません。日付マスクの指定方法については、「操作の基礎」の章で説明しました。たとえば、

入力側に、 `yy m m d d`
 出力側に、 `yy yy - mm - d d` のように指定すると、

コード変換後に、入力側6桁の日付データ項目を、出力側10桁の日付データ項目に編集します。その際に、年設定、日付区切り設定が適用になります。

日付マスクにはオプションが指定可能です。入力、または出力コードがUCS-2の時、-D i s p指定では1桁2バイト単位で処理します。それ以外の場合は1桁1バイト単位で処理します。これが省略値です。-A l p h a指定ではコードに関係なく1桁1バイト(ゾーン形式)で処理します。

[表示・ゾーン]

ホストの文字形式数値項目を、Windows COBOLのゾーン形式数値項目に変換します。通常は、ホストがUnixまたはWindowsの場合に使います。

入力幅は、バイト数で指定します。省略すると入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字を5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が省略値を超えるときは、

入力幅=40、ピクチャ=u15.2 のように、ダミーの入力幅を指定します。

[2進・表示]

ホストの2進形式整数・小数項目を、Windowsの文字形式数値項目に変換します。変換結果は右詰めになります。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

出力幅は省略できます。出力幅を省略すると、ピクチャに従って出力されます。

出力幅、ピクチャ共に省略すると、

入力のバイト数	1	2	3	4	5	6	7	8
出力幅	3	5	8	10	13	15	17	18

の要領で2進キャストから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。例を示すと、入力2進キャスト=i4s、出力幅=省略という指定は、出力コードがUCS-2の場合、出力幅=20という指定と同じです。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[2進・ゾーン]

ホストの2進形式整数・小数項目を、Windows COBOLのゾーン形式数値項目に変換します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を5バイトの項目に記録するとすれば、ピクチャは **s4.1** と指定します。

2進キャストにつづくピクチャも出力ピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、2進キャスト/ピクチャ=i4s4.1、出力ピクチャ=省略という指定は、出力ピクチャ=s4.1という指定と同じです。

[表示へ]

入力した数値定数、またはシステム変数の値を、文字形式数値項目に変換してから、出力側に挿入します。変換結果は右詰めになります。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

入力値は省略できません。

出力幅は省略できます。出力幅を省略すると、「入力した値を正確に出力するために必要な幅」で出力されます。たとえば、入力値 = `[-123.4]`、出力幅 = 省略という指定は出力幅 = 1 (符号) + 3 (整数部桁数) + 1 (小数点) + 1 (小数部桁数) = 6 を指定したのと同じ意味になります。(出力コードが UCS-2 の場合は 12)

[ゾーンへ]

入力した数値定数、システム変数の値を、出力ピクチャに従って `Windows COBOL` のゾーン形式数値項目に変換してから、出力側に挿入します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という定数を入力値として指定し、5 バイトの符号つきゾーン形式の項目に記録するとすれば、出力ピクチャは `s4.1` と指定します。

入力値、出力ピクチャともに省略できません。

[Alphaへ]

入力した数値定数、またはシステム変数の値を、文字形式数値項目に変換してから、出力側に挿入します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

入力値は省略できません。

出力幅は省略できます。出力幅を省略すると、「入力した値を正確に出力するために必要な幅」で出力されます。たとえば、入力値 = [-123.4]、出力幅 = 省略という指定は出力幅 = 1 (符号) + 3 (整数部桁数) + 1 (小数点) + 1 (小数部桁数) = 6 を指定したのと同じ意味になります。

[Ank化]

ホスト側 (入力) 文字列をできる限り ANK (半角) に変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定の ANK コードの設定で決まります (ふつう、セットアップ時に一回だけ行います)。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

\$ **ホスト側 (入力) のレコード長**
***** **ホスト側 (入力) の残りバイト数** という特殊変数が使えます。

入力幅を省略すると * を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のANK項目のおわりにスペースが詰められます。

[漢字化]

ホスト側（入力）文字列をできる限り漢字（全角）に変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します（漢字の文字数ではありません）。式による指定もでき、そのなかでは、

- \$** **ホスト側（入力）のレコード長**
***** **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

[Alpha・ゾーン]

ホストの文字形式数値項目を、Windows COBOLのゾーン形式数値項目に変換します。通常は、ホストがUnixまたはWindowsの場合に使います。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が15バイトを超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が15バイトを超えるときは、

入力幅=20、ピクチャ=u15.2 のように、ダミーの入力幅を指定します。

[ゾーン・Alpha]

ホストのCOBOLのゾーン形式数値項目を、文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
 - 1 2 3 . 4 という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、
 ピクチャは **s 4 . 1** と指定します。なお、ピクチャは省略できません。

出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値が使われます。たとえば、ピクチャ = s 4 . 1、
 出力幅 = 省略という指定は、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[パック・Alpha]

ホストのCOBOLのパック形式数値項目、BCD形式数値項目を、文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
 - 1 2 3 . 4 という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、
 ピクチャは **s 4 . 1** と指定します。**1 2 3 . 4** という数字が3バイトのBCD形式の項目
 に記録されているとすれば、ピクチャは **b 5 . 1** と指定します。なお、ピクチャは省略できません。

パック形式では、整数部桁数+小数部桁数を奇数にしておくのが通例です。

BCD形式では、整数部桁数+小数部桁数を偶数にしておくのが通例です。

整数部の最上位桁に意味があるのかないのかは、半々の割合です。

出力幅は省略できます。出力幅を省略すると、

符号つきパック形式なら1、符号なしパック形式・BCD形式なら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値が使われます。たとえば、パック形式でピクチャ
 = s 4 . 1、出力幅 = 省略という指定は、出力幅に7を指定したのと同じ意味になります。BCD
 形式でピクチャ = b 5 . 1、出力幅 = 省略という指定は、出力幅に7を指定したのと同じ意味
 になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[2進・Alpha]

ホストの2進形式整数・小数項目を、Windowsの文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト／ピクチャは `i4s4.1` と指定します。なお、2進キャストは省略できません。

出力幅は省略できます。出力幅を省略すると、ピクチャに従って出力されます。

出力幅、ピクチャ共に省略すると、

入力のバイト数	1	2	3	4	5	6	7	8
出力幅	3	5	8	10	13	15	17	18

の要領で2進キャストから自動的に計算された値が使われます。例を示すと、入力2進キャスト=`i4s`、出力幅=省略という指定は、出力幅=`10`という指定と同じです。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

(6) 引用符はずし／くくりの指定

ホストがUnix、Windowsの場合、デリミタ形式からの変換の場合には、項目が引用符(“ ” や ‘ ’) でくくられていることがあります。その引用符は無視します。また、引用符でくくったなかにデリミタがあっても、ただの文字データとして扱います。

デリミタ形式に変換するときは、変換後の文字列を引用符でくくることができます。ただし、プリント形式への変換のときは、この引用符くくりの機能は無効になります。

引用符にはつぎの3種類があります。

- | | |
|------|----------------------------|
| 両くくり | 一つの項目を単独でくくる |
| 左くくり | 複数の項目を一つにくくるときに、引用符の始まりを表す |
| 右くくり | 複数の項目を一つにくくるときに、引用符の終わりを表す |

市販ソフトの入力用には、文字項目だけを引用符でくくることが多いのですが、引用符を使わないソフトもあれば、すべての項目を引用符でくくるソフトもあります。

◆参考...

たとえば、デリミタ形式のうちK3フォーマットと呼ばれるものは、

- | | |
|--------------------|----------------|
| 各項目はコンマで区切る | |
| 数値はそのまま(位取りのコンマ不可) | |
| 文字列は引用符でくくる | というルールになっています。 |

(7) デリミタ検出／挿入の有無の指定

ホストがUnix、Windowsの場合、デリミタ形式からの変換の場合には、デリミタ検出を無効にすることができます。デリミタ検出を無効にすると、その次の項目と合わせて1項目とみなします。

デリミタ形式に変換するときは、デリミタ挿入を無効にすることができます。デリミタを無効にすると、その次の項目と合わせて1項目とみなします。ただし、プリント形式への変換のときは、このデリミタ有無の指定の機能は無効になります。

たとえばつぎのように、デリミタ形式へ変換するときに、ある項目（Ank 20）の前に定数（Data=）を挿入するときなどに、デリミタ挿入を無効にします。

```

      :
Ank   ['Data='] &
Ank   20
      :
```

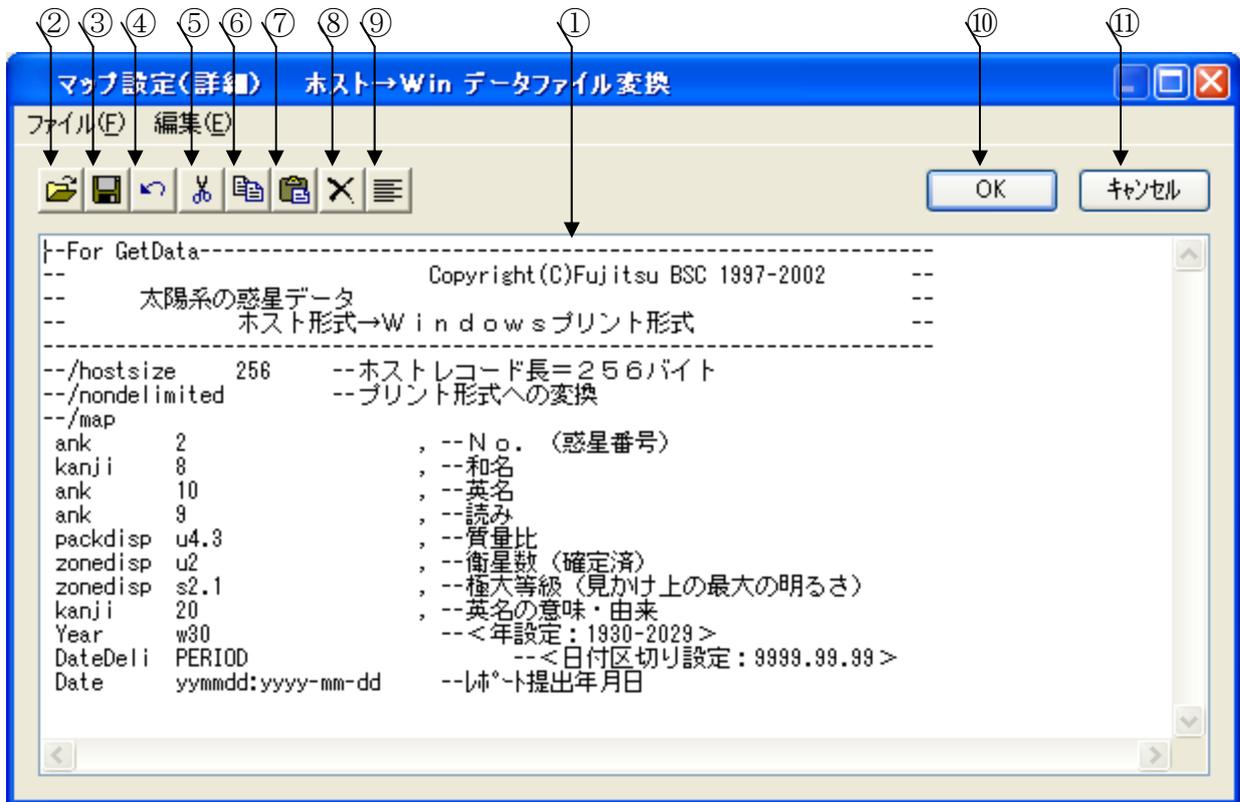
(8) 改行コード挿入の指定

任意のところに改行コードを挿入する（改行コードで項目を区切る）こともできます。改行コード挿入には、つぎの2種類があります。

通常 プリント形式変換時は、指定が無効になる
強制 プリント形式変換時も、指定が有効になる

1レコード変換がおわったときに、自動的にレコード末尾に改行コードが付加される機能とは別のものですから、混同しないでください。

<マップ設定（詳細）ウィンドウ>



①マップ設定の記述をするエディットボックスです。

②～⑨はファイル (F)、編集 (E) のメニューの機能がボタン化されています。

②既存のパラメータファイル等を読みます。

③①の内容をパラメータファイルへ保存します。

④直前の編集作業が無効になり、元に戻ります。

⑤選択した文字列が切り取られ、カットバッファに入ります。

⑥選択した文字列がカットバッファに入ります。

⑦⑤／⑥の操作でカットバッファに入った内容を、カーソルがある位置へ貼りつけます。

カーソルがある位置以降に文字列があれば、挿入になります。

⑧選択した文字列が削除されます。

⑨①の文字列をすべて選択状態にします。

⑩設定が完了したら、OKボタンをクリックします。

⑪キャンセルボタンをクリックすると、すべての入力／編集作業が無効になります。

◆注意 --- マップ設定（詳細）の記述内容について

マップ設定（詳細）のエディットボックスの中に記述する内容は、レコードレイアウト（/map を含まない /map 以降）のみの記述となります。既存のパラメータファイルを読み込むと、それ以外の記述（/delimited～、/map 等）は自動的にコメントになります。記述に誤りがある状態でOKボタンをクリックすると、エラーとなる場合があります。修正してから再度OKボタンをクリックしてください。/map 以降の文法については、「コマンド編」の「コマンド型の実行」の章を参照してください。

例) マップ設定（詳細）にパラメータファイル“PNGETPRN.P”を読み込む

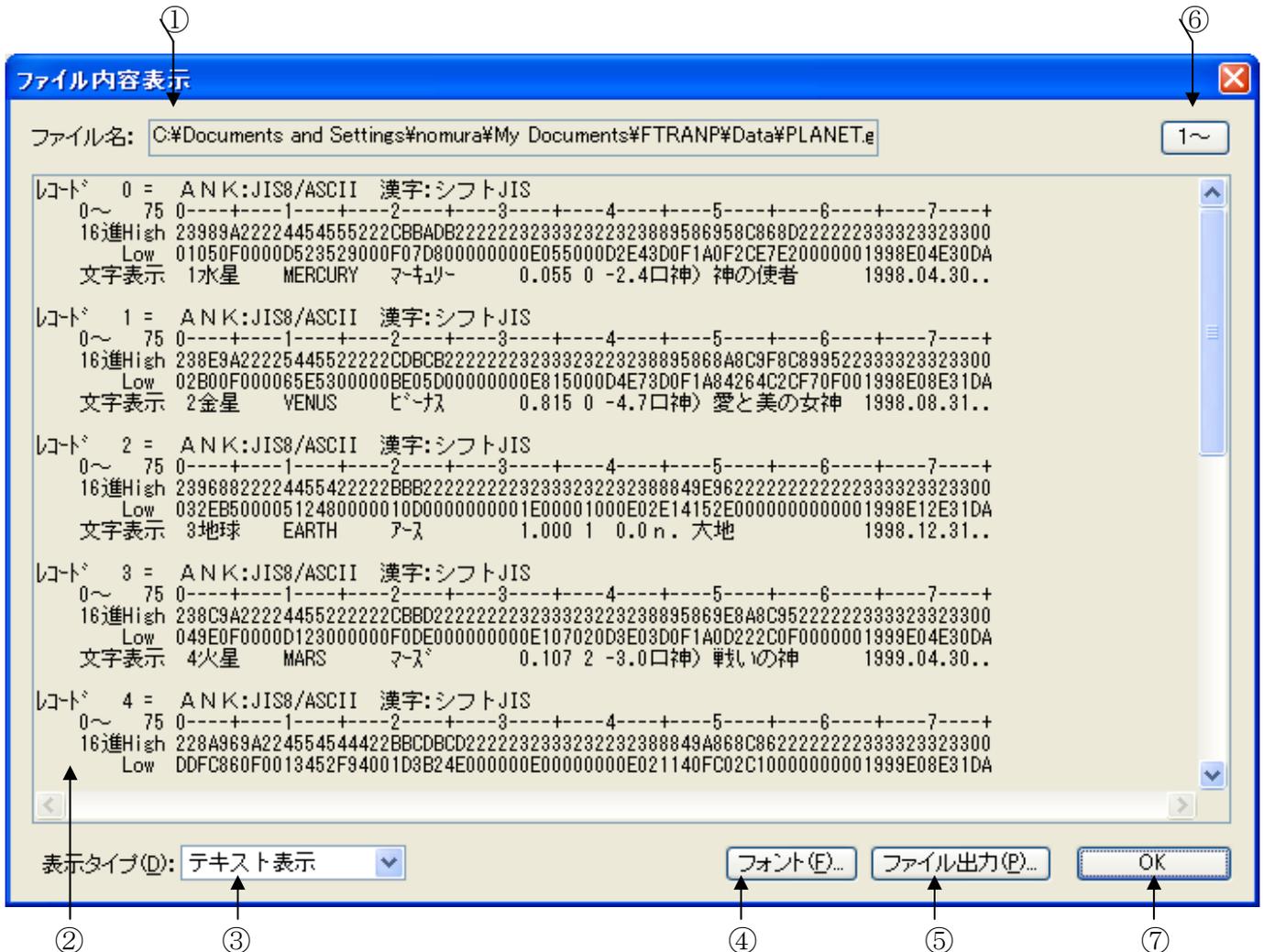
--/hostsize 256		--ホストレコード長=256バイト	} コメントになる行
--/nondelimited		--プリント形式への変換	
--/map			
ank 2		--No. (惑星番号)	} 変換時に、 使われる行
kanji 8		--和名	
ank 10		--英名	
ank 9		--読み	
packd u4.3		--質量比	
zoned u2		--衛星数 (確定済)	
zoned s2.1		--極大等級 (見かけ上の最大の明るさ)	
kanji 20		--英名の意味・由来	
year w30		--<年設定: 1930-2029>	
dated period		--<日付区切り設定: 9999.99.99>	
date yymmdd:yyyy-mm-dd		--レポート提出年月日	

■ホスト→Winデータファイル変換の実行

変換ボタンをクリックすると、ホスト→Winデータファイル変換が始まります。

●ファイル内容表示ウインドウ

変換結果表示 (V) のチェックボックスをONにし、ホスト→Winデータファイル変換を実行すると、実行ウインドウを閉じた後につぎのファイル内容表示ウインドウが開き、変換先のファイルの中身を確認することができます。



- ①変換先のファイル名が表示されます。
- ②変換先のファイルの中身が、③で選択されている表示タイプで出力されます。
- ③②の表示タイプをランダム表示、テキスト表示、テキストのみ表示に切り替えます。
- ④フォント (F) ボタンをクリックして、②の出力文字フォントを切り替えます。
- ⑤ファイル出力 (P) ボタンをクリックして、②の内容をファイルに保存します。
- ⑥桁位置のガイド表示を、0 起点 (先頭を0として数える：デフォルト)、1 起点 (先頭を1として数える) のどちらかに切り替えます。
- ⑦OKボタンをクリックして、ファイル内容表示ウインドウを閉じます。

ファイル内容表示は変換結果のプレビュー表示機能のため、表示できる容量に制限があります。全ての内容が表示されている場合はデータの最後に、

----- ファイルの最後まで表示しました. -----

全ての内容が表示できなかった場合には、

----- ファイルの途中で表示を切りました. -----

と表示されます。

■注意事項

漢字があるときの注意

漢字項目変換やAnk・漢字まじり項目変換を行うときは、あらかじめ

変換設定の漢字変換方式設定で、適当な漢字変換方式を割り当てておく

のを忘れないでください。また、入力幅、出力幅は漢字データについても

バイト単位で指定

します。漢字の文字数ではないことに注意してください。

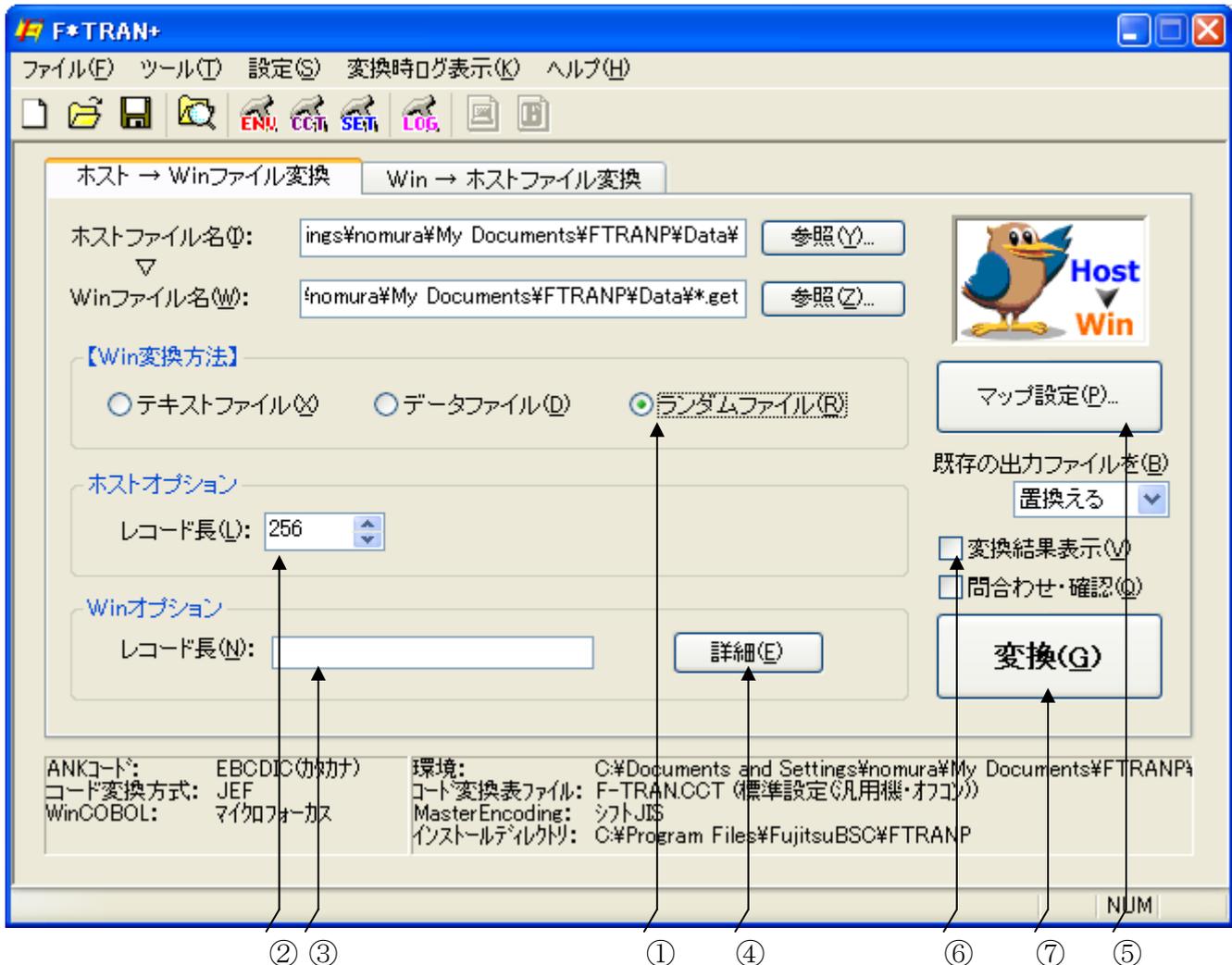
その他の注意事項

「ホスト→Winファイル変換」の節を参照してください。

4.5 ホスト→Winランダムファイル変換

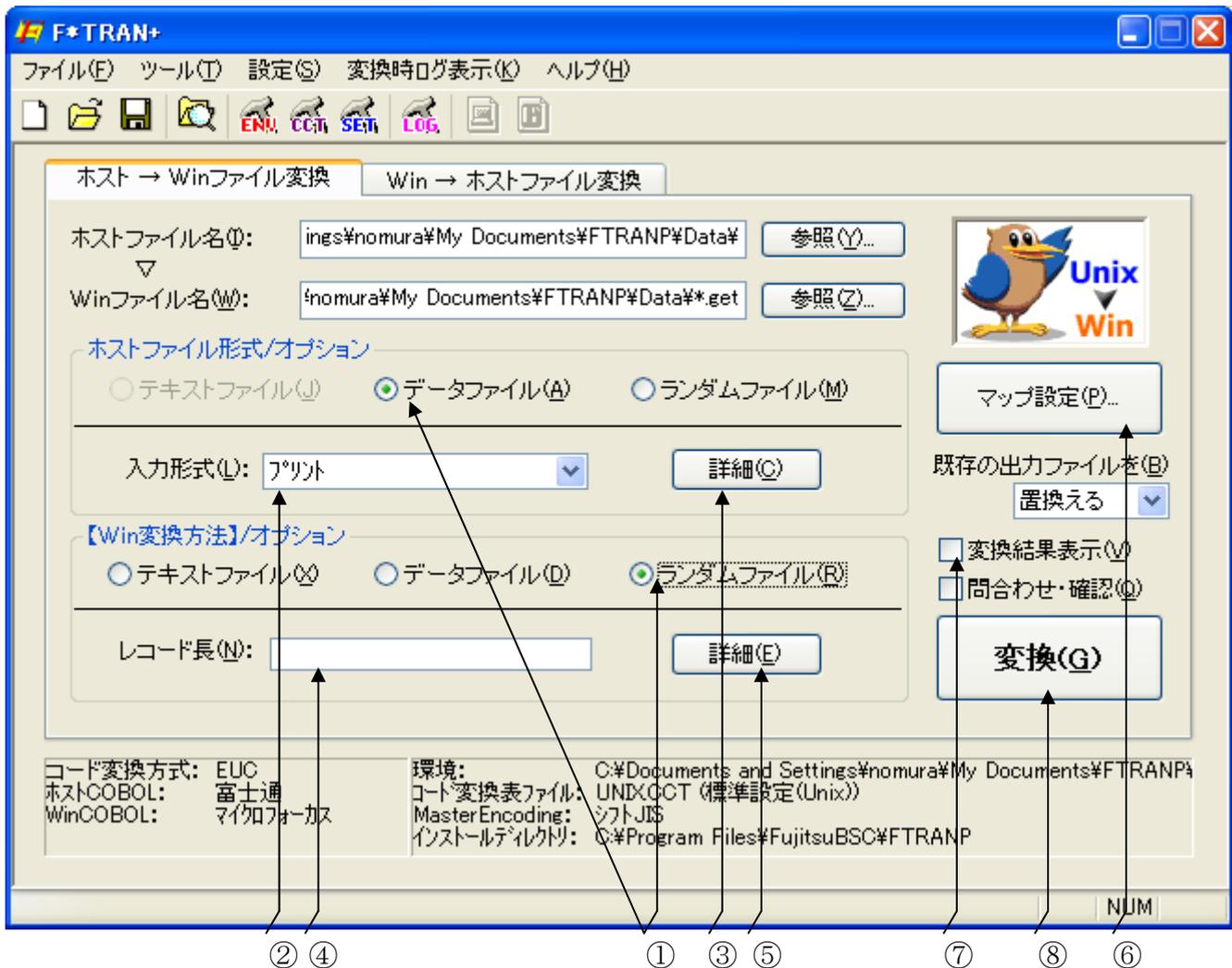
ホスト→Winランダムファイル変換の操作方法について説明します。

●ホストが汎用機・オフコンの場合

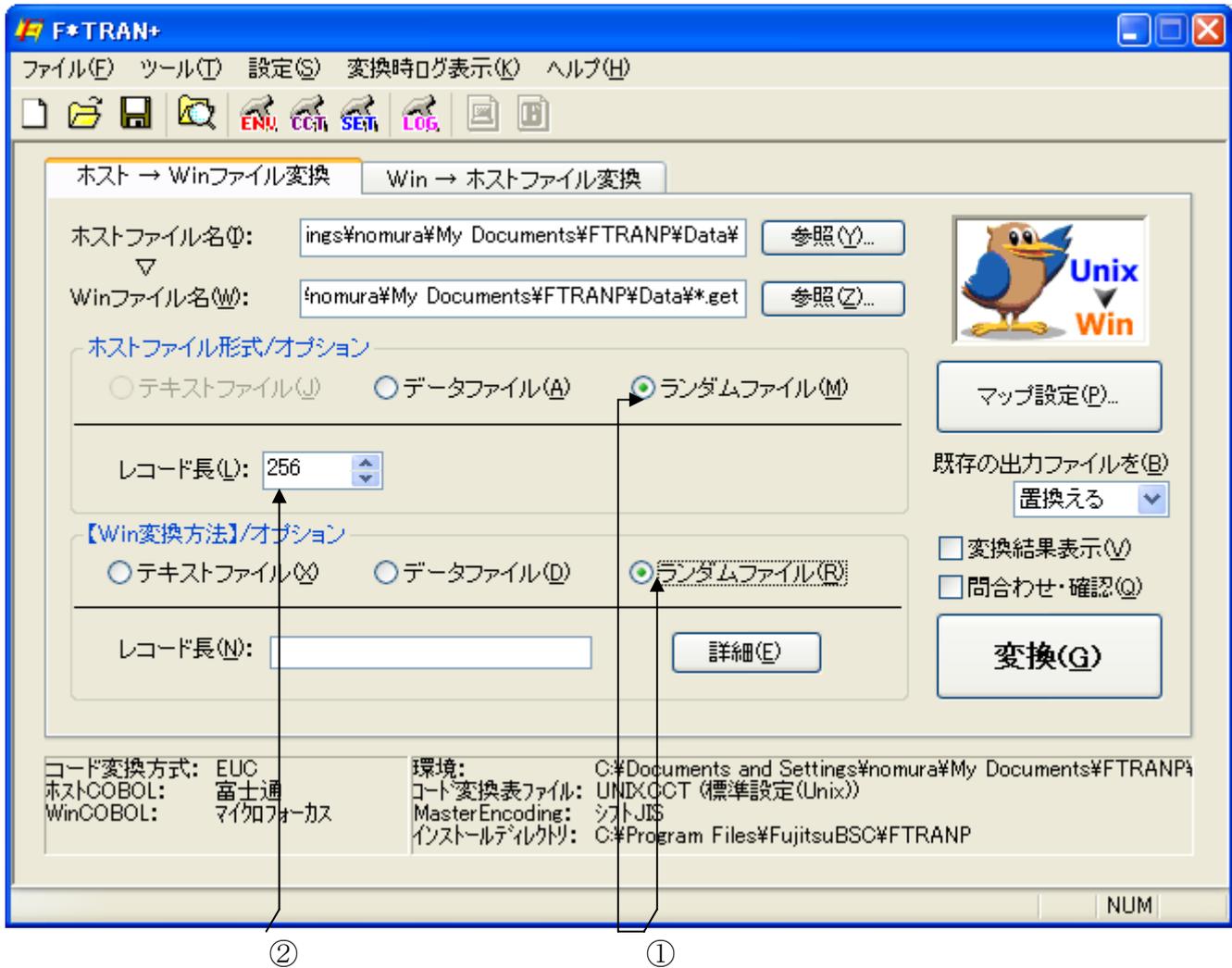


- ①変換方法のランダムファイル (R) をクリックし、ホスト→Winランダムファイル変換を選択します。
- ②ホストファイルのレコード長を指定します。
- ③出力レコード長を指定するオプションです。
- ④Winファイルの詳細オプションボタンです。Windows側エンコーディングの設定ができます。
- ⑤マップ設定 (レコードレイアウト設定) をするオプションボタンです。
- ⑥変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑦変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

●ホストがUnix、Windowsの場合



- ① Win変換方法のランダムファイル (R) をクリックし、ホスト→Winランダムファイル変換を選択します。ホストファイル形式は、データファイル (A)、ランダムファイル (M) の選択ができますが、上の例はホストファイル形式がデータファイル (A) です。
- ② 入力ファイル形式を指定するオプションです。
- ③ ホストファイルの詳細オプションボタンです。タブ拡張、空行無視等の指定ができます。
- ④ 出力レコード長を指定するオプションです。
- ⑤ Winファイルの詳細オプションボタンです。Windows側エンコーディングの設定ができます。
- ⑥ マップ設定 (レコードレイアウト設定) をするオプションボタンです。
- ⑦ 変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑧ 変換を実行するボタンです。このボタンをクリックすると、変換が始まります。



- ① Win変換方法のランダムファイル (D) をクリックし、ホスト→Winランダムファイル変換を選択します。ホストファイル形式は、データファイル (A)、ランダムファイル (M) の選択ができますが、上の例はホストファイル形式がランダムファイル (M) です。
- ② ホストファイルのレコード長を指定します。

■オプションの指定

●ホストファイル・レコード長指定 <ホストファイルが固定長の場合のみ>

ホストファイルのレコード長を1～32767の範囲の10進数で指定します。デフォルトは、**256バイト**です。レコード長の指定が間違っていると正しいデータ変換が行われませんので、この指定は極めて重要です。

●入力形式オプション <ホストファイルが可変長の場合のみ>

変換するホストファイルがプリント形式かデリミタ形式か、デリミタ形式ならコンマ区切り・タブ区切り・スペース区切りのどれなのかを指定します。

プリント ----- プリント形式を変換する
 デリミタ (コンマ) ----- デリミタ形式 (コンマ) を変換
 デリミタ (タブ) ----- デリミタ形式 (タブ) を変換
 デリミタ (スペース) ---- デリミタ形式 (スペース) を変換

プリントを指定すると、デリミタ (区切り文字) なしのプリント形式のファイル (固定長テキストファイル [SDF形式])。ただし、最終項目だけ可変長でもよい) を変換できます。

デリミタを指定すると、デリミタ形式 (区切り文字つき) のテキストファイルを変換できます。区切り文字の種類によって、さらに細かい形式が決まります。デリミタは、

コンマ区切り形式 (CSV形式、K3形式)
タブ区切り形式 (TAB=09H)
スペース区切り形式 (SP=20H)

の3つのなかから指定できます。

後述のマップ設定オプションで、デリミタ検出=コンマ (,) を指定したところに、上記の区切り文字があるとみなされます。

● E O F 検査指定 <ホストがWindowsで、可変長ファイルの場合のみ>

E O F コード (1 A H) を検査するか否かを指定します。

E O F 検査のチェックボックスを、

ONにすると、E O F コードを検査する指定

OFFにすると、E O F コードを検査しない指定 となります。

E O F 検査指定がONであれば、E O F コードを検査し、E O F コードが現れたら変換を終了します。こちらがデフォルトです。

E O F 検査指定がOFFであれば、E O F コードを検査しません。単なるデータとして扱います。

● タブ拡張のオプション <ホストファイルが可変長の場合のみ>

タブ拡張の有無と、タブ拡張するときのタブ間隔を指定します。タブ拡張とは、TAB (0 9 H) をつぎのタブ位置の直前までの連続スペースに展開することです。

タブ拡張のチェックボックスを、

ONにすると、タブ拡張する指定

OFFにすると、タブ拡張しない指定 となります。

タブ拡張指定がONであれば、タブ拡張します。タブ間隔は、2～255の範囲で指定し、それがレコードのおわりまで繰り返し適用されます。**タブ間隔のデフォルトは標準のタブ間隔(8桁きざみ)になります。**これがデフォルトです。

タブ拡張指定がOFFであれば、タブ拡張はしません。

● 空行無視のオプション <ホストファイルが可変長の場合のみ>

空行を無視するかどうかを指定します。

ONにすると、空行を無視する指定

OFFにすると、空行を無視しない指定 となります。

●引用符はずし <ホストがUnix/Windowsで、デリミタ形式の場合のみ>

変換するホストファイルがデリミタ形式の場合には、項目が引用符でくくられていることがあります。その引用符の種類や扱い方を指定します。なお、この設定はデリミタ形式からの変換の場合のみ有効です。

(1) 引用符の種類

変換元のホストファイルで使われている引用符の種類を指定します。

なし 引用符が使われていない時の指定。(もしくは引用符をはずさない指定)

単一引用符 単一引用符「' (半角)」をはずす指定。

2重引用符 2重引用符「" (半角)」をはずす指定。

◆注意 ----- 引用符の種類指定は一種類のみ

ホストファイルに単一引用符(')でくくられた項目と2重引用符(")でくくられた項目が混在していても、どちらか片方の引用符しかはずすことができません。

(2) 引用符項目の取得方法

データの中に引用符を含むのに全体が引用符でくくられていない

,3.5" FDD,

のような例外的な形式のデータの扱いは、ソフトによって解釈がまちまちです。そこで、F*TRAN+では項目取り出し方式を複数用意しておくことで例外的データに対処しています。

【旧F*TRAN風】

- ・ 引用符は必ず対にして使う
- ・ 引用符対は、項目内のどこに何回現れてもよい
- ・ 一对の引用符の中には区切り文字が使われてもよい
- ・ 対の外では、区切り文字をデータとして使えない
- ・ 引用符が2連化してあってもデータとしては認識しない
- ・ 結果的に、すべての引用符が消える

例) ,AAA" B,B" CCC" " DDD,

→

AAAB, BCCDDDD

【新F*TRAN風】

- ・ 基本的には旧F*TRAN風と同様である
- ・ ただし、2連化した引用符をデータとしての引用符として認識する

例) ,AAA” B,B” CCC” ” DDD, → AAAB, BCCC” DDD

【MS-Excel風】

- ◎ 引用符で始まる項目
 - ・ 対になる引用符までは、区切り文字が使われてもよい
 - ・ 対になる引用符までは、2連化した引用符をデータとして扱う
 - ・ 対になる引用符以降は、区切り文字が現れるまで引用符もデータとして扱う
 - ・ 項目内改行はサポートしない
- ◎ 普通の文字で始まる項目
 - ・ 2文字目以降の引用符は、単なるデータとして扱う
 - ・ 区切り文字をデータとして使うことはできない

例) ,3.5” FDD, → 3.5” FDD

【MS-Access風】

- ◎ 引用符で始まる項目
 - ・ 対になる引用符までは、区切り文字が使われてもよい
 - ・ 対になる引用符までは、2連化した引用符をデータとして扱う
 - ・ 対になる引用符以降は、区切り文字が現れるまで削除する
 - ・ 項目内改行はサポートしない
- ◎ 普通の文字で始まる項目
 - ・ 区切り文字をデータとして使うことはできない
 - ・ 2文字目以降に引用符が現れると、それ以降区切り文字までを削除する

例) ,3.5” FDD, → 3.5

◆注意 ---- MS-Excel風とMS-Access風について

MS-Excel風とMS-Access風では、引用符は項目先頭と項目の終わりで各1つずつ存在するのを正常系とし、それ以外については例外系として扱う方式です。両者の違いは、例外系データの扱い方に限られます。入力側データが正常系でありさえすれば、両者に違いは生じないわけです。本マニュアルでのサンプルの例はあくまで例外系のものと認識してください。

◆注意 ---- 「引用符項目の取得方法」の存在について

「引用符の項目の取得方法」はホストがUnix/Windowsで、デリミタ形式の場合に指定可能なオプションです。ホストが汎用機・オフコンの場合には指定できません。

●出力レコード長オプション

Windows側（出力）のレコード長を指定します。Windowsレコード長は、通常、10進数で指定します。式による指定もでき、そのなかでは、

\$ ホスト側（入力）のレコード長 という特殊変数が使用できます。

ホストファイルのレコード長を基準にして、Windowsファイルのレコード長を決めることができます。たとえば、

\$+2 と指定すれば、

「元のホストファイルのレコード長に+2したものをWindows側のレコード長にせよ」という意味になります。

出力レコード長オプションでWindowsレコード長を指定できることは、ホスト→Winランダムファイル変換の性格をよく反映しています。

ゾーン形式とパック形式の変換、日付データの変換をサポートしているため、レコード長を増減する機能がないと困るからです。また、不要な項目がいくつもあるときや、予備領域が多すぎるときはレコード長を縮めたいものです。

●エンコーディング

Windows側のエンコーディングを以下の4通りから選択します。

Master Encodingに従う (詳細は「4.12 環境設定」を参照)
シフトJIS
Unicode (UCS-2)
Unicode (UTF-8)

Unicodeの場合は以下の選択も可能です。

BOMつき
BOMなし
BOM自動

『BOM自動』の指定ではUCS-2ではBOMを付け、UTF-8ではBOMは付けません。

さらにUCS-2の場合は以下の選択も可能です。

Little Endian
Big Endian

◆注意 ---- Windows自身には「レコード長の概念がない」

Windows自体にはファイルごとに登録されたレコード長というものはありません。そのため、かなり自由が利きます。アプリケーションがファイルを扱う論理的な単位をここではWindows側のレコード長、あるいはWindowsレコード長と呼んでいます。

●マップ設定オプション

このマップ設定オプションで、項目別の細かい変換方法を指示します。本来なら、自動的に項目を認識して変換ができると便利です。しかし、ホストの、とくにCOBOLのデータには、データ自身に桁数や小数点位置の判断に必要な情報が含まれていない、という特性があります。そのため自動変換は原理的に不可能なのです。

マップ設定のボタンをクリックすると、つぎのウインドウが開きます。



マップ設定には、

- 簡易設定 —— 設定ウインドウに従って、必要な値を入力するだけで設定ができます。通常は、こちらの機能だけで間に合う場合がほとんどです。
- 詳細設定 —— フリーフォーマットで、マップ設定を記述するエディタモードです。簡易設定だけでは設定できない特殊な設定をする場合、あるいは、既存のパラメータファイル等を流用したい場合等に使用します。マップ設定の文の書き方を覚えなくては設定できません。

の2通りの方法があります。

◆注意 ---- **簡易設定と詳細設定の組み合わせ**

簡易設定で設定した値は、詳細設定で読み込むことができるテキスト形式に直されます。よって、簡易設定をした後の項目に、詳細設定で特殊設定をつけ足すこともできます。

◆注意 ---- **詳細設定→簡易設定の順で設定した場合の制約**

詳細設定→簡易設定の順で設定した場合は、原形が変わってしまう場合があります。これは、詳細設定で設定した項目が簡易設定で読み込まれることにより、スリム化されてしまうからです。ただし、原形が変わってしまっても、設定内容が変わるわけではありません。

◆注意 ---- **入力幅、出力幅の指定について**

入力側や出力側の文字コードによって、入力幅、および出力幅の指定には注意が必要です。

たとえばANKデータを変換する場合、入力がWindowsでエンコーディングがUCS-2の場合は文字数×2の入力幅を指定する必要があります。

出力幅が省略された場合の省略値は入力コードと出力コードから入力幅に指定された値を最小として自動的に計算して求めています。ただし、漢字とANKが混ざった項目に対して漢字項目として変換した場合（ANK文字が全角化される）や、汎用機・オフコンの半角カナをANK変換でUTF-8に変換する場合など、明確に出力幅を指定しないと出力データが途中で切れてしまう場合もあります。

文字コードとしてUnicodeを指定する場合や半角カナ文字がある場合などは、入力幅や出力幅の指定には十分注意する必要があります。

<マップ設定（簡易）ウィンドウ>

The screenshot shows the 'Map Setting (Simple) Window' with a menu bar (File, Edit, Tools) and a toolbar. The main area is divided into 'Data Format' and a table. The 'Data Format' section contains various buttons for selecting data types like Alpha, Ank, Kanji, etc. The table below has columns for No., Host, Data Format, Input, Output, Delimiter, and Comment. Numbered callouts 1-18 point to specific elements: 1-3 point to the 'No.' column, 4 points to the 'Input' column, 5 points to the 'Output' column, 6 points to the 'Delimiter' column, 7 points to the 'Comment' column, 8 points to the 'Host' column, 9 points to the 'Data Format' column, 10 points to the 'Ank' button, 11 points to the 'Kanji' button, 12 points to the 'Ank·Kanji' button, 13 points to the 'OK' button, 14 points to the 'Cancel' button, 15 points to the 'Input Length' field, 16 points to the 'File' menu, and 17 points to the 'Edit' menu.

No.	ホスト名	データ形式	入力	出力	デリミタ	改行	コメント
1							For GetRand-----
2							Copyright (C)Fu
3							太陽系の惑星データ
4							ホスト形式→Windowsラン
5							-----
6							/hostsize256--ホストレコード長=256バイト
7							/size74--Windowsレコード長=74バイト
8							/map
9		Ank	2				No. (惑星番号)
10		漢字	8				和名
11		Ank	10				英名
12		Ank	9				読み
13		パック・表示	u4.3				質量比
14		ゾーン・表示	u2				衛星数 (確定済)
15		ゾーン・表示	s2.1				極大等級 (見かけ上の最大の明るさ)
16		漢字	20				英名の意味・由来
17		年設定	w30				<年設定: 1930-2029>
18		区切記号	年.月.日				<日付区切り設定: 9999.99.99>
19		日付	yyymmdd	yyyy-mm-dd			ホスト提出年月日
20							

- ①No. は、1～2000です。ここは非入力項目で、行選択の時にクリックします。
- ②項目の桁位置等を入力します。通常、項目を組み替えるとき以外は入力しません。
ダブルクリックすると、入力サブウィンドウが開きます。
- ③項目のデータ形式を指定します。ダブルクリックすると、入力サブウィンドウが開きます。
- ④項目の入力幅を入力します。省略すると、残りのバイト数を指定したことになります。
データ形式がゾーン/パック/2進指定の場合、ピクチャまたは2進ピクチャで指定します。
Ank/漢字/Ank・漢字/バイナリ/バイナリ反転の場合、定数の指定もできます。
Alphaへ/表示へ/ゾーンへ/パックへ/2進への場合、数値定数かシステム変数を入力します。
- ⑤項目の出力幅を入力します。出力幅=入力幅の場合は、省略するのがふつうです。
- ⑥ホストがUnix、Windowsの場合、デリミタ形式から変換するときに引用符はずしの指定ができます。
- ⑦ホストがUnix、Windowsの場合、デリミタ形式から変換するときにデリミタ検出を無効にする指定ができます。

⑧改行コードを挿入する指定です。

改行コード挿入を指定すると、変換後の文字列の後に改行コードを挿入します。通常、項目長の増減がない単純な変換の最終項目に指定し、テキストファイル化するのに使います。

⑨コメントの入力ができます。

⑩セルポインタの入力ガイダンスメッセージが出力されるフィールドです。

誤入力をした場合のエラーメッセージも出力されます。

⑪は編集 (E)、ツール (T) のメニューの機能がボタン化されています。



直前の編集作業が無効になり、元に戻ります。



1行または複数行を選択 (①の操作) をした後に、このボタンをクリックすると、選択した項目の内容が切り取られ、カットバッファに入ります。



1行または複数行を選択 (①の操作) をした後に、このボタンをクリックすると、選択した項目の内容がカットバッファに入ります。



カットバッファに入った内容を、セルポインタがある行へ貼りつけます。すでに、セルポインタ以降の行に設定項目がある場合は、挿入になります。



セルポインタがある行以降が1行ずつ下がります。通常は行挿入をする場合に使用します。



選択した項目の内容が削除されます。



変換プレビュー (変換前/変換後のデータ確認) ができます。



Winファイルエディタが起動します。

ホストファイルの中身を見ること (編集も可能) ができます。

⑫セルポインタが選択項目にある場合に、選択肢のボタンが表示されるエリアです。

⑬ホストファイルのレコード長を指定できます。変換プレビューの時の重要項目です。

⑭設定が完了したら、OKボタンをクリックします。

⑮キャンセルボタンをクリックすると、すべての入力/編集作業が無効になります。

⑯マルチレコードの設定ができるAtlasモードに切り替えるボタンです。*1

⑰COBOLのソースファイル (Copy句) を読み込むボタンです。*1

⑱フェーズボタンをクリックすると、フェーズを設定するウィンドウが開きます。*1

*1) 詳細は、マルチレコード編のマニュアルを参照してください。

◆注意 ----- マップ未設定時のデータ変換動作

マップ未設定の場合は、すべてをANKデータとみなしてデータ変換を行います。

<マップ設定（簡易）サブウィンドウ>

ポジションの欄をダブルクリックすると、つぎのサブウィンドウが開きます。入出力桁位置や入出力スキップの設定を簡便に行うことができます。



データ形式の欄をダブルクリックすると、つぎのサブウィンドウが開きます。データ形式から引用符くくりまでの設定を簡便に行うことができます。



<マップ設定（簡易）・変換プレビュー機能>

変換プレビューボタンをクリックすると、マップ設定（簡易）のウィンドウはつぎのような状態になり、変換前／変換後のデータを確認することができます。

(1) ポジション（桁位置）の指定

変換対象にするホスト側（入力）の桁位置や、変換結果を書き込むWindows側（出力）の桁位置を、別の任意の位置に移動できます。現在、処理対象にしている桁位置を、この指定で強制的に変更できます。この機能を利用すると、項目の組み替えなどが簡単に実現できます。

入力桁位置は、ふつう10進数で指定します。式による指定もでき、そのなかでは、

\$ **ホスト側（入力）のレコード長**
. **ホスト側（入力）の現在の桁位置** という特殊変数が使えます。

たとえば、

\$-8 と指定すれば、 **ホスト側（入力）レコード末尾の8バイト前に位置する**
.-40 と指定すれば、 **現在の入力桁位置から40バイト戻る**

という意味になります。

出力桁位置を移動することもできます。ふつう、

:10進数 で桁位置を指定します。

10進数の代わりに式による指定もでき、そのなかでは、

\$ **Windows側（出力）のレコード長**
. **Windows側（出力）の現在の桁位置** という特殊変数が使えます。

◆注意 ---- 先頭を0桁目とする

F*TRAN+では、レコードの先頭を0桁目として数えます。

(2) ポジション（デリミタ位置）の指定

変換対象にするホスト側（入力）データのデリミタ形式の項目位置を、別の任意の項目位置に移動できます。現在、処理対象にしている項目位置を、この機能で強制的に変更できます。この機能を利用すると、デリミタ形式の項目の組み替えなどが簡単に実現できます。

デリミタ位置は、 **@@入力項目番号** で指定します。式による指定もでき、そのなかでは、

- .** ホスト側（入力）の現在の項目位置
- \$** ホスト側（入力）の入力全体の項目数
- *** ホスト側（入力）の残りの項目数 という特殊変数が使えます。

たとえば、

- @@. - 2** と指定すれば、 **今の入力項目から2項目戻る**
- @@\$ - 2** と指定すれば、 **ホスト側（入力）レコード末尾から2項目に位置する**

という意味になります。

◆注意 ---- 先頭を0項目目とする

F*TRAN+では、レコードの先頭項目を0項目目として数えます。

(3) ポジション (入力スキップ) の指定

ホスト側 (入力) レコードに不要な項目があるとき、それをスキップして変換できます。

スキップする幅は、 `^バイト数` で指定します。

たとえば、3バイト分スキップしたいなら、 `^3` と指定します。

バイト数は省略でき、省略すると1バイトとみされるので、

`^3` は `^^^` と指定したのと同じです。

スキップする幅は式による指定もでき、そのなかでは、

* **ホスト側 (入力) の残りバイト数** という特殊変数が使えます。

(4) ポジション (出力スキップ) の指定

入力スキップとは逆に、Windows側 (出力) に何桁か空きを作ることもできます。空項目を作るのがおもな用途です。

スキップする幅は、 `__バイト数` で指定します。

たとえば、3バイト分スキップしたいなら、 `__3` と指定します。

バイト数は省略でき、省略すると1バイトとみされるので、

`__3` は `_____` と指定したのと同じです。

スキップする幅は式による指定もでき、そのなかでは、

* **Windows側 (出力) の残りバイト数** という特殊変数が使えます。

(5) データ形式

[Alpha]

ホスト側のデータに関係なく1バイトごとにANK項目として変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。出力側のコード体系に関係なく、ANKコードの設定に従って1バイトデータに変換されて出力します。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ **ホスト側（入力）のレコード長**
- * **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のANK項目のおわりにスペース（20H）が詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ **Windows側（出力）のレコード長**
- * **Windows側（出力）の残りバイト数** という特殊変数が使えます。

[Ank]

ANK項目を変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ **ホスト側（入力）のレコード長**
- * **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅の代わりにANK形式の定数を指定することもできます。指定した定数はWindows側に指定されたコードに変換されて出力されます。入力値として、つぎの定数が使えます。

文字列定数	文字列はANK形式
汎用定数	Space、LowValue、HighValue

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。入力に定数を指定した場合は、出力幅の省略値は「**見た目の文字数×2**」になります。指定された定数は内部的にUCS-2コードなので、出力コードがUCS-2以外の場合は明示的に出力幅を指定する必要があります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のANK項目のおわりにスペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ Windows側（出力）のレコード長
- * Windows側（出力）の残りバイト数 という特殊変数が使えます。

[漢字]

漢字項目を変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します（漢字の文字数ではありません）。式による指定もでき、そのなかでは、

- \$ ホスト側（入力）のレコード長
- * ホスト側（入力）の残りバイト数 という特殊変数が使えます。

入力幅の代わりに漢字形式の定数を指定することもできます。指定した定数はWindows側に指定されたコードに変換されて出力されます。入力値として、つぎの定数が使えます。

文字列定数	文字列は漢字形式
汎用定数	Space

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。入力に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を大きくしなければならない場合があります。拡張漢字のクエスチョン変換を使っていて、オーバーフローの危険があるときなどです（最大で入力幅の3倍）。そのときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、漢字項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、Windows側の漢字項目のおわりに漢字変換方式で設定されている漢字スペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ Windows側（出力）のレコード長
- * Windows側（出力）の残りバイト数 という特殊変数が使えます。

[ANK・漢字]

ANK・漢字まじり項目を変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。ホストが汎用機・オフコンの場合、漢字の前後にKI/KOがついていないと、この変換方法は適用できません。変換後、KI/KOが取れて、その分左詰めにされます。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ ホスト側（入力）のレコード長
- * ホスト側（入力）の残りバイト数 という特殊変数が使えます。

入力幅の代わりにANK・漢字まじり形式の定数を指定することもできます。指定した定数はWindows側に指定されたコードに変換されて出力されます。入力値としてつぎの定数が使えます。

文字列定数	文字列はANK・漢字まじり形式
汎用定数	Space、LowValue、HighValue

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。入力に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を変更したい場合があります。KI/KOが取れて短くなることや、逆に、拡張漢字のクエスチョン変換を使っていてオーバーフローの危険があるときなどです（最大で入力幅の3倍-KI・KOのバイト数の合計）。そのときは、出力幅を指定します。

項目長を縮めると、ANK・漢字まじり項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、Windows側のANK・漢字まじり項目のおわりにスペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ Windows側（出力）のレコード長
- * Windows側（出力）の残りバイト数 という特殊変数が使えます。

[数値]

文字形式の数値項目どうしの変換をします。

入力幅は、バイト数で指定します。省略すると、入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

数値変換は、Ank項目変換と後述のゾーン・表示変換の中間的なものです。Ank項目変換と比較すると、

文字形式数値しか通さない

入力幅を省略すると15桁とみなす

右詰めになる

などの点が異なります。

「文字形式数値しか通さない」というのは、具体的には、

+、-、0~9、ピリオド(.)、E、e、D、d

しか変換しないで、これら以外の文字は捨ててしまうということです。たとえば、通貨記号(¥/\$)や位取りのコンマ(,)などは削除されるので、リストファイルから入力データファイルを作るときなどに利用できます。

[ゾーン・表示]

ホストのCOBOLのゾーン形式数値項目を、文字形式数値項目に変換します。変換結果は右詰めになります。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする

+整数部桁数

+1+小数部桁数(小数部があれば)

の要領でピクチャから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。たとえば、ピクチャ=s4.1、出力幅=省略という指定で出力コードがUCS-2以外の場合、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[パック・表示]

ホストのCOBOLのパック形式数値項目、BCD形式数値項目を、文字形式数値項目に変換します。変換結果は右詰めになります。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
— 1 2 3 . 4 という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、ピクチャは **s 4 . 1** と指定します。**1 2 3 . 4** という数字が3バイトのBCD形式の項目に記録されているとすれば、ピクチャは **b 5 . 1** と指定します。なお、ピクチャは省略できません。

パック形式では、整数部桁数+小数部桁数を奇数にしておくのが通例です。

BCD形式では、整数部桁数+小数部桁数を偶数にしておくのが通例です。

整数部の最上位桁に意味があるのかないのかは、半々の割合です。

出力幅は省略できます。出力幅を省略すると、

符号つきパック形式なら1、符号なしパック形式・BCD形式なら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。たとえば、パック形式でピクチャ=s 4 . 1、出力幅=省略という指定で出力コードがUCS-2以外の場合は、出力幅に7を指定したのと同じ意味になります。BCD形式でピクチャ=b 5 . 1、出力幅=省略という指定で出力コードがUCS-2以外の場合は、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[バイナリ]

バイナリ変換（無変換）は、「コード変換を一切しない」という変換方法です。バイナリ変換には、Alpha項目変換の説明がほぼそのまま当てはまります。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$** **ホスト側（入力）のレコード長**
- *** **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

また、入力幅の代わりにバイナリ形式の定数を指定することもできます。その場合、指定した定数はコード変換されずに、そのまま挿入されます。入力値として、つぎの定数が使えます。

16進列定数、汎用定数（LowValue、HighValue）

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなされます。入力に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、バイナリ項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のバイナリ項目のおわりにNUL（00H）が詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$** **Windows側（出力）のレコード長**
- *** **Windows側（出力）の残りバイト数** という特殊変数が使えます。

Alpha項目変換の説明が当てはまると書きましたが、実際の運用はかなり違ったものになります。とくに、レコード全体をバイナリ変換することが多いことです。このとき、レコード長が変わらないように出力レコード長は\$を指定します。また、レコードの一部をバイナリ項目として扱うこともあります。

[ユーザーA], [ユーザーB]

ユーザーA/B変換は、利用者独自のバイト単位の変換処理が必要なときに、ユーザーA/B変換表を書き替えて利用します。ユーザーA/B変換には、Alpha項目変換の説明がほとんどそのまま当てはまります。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$** **ホスト側（入力）のレコード長**
- *** **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ユーザーA/B項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のユーザーA/B項目のおわりにNULが詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ Windows側(出力)のレコード長
- * Windows側(出力)の残りバイト数 という特殊変数が使えます。

[ゾーン・ゾーン]

ホストのCOBOLのゾーン形式数値項目を、Windows COBOLのゾーン形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、

—123.4 という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、入力ピクチャは s4.1 と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s4.1という指定は、入力ピクチャ=s4.1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s4.1、出力ピクチャ=省略という指定は、出力ピクチャ=s4.1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[パック・パック]

ホストのCOBOLのパック形式数値項目、BCD形式数値項目を、Windows COBOLのパック、およびBCD形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、

—123.4 という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、入力ピクチャは s4.1 と指定します。123.4 という数字が3バイトのBCD形式の項目に記録されているとすれば、入力ピクチャは b5.1 と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s4.1という指定は、入力ピクチャ=s4.1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s4.1、出力ピクチャ=省略という指定は、出力ピクチャ=s4.1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[ゾーン・パック]

ホストのCOBOLのゾーン形式数値項目を、Windows COBOLのパック、およびBCD形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**— 1 2 3. 4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、入力ピクチャは **s 4. 1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s 4. 1という指定は、入力ピクチャ=s 4. 1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s 4. 1、出力ピクチャ=省略という指定は、出力ピクチャ=s 4. 1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[パック・ゾーン]

ホストのCOBOLのパック形式数値項目、BCD形式数値項目を、Windows COBOLのパック形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**— 1 2 3. 4** という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、入力ピクチャは **s 4. 1** と指定します。**1 2 3. 4** という数字が3バイトのBCD形式の項目に記録されているとすれば、入力ピクチャは **b 5. 1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s 4. 1という指定は、入力ピクチャ=s 4. 1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s 4. 1、出力ピクチャ=省略という指定は、出力ピクチャ=s 4. 1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[年設定]

日付データ項目を変換する際の、年の2桁（yy）と4桁（yyyy）の変換方式を設定します。Wnn、または、Snnの形式で設定します。Wnnの“W”はウインドウ方式を、Snnの“S”はシフト方式を意味し、“nn”は00～99の数字で指定します。

入力側に **W30** と指定すれば、入力データの年を1930～2029年とみなし、出力側に **S25** と指定すれば、出力データの年下2桁を、-25します。

また、シフト方式（“Snn”指定）では、つぎの特殊指定ができます。

SShowa は、“S25”の指定と同じ（昭和通年方式）

SHeisei は、“S88”の指定と同じ（平成通年方式）

年設定は日付項目変換が実行された時に適用になり、複数の年設定がなされている場合は、日付項目変換の直前の年設定が有効になります。

年設定がない場合の日付項目変換のデフォルトは、入力／出力ともに **W30** となり、入出力データの年を1930～2029年とみなします。

[区切記号]

日付データ項目を出力する際の日付区切り記号を、つぎの3つの中から設定します。入力に、SLASH、または、HYPHEN、または、PERIODと入力します。

<u>指 定 文 字</u>	<u>日付区切り記号</u>	<u>デ ー タ 例</u>
SLASH	／（スラッシュ）	1998／12／31
HYPHEN	－（ハイフン）	1998－12－31
PERIOD	．（ピリオド）	1998．12．31

日付区切り設定は日付項目変換が実行された時に適用され、複数の日付区切り設定がなされている場合は、日付項目変換の直前の日付区切り設定が有効になります。

日付区切り設定がない場合の日付項目変換のデフォルトは、**SLASH** となり、日付区切り記号を“／”にします。

[日付]

日付データ項目を変換します。

入力と出力は、必ず日付マスクの形式で指定し、省略はできません。日付マスクの指定方法については、「操作の基礎」の章で説明しました。たとえば、

入力側に、 `yy m m d d`
出力側に、 `yy yy - mm - d d` のように指定すると、

コード変換後に、入力側6桁の日付データ項目を、出力側10桁の日付データ項目に編集します。その際に、年設定、日付区切り設定が適用になります。

日付マスクにはオプションが指定可能です。入力、または出力コードがUCS-2の時、-Disp指定では1桁2バイト単位で処理します。それ以外の場合は1桁1バイト単位で処理します。これが省略値です。-Alpha指定ではコードに関係なく1桁1バイト(ゾーン形式)で処理します。

[表示・ゾーン]

ホストの文字形式数値項目を、Windows COBOLのゾーン形式数値項目に変換します。通常は、ホストがUnixまたはWindowsの場合に使います。

入力幅は、バイト数で指定します。省略すると入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という数字を5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは `s4.1` と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が省略値を超えるときは、

入力幅=40、ピクチャ=u15.2 のように、ダミーの入力幅を指定します。

[表示・パック]

ホストの文字形式数値項目を、Windows COBOLのパック形式数値項目に変換します。通常は、ホストがUnixまたはWindowsの場合に使用します。

入力幅は、バイト数で指定します。省略すると入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を3バイトの符号つきパック形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が省略値を超えるときは、

入力幅=40、ピクチャ=u15.2 のように、ダミーの入力幅を指定します。

[表示・2進]

ホストの文字形式数値項目を、Windowsの2進形式整数・小数項目に変換します。

入力幅は、バイト数で指定します。省略すると入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

まとめると、プリント形式からの変換の場合は

入力幅=10、2進キャスト/ピクチャ=i4s4.1 のような指定になり、

デリミタ形式からの変換の場合は

入力幅=省略、2進キャスト/ピクチャ=i4s4.1 のような指定になるのがふつうです。ただし、入力幅が省略値を超えるときは、

入力幅=40、2進キャスト/ピクチャ=i8u15.2 のように、ダミーの入力幅を指定します。

[2進・表示]

ホストの2進形式整数・小数項目を、Windowsの文字形式数値項目に変換します。変換結果は右詰めになります。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

出力幅は省略できます。出力幅を省略すると、ピクチャに従って出力されます。

出力幅、ピクチャ共に省略すると、

入力のバイト数	1	2	3	4	5	6	7	8
出力幅	3	5	8	10	13	15	17	18

の要領で2進キャストから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。例を示すと、入力2進キャスト=i4s、出力幅=省略という指定は、出力コードがUCS-2の場合、出力幅=20という指定と同じです。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[ゾーン・2進]

ホストのCOBOLのゾーン形式数値項目を、Windowsの2進形式整数・小数項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、入力ピクチャは **s4.1** と指定します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

入力ピクチャも2進キャストにつづくピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、入力ピクチャ=省略、2進キャスト/ピクチャ=i4s4.1という指定は、入力ピクチャ=s4.1を指定したのと同じ意味になります。

[パック・2進]

ホストのCOBOLのパック、およびBCD形式数値項目を、Windowsの2進形式整数・小数項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、入力ピクチャは **s4.1** と指定します。

2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの項目に記録するとすれば、2進キャスト／ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

入力ピクチャも2進キャストにつづくピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、入力ピクチャ=省略、2進キャスト／ピクチャ=**i4s4.1**という指定は、入力ピクチャ=**s4.1**を指定したのと同じ意味になります。

[2進・ゾーン]

ホストの2進形式整数・小数項目を、Windows COBOLのゾーン形式数値項目に変換します。

2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト／ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を5バイトの項目に記録するとすれば、ピクチャは **s4.1** と指定します。

2進キャストにつづくピクチャも出力ピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、2進キャスト／ピクチャ=**i4s4.1**、出力ピクチャ=省略という指定は、出力ピクチャ=**s4.1**を指定したのと同じ意味になります。

[2進・パック]

ホストの2進形式整数・小数項目を、Windows COBOLのパック、BCD形式数値項目に変換します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を3バイトの項目に記録するとすれば、ピクチャは **s4.1** と指定します。

2進キャストにつづくピクチャも出力ピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、2進キャスト/ピクチャ=**i4s4.1**、出力ピクチャ=省略という指定は、出力ピクチャ=**s4.1**を指定したのと同じ意味になります。

[2進・2進]

ホストの2進形式整数・小数項目を、Windowsの2進形式整数・小数項目項目に変換します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目とする場合、2進キャスト/ピクチャは **i4s4.1** と指定します。

2進キャスト/ピクチャは入力または出力のどちらかを省略できます。入力を省略すると「出力と同じ」とみなされ、出力を省略すると「入力と同じ」とみなされます。なお、入力2進キャスト/ピクチャと出力2進キャスト/ピクチャを同時に省略することはできません。

[バイナリ反転]

バイナリ反転は、「コード変換を一切せずに、幅分のデータをバイト単位で左右反転する」という変換方法です。2進数値データ(整数、小数、実数)は、ホスト側が正順であるのに対して、Windows側が逆順であることが多いので、2進数値データの内容をそのままバイト単位で左右反転する場合等に使用します。2進・2進変換のような加工機能はありませんが、その分だけ処理が高速です。

幅は、10進のバイト数で指定します。たとえば4バイトの入力データが、16進表現で、

01 AB CD EF であれば、出力データは **EF CD AB 01** になります。

また、幅の代わりに16進列定数を指定することもできます。その場合、指定した16進列定数は左右反転してからWindows側に挿入されます。

[表示へ]

入力した数値定数、またはシステム変数の値を、文字形式数値項目に変換してから、出力側に挿入します。変換結果は右詰めになります。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

入力値は省略できません。

出力幅は省略できます。出力幅を省略すると、「入力した値を正確に出力するために必要な幅」で出力されます。たとえば、入力値 = `[-123.4]`、出力幅 = 省略という指定は出力幅 = 1 (符号) + 3 (整数部桁数) + 1 (小数点) + 1 (小数部桁数) = 6 を指定したのと同じ意味になります。(出力コードが UCS-2 の場合は 12)

[ゾーンへ]

入力した数値定数、システム変数の値を、出力ピクチャに従って `Windows COBOL` のゾーン形式数値項目に変換してから、出力側に挿入します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という定数を入力値として指定して、5 バイトの符号つきゾーン形式の項目に記録するとすれば、出力ピクチャは `s4.1` と指定します。

入力値、出力ピクチャともに省略できません。

[パックへ]

入力した数値定数、システム変数の値を、出力ピクチャに従ってWindows COBOLのパック、およびBCD形式数値項目に変換してから、出力側に挿入します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**—123.4** という定数を入力値として指定して、3バイトの符号つきパック形式の項目に記録するとすれば、出力ピクチャは **s4.1** と指定します。

入力値、出力ピクチャともに省略できません

[2進へ]

入力した数値定数、システム変数の値を、出力ピクチャに従ってWindowsの2進形式整数・小数項目に変換してから、出力側に挿入します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**—123.4** という定数を入力値として指定して、4バイトの符号つき2進形式の項目に記録するとすれば、出力2進キャスト／ピクチャは **i4s4.1** と指定します。

入力値、出力2進キャスト／ピクチャともに省略できません。

[Alphaへ]

入力した数値定数、またはシステム変数の値を、文字形式数値項目に変換してから、出力側に挿入します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

入力値は省略できません。

出力幅は省略できます。出力幅を省略すると、「入力した値を正確に出力するために必要な幅」で出力されます。たとえば、入力値 = [-123.4]、出力幅 = 省略という指定は出力幅 = 1 (符号) + 3 (整数部桁数) + 1 (小数点) + 1 (小数部桁数) = 6 を指定したのと同じ意味になります。

[Ank化]

ホスト側 (入力) 文字列をできる限り ANK (半角) に変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定の ANK コードの設定で決まります (ふつう、セットアップ時に一回だけ行います)。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

\$ **ホスト側 (入力) のレコード長**
***** **ホスト側 (入力) の残りバイト数** という特殊変数が使えます。

入力幅を省略すると * を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のANK項目のおわりにスペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

\$ **Windows側 (出力) のレコード長**
***** **Windows側 (出力) の残りバイト数** という特殊変数が使えます。

[漢字化]

ホスト側（入力）文字列をできる限り漢字（全角）に変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します（漢字の文字数ではありません）。式による指定もでき、そのなかでは、

- \$** **ホスト側（入力）のレコード長**
- *** **ホスト側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を大きくしなければならない場合があります。拡張漢字のクエスチョン変換を使っていて、オーバーフローの危険があるときなどです（最大で入力幅の3倍）。そのときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、漢字項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、Windows側の漢字項目のおわりに漢字変換方式で設定されている漢字スペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$** **Windows側（出力）のレコード長**
- *** **Windows側（出力）の残りバイト数** という特殊変数が使えます。

[Alpha・ゾーン]

ホストの文字形式数値項目を、Windows COBOLのゾーン形式数値項目に変換します。通常は、ホストがUnixまたはWindowsの場合に使用します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が15バイトを超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字を5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が15バイトを超えるときは、

入力幅=20、ピクチャ=u15.2 のように、ダミーの入力幅を指定します。

[Alpha・パック]

ホストの文字形式数値項目を、Windows COBOLのパック、およびBCD形式数値項目に変換します。通常は、ホストがUnixまたはWindowsの場合に使用します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が15バイトを超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字を3バイトの符号つきパック形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が15バイトを超えるときは、

入力幅=20、ピクチャ=u15.2 のように、ダミーの入力幅を指定します。

[Alpha・2進]

ホストの文字形式数値項目を、Windowsの2進形式整数・小数項目に変換します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が15バイトを超えることは少ないので、省略するほうがふつうです。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

まとめると、プリント形式からの変換の場合は

入力幅=10、2進キャスト/ピクチャ=i4s4.1 のような指定になり、

デリミタ形式からの変換の場合は

入力幅=省略、2進キャスト/ピクチャ=i4s4.1 のような指定になるのがふつうです。ただし、入力幅が15バイトを超えるときは、

入力幅=20、2進キャスト/ピクチャ=i8u15.2 のように、ダミーの入力幅を指定します。

[ゾーン・Alpha]

ホストのCOBOLのゾーン形式数値項目を、文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする

+整数部桁数

+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値が使われます。たとえば、ピクチャ=**s4.1**、出力幅=省略という指定は、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[パック・Alpha]

ホストのCOBOLのパック形式数値項目、BCD形式数値項目を、文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**−123.4** という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、ピクチャは **s4.1** と指定します。**123.4** という数字が3バイトのBCD形式の項目に記録されているとすれば、ピクチャは **b5.1** と指定します。なお、ピクチャは省略できません。

パック形式では、整数部桁数+小数部桁数を奇数にしておくのが通例です。

BCD形式では、整数部桁数+小数部桁数を偶数にしておくのが通例です。

整数部の最上位桁に意味があるのかないのかは、半々の割合です。

出力幅は省略できます。出力幅を省略すると、

符号つきパック形式なら1、符号なしパック形式・BCD形式なら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値が使われます。たとえば、パック形式でピクチャ = **s4.1**、出力幅 = 省略という指定は、出力幅に7を指定したのと同じ意味になります。BCD形式でピクチャ = **b5.1**、出力幅 = 省略という指定は、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[2進・Alpha]

ホストの2進形式整数・小数項目を、Windowsの文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト／ピクチャは `i4s4.1` と指定します。なお、2進キャストは省略できません。

出力幅は省略できます。出力幅を省略すると、ピクチャに従って出力されます。

出力幅、ピクチャ共に省略すると、

入力のバイト数	1	2	3	4	5	6	7	8
出力幅	3	5	8	10	13	15	17	18

の要領で2進キャストから自動的に計算された値が使われます。例を示すと、入力2進キャスト = `i4s`、出力幅 = 省略という指定は、出力幅 = 10という指定と同じです。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

(6) 引用符はずしの指定

ホストがUnix、Windowsの場合、デリミタ形式からの変換の場合には、項目が引用符(“ ” や ‘ ’) でくくられていることがあります。その引用符は無視します。また、引用符でくくったなかにデリミタがあっても、ただの文字データとして扱います。

引用符にはつぎの3種類があります。

両くくり	一つの項目を単独でくくる
左くくり	複数の項目を一つにくくるときに、引用符の始まりを表す
右くくり	複数の項目を一つにくくるときに、引用符の終わりを表す

(7) デリミタ検出の有無の指定

ホストがUnix、Windowsの場合、デリミタ形式からの変換の場合には、デリミタ検出を無効にすることができます。デリミタ検出を無効にすると、その次の項目と合わせて1項目とみなします。

たとえばつぎのように、ある項目 (**Ank : 20**) の前に定数 (**Data=**) を挿入するときなどに、デリミタ検出を無効にします。

```

      :
Ank  ['Data='] &
Ank  : 20
      :
```

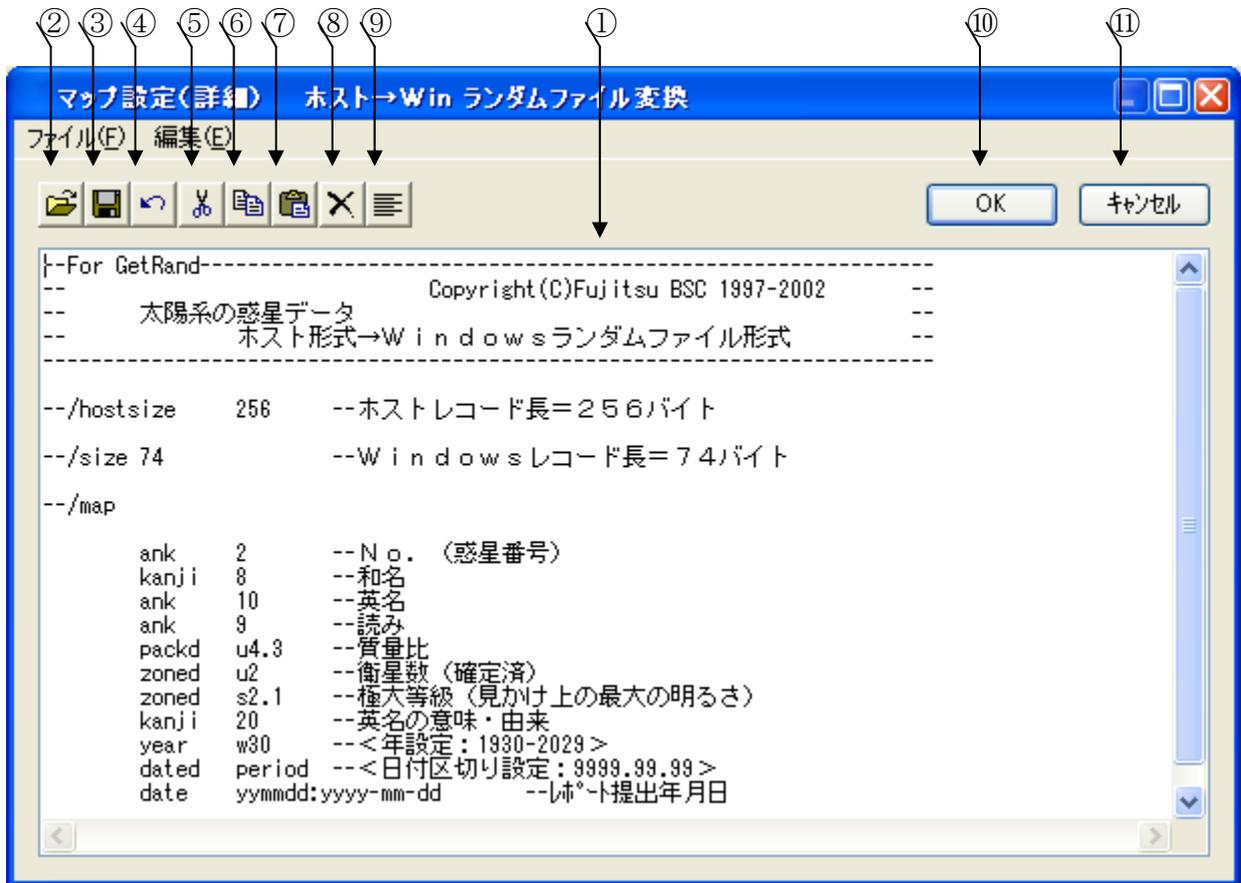
(8) 改行コード挿入の指定

Windows側(出力)レコードに改行コード(CR/LF)を挿入します。改行コード挿入には **通常、強制** の2種類がありますが、ホスト→Winランダムファイル変換では違いはありません。

ふつう、項目長の増減がない単純な変換のときマップ項目の最後に指定し、テキストファイル化するのに使います。そのときは、改行コードの2バイト分Windowsレコード長が増えるので、出力レコード長オプションでその分を\$+2と指定して調整するのを忘れないでください。

変換結果のテキストファイル化には、本来はホスト→Winデータファイル変換のプリント形式への変換機能を使うべきです。これなら、変換後のレコード長をほとんど意識する必要がありません。

<マップ設定（詳細）ウィンドウ>



①マップ設定の記述をするエディットボックスです。

②～⑨はファイル (F)、編集 (E) のメニューの機能がボタン化されています。

②既存のパラメータファイル等を読みます。

③①の内容をパラメータファイルへ保存します。

④直前の編集作業が無効になり、元に戻ります。

⑤選択した文字列が切り取られ、カットバッファに入ります。

⑥選択した文字列がカットバッファに入ります。

⑦⑤／⑥の操作でカットバッファに入った内容を、カーソルがある位置へ貼りつけます。

カーソルがある位置以降に文字列があれば、挿入になります。

⑧選択した文字列が削除されます。

⑨①の文字列をすべて選択状態にします。

⑩設定が完了したら、OKボタンをクリックします。

⑪キャンセルボタンをクリックすると、すべての入力／編集作業が無効になります。

◆注意 --- マップ設定（詳細）の記述内容について

マップ設定（詳細）のエディットボックスの中に記述する内容は、レコードレイアウト（/mapを含まない/map以降）のみの記述となります。既存のパラメータファイルを読み込むと、それ以外の記述（/delimited～、/map等）は自動的にコメントになります。記述に誤りがある状態でOKボタンをクリックすると、エラーとなる場合があります。修正してから再度OKボタンをクリックしてください。/map以降の文法については、「コマンド編」の「コマンド型の実行」の章を参照してください。

例) マップ設定（詳細）にパラメータファイル“PNGETRND.P”を読み込む

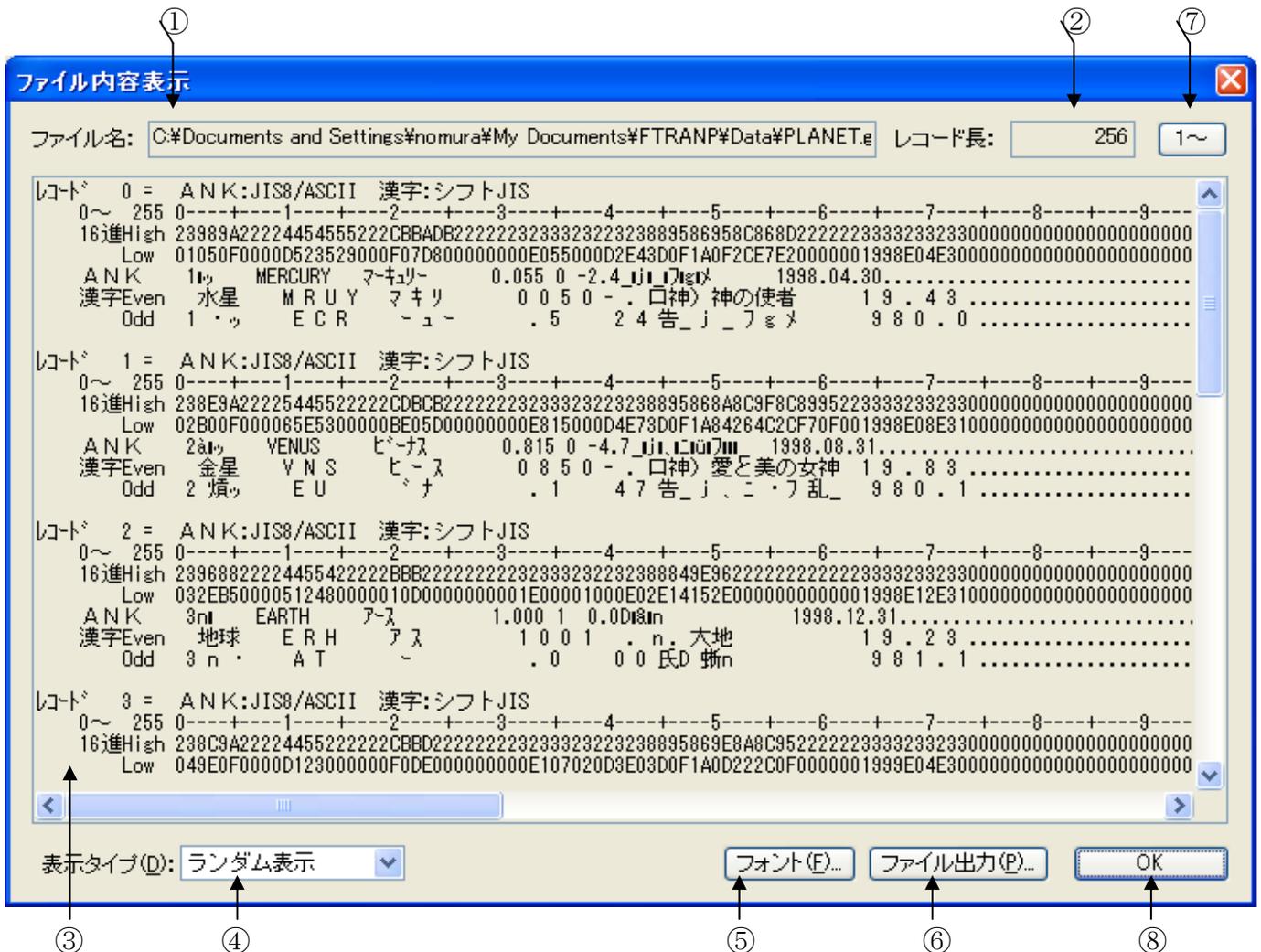
--/hostsize 256	--ホストレコード長=256バイト	}	コメントになる行
--/size 74	--Windowsレコード長=74バイト		
--/map			
ank 2	--No. (惑星番号)	}	変換時に、 使われる行
kanji 8	--和名		
ank 10	--英名		
ank 9	--読み		
packd u4.3	--質量比		
zoned u2	--衛星数 (確定済)		
zoned s2.1	--極大等級 (見かけ上の最大の明るさ)		
kanji 20	--英名の意味・由来		
year w30	--<年設定: 1930-2029>		
dated period	--<日付区切り設定: 9999.99.99>		
date yymmdd:yyyy-mm-dd	--レポート提出年月日		

■ホスト→Winランダムファイル変換の実行

変換ボタンをクリックすると、ホスト→Winランダムファイル変換が始まります。

●ファイル内容表示ウインドウ

変換結果表示 (V) のチェックボックスをONにし、ホスト→Winランダムファイル変換を実行すると、実行ウインドウを閉じた後につぎのファイル内容表示ウインドウが開き、変換先のファイルの中身を確認することができます。



- ①変換先のファイル名が表示されます。
- ②変換先のファイルのレコード長が表示されます。
- ③変換先のファイルの中身が、④で選択されている表示タイプで出力されます。
- ④③の表示タイプをランダム表示、テキスト表示、テキストのみ表示に切り替えます。
- ⑤フォント (F) ボタンをクリックして、③の出力文字フォントを切り替えます。
- ⑥ファイル出力 (P) ボタンをクリックして、③の内容をファイルに保存します。
- ⑦桁位置のガイド表示を、0 起点 (先頭を0として数える：デフォルト)、1 起点 (先頭を1として数える) のどちらかに切り替えます。
- ⑧OKボタンをクリックして、ファイル内容表示ウインドウを閉じます。

ファイル内容表示は変換結果のプレビュー表示機能のため、表示できる容量に制限があります。全ての内容が表示されている場合はデータの最後に、

----- ファイルの最後まで表示しました. -----

全ての内容が表示できなかった場合には、

----- ファイルの途中で表示を切りました. -----

と表示されます。

■注意事項

漢字があるときの注意

漢字項目変換やAnk・漢字まじり項目変換を行うときは、あらかじめ

変換設定の漢字変換方式設定で、適当な漢字変換方式を割り当てておく

のを忘れないでください。また、入力幅、出力幅は漢字データについても

バイト単位で指定

します。漢字の文字数ではないことに注意してください。

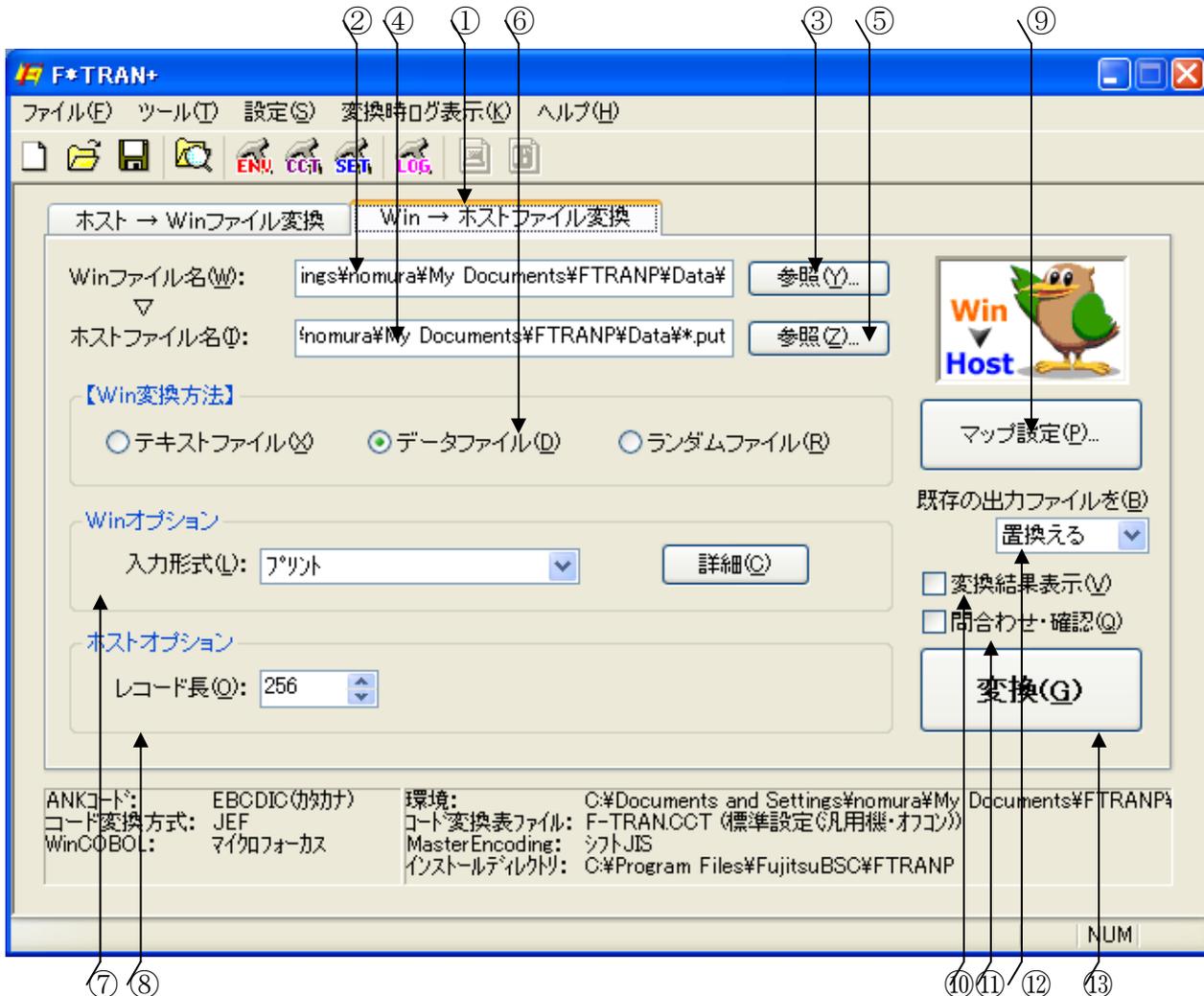
その他の注意事項

「ホスト→Winファイル変換」の節を参照してください。

4.6 Win→ホストファイル変換のファイル指定と共通オプション

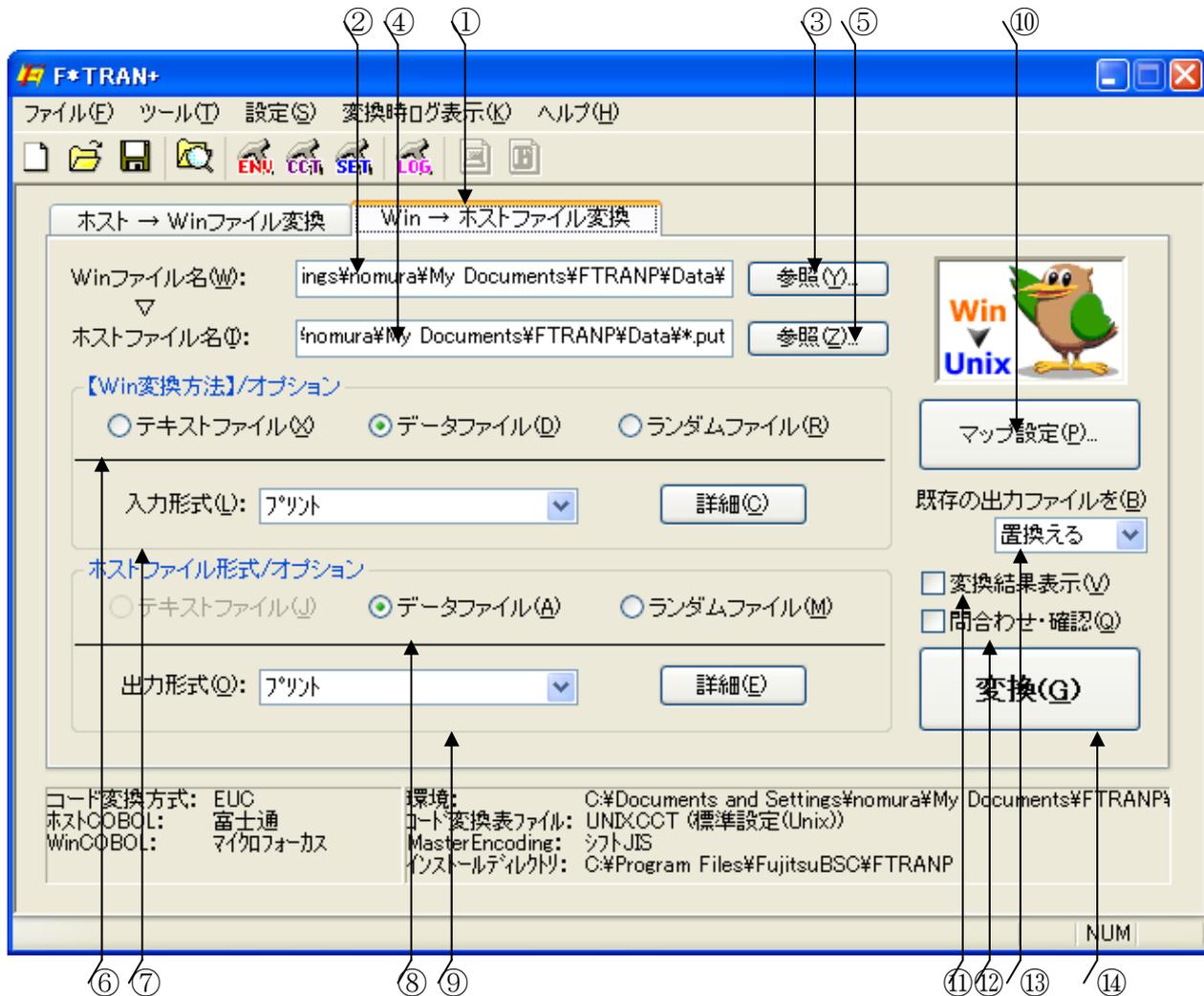
ここでは、Win→ホストファイル変換の共通事項を説明します。

●ホストが汎用機・オフコンの場合



- ① Win→ホストのタブをクリックし、メイン処理部をWin→ホストファイル変換にします。
- ② 変換元のWindowsファイル名を指定します。
- ③ 変換元のWindowsファイル名を決定する参照ウインドウを開くボタンです。
- ④ 変換先のホストファイル名を指定します。
- ⑤ 変換先のホストファイル名を決定する参照ウインドウを開くボタンです。
- ⑥ 変換方法を選択するラジオボタンです。
- ⑦ Winファイルのオプション項目です。⑥の選択によって、オプションの内容が変わります。
- ⑧ ホストファイルのオプション項目です。⑥の選択によって、オプションの内容が変わります。
- ⑨ マップ設定（レコードレイアウト設定）をするオプションボタンです。
- ⑩ 変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑪ 変換実行時の問い合わせ・確認ウインドウ表示の選択をするチェックボックスです。
- ⑫ 変換先ファイルが既に存在した場合の動作を指定するコントロールです。
- ⑬ 変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

●ホストがUnix、Windowsの場合



- ① Win→のホストタブをクリックし、メイン処理部をWin→ホストファイル変換にします。
- ② 変換元のホストファイル名を指定します。
- ③ 変換元のホストファイル名を指定する参照ウインドウを開くボタンです。
- ④ 変換先のWindowsファイル名を指定します。
- ⑤ 変換先のWindowsファイル名を指定する参照ウインドウを開くボタンです。
- ⑥ Winファイルの変換方法を選択するラジオボタンです。変換の基準となる項目です。
- ⑦ Winファイルのオプション設定です。
- ⑧ ホストファイル（出力ファイル）の形式を選択するラジオボタンです。
- ⑨ ホストファイルのオプション設定です。
- ⑩ マップ設定（レコードレイアウト設定）をするオプションボタンです。
- ⑪ 変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑫ 変換実行時の問い合わせ・確認ウインドウ表示の選択をするチェックボックスです。
- ⑬ 変換先ファイルが既に存在した場合の動作を指定するコントロールです。
- ⑭ 変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

■Windowsファイル指定

[d:] [パス名指定] 基本ファイル名 [. 拡張子]

d : はドライブ名

入力側のWindowsファイルを指定します。

ドライブ名は、A : ~ Z : 、@ : 、? : のどれかで指定します。ドライブ名を指定すると、そのドライブを検索します。

ドライブ名は省略可能です。ドライブ名を省略すると、カレントドライブを検索します。

パス名指定 (¥ディレクトリ名¥サブディレクトリ名¥・・・) ができます。指定したディレクトリ配下のファイルを検索します。パス名指定を省略すると、カレントディレクトリを検索します。

基本ファイル名と拡張子には、ワイルドカード文字 (*と?) を使うことができます。ワイルドカード文字を使うと、一致するファイルをすべて検索し変換の対象にします。

まとめると、あるディレクトリのファイルをすべて変換したいなら、

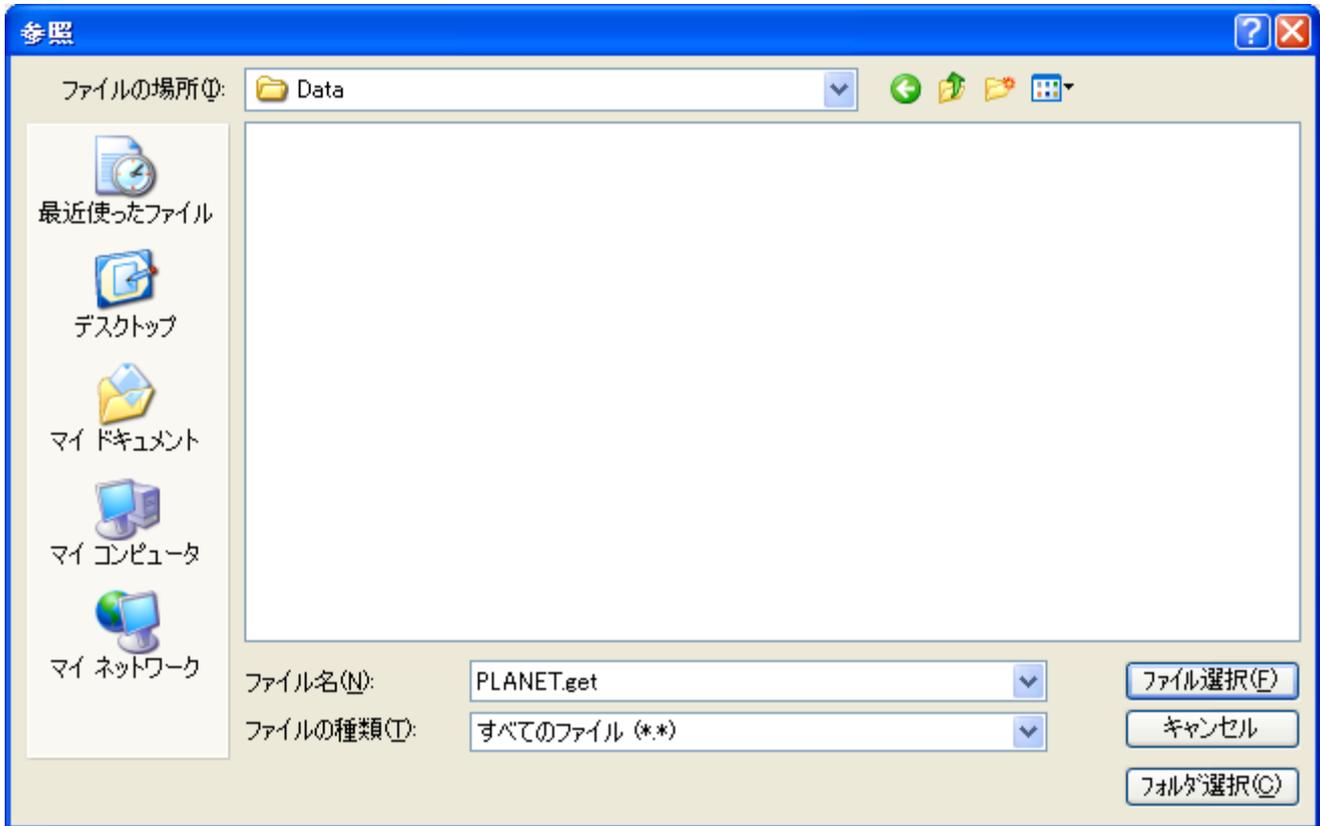
C : * . * のように指定し、

拡張子が . DAT のファイルをすべて変換したいなら、

C : * . DAT のような指定になります。

●Windowsファイル参照ウインドウ

Windowsファイルの参照ボタンをクリックすると、つぎのウインドウが開きます。



変換元のファイルを直接指定する場合は、ファイル名を選択して、ファイル選択 (F) ボタンをクリックします。

変換元の場所 (どこのフォルダのファイルを変換の対象にするか) を指定する場合は、フォルダ選択 (C) ボタンをクリックします。ファイル名にフォルダまでのフルパスが入力され、確定します。

■ホストファイル指定

[d:] [パス名指定] 基本ファイル名 [. 拡張子]

d : はドライブ名

出力側のホストファイルを指定します。

ドライブ名は、A : ~ Z : 、@ : 、? : のどれかで指定します。ドライブ名を指定すると、そのドライブにファイルを作ります。

ドライブ名は省略可能です。ドライブ名を省略すると、カレントドライブにファイルを作ります。

パス名指定 (¥ディレクトリ名¥サブディレクトリ名¥・・・) ができます。指定したディレクトリ配下にファイルを作ります。パス名指定を省略すると、カレントディレクトリにファイルを作ります。

基本ファイル名の部分には、ふつうの基本ファイル名、または*を指定します。*を指定するとWindowsファイルの基本ファイル名がホストファイルの基本ファイル名になります。

拡張子を省略すると、拡張子なしのファイルになります。しかし、拡張子をつけたほうがファイルの管理が容易になるので、なるべく適当な拡張子を指定してください。拡張子に*を指定するとWindowsファイルの拡張子がホストファイルの拡張子になります。

以上がディスクファイルの指定方法です。まとめると、基本ファイル名を引き継ぐときは

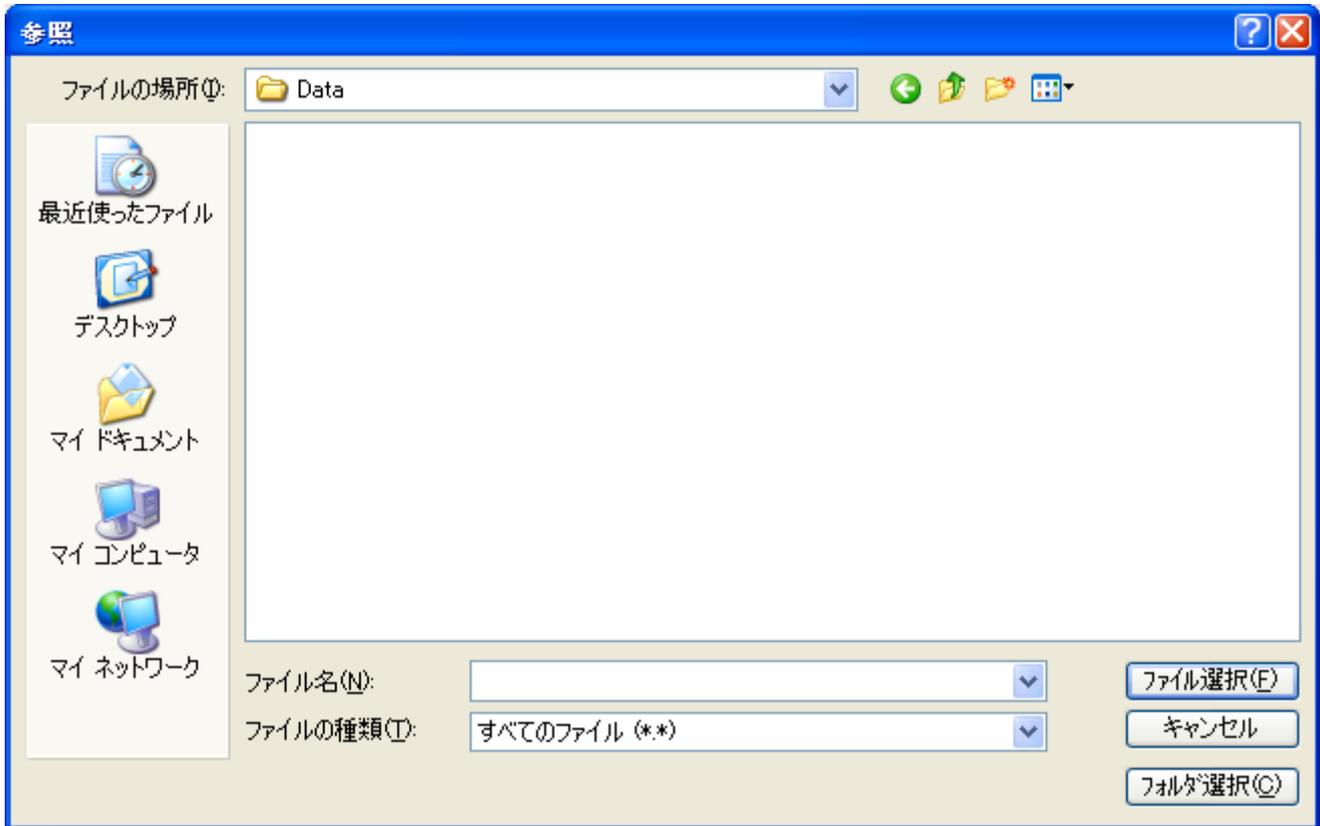
C : *. DAT のような指定になり、

ファイル名をつけ替えるときは

C : NEWNAME. DAT のような指定になります。

●ホストファイル参照ウインドウ

ホストファイルの参照ボタンをクリックすると、つぎのウインドウが開きます。



変換先のファイルを直接指定する場合は、ファイル名を選択して、ファイル選択 (F) ボタンをクリックします。

変換先の場所 (どこのフォルダに変換後のファイルを作成するか) を指定する場合は、フォルダ選択 (C) ボタンをクリックします。ファイル名にフォルダまでのフルパスが入力され、確定します。

■その他の指定

●変換結果表示指定

変換後に、変換先のWindowsファイルの中身を表示ウインドウに出力するかどうかを指定します。出力する場合は、変換結果表示のチェックボックスをONにします。

●問合わせ・確認指定

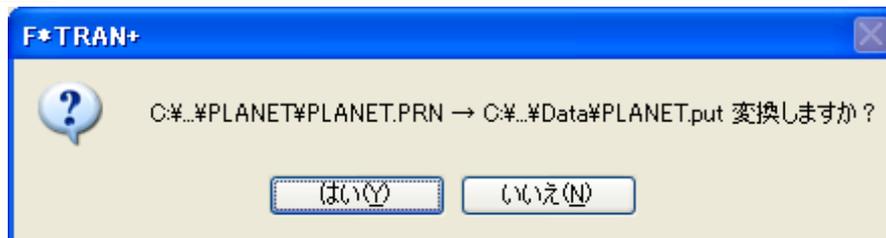
1ファイルごとに処理を問合わせるか否かを指定します。
問合わせ・確認のチェックボックスを、

ONにすると、変換するか否かを問合わせる指定

OFFにすると、ファイル名の確認なしで自動変換する指定 となります。

問合わせ・確認指定がONであれば、ファイル名を確認しながら変換できます。
F*TRAN+は、1ファイルごとに変換を実行するか問わせてきます。
つぎのどれかで応答してください。この機能は、比較的小さいファイルが多数あって、そのうちいくつかを選んで変換したいときなどに、便利です。

1ファイルの変換



はい (Y) 表示中のファイルを変換する
いいえ (N) 表示中のファイルは変換しない

2ファイル以上の変換



はい (Y) 表示中のファイルを変換する
すべて変換 (A) 全ファイル変換に切り替え、以降のファイルをすべて変換する
いいえ (N) 表示中のファイルは変換しない
キャンセル これ以降の変換処理を中断する

問合わせ・確認指定がOFFであれば、ファイル名の確認なしで、自動的に指定のファイルをすべて変換します。こちらがデフォルトです。

●既存の出力ファイルの置き換え指定

変換先ファイルと同名のファイルが同一フォルダ上に存在した場合の動作を以下の3通りから指定します。

◇「置換える」指定時（省略値）

出力先ファイルと同名のファイルが同一フォルダ上に存在した場合は、自動で置き換えます。

◇「置換えない」指定時

出力先ファイルと同名のファイルが同一フォルダ上に存在した場合は以下のメッセージをログに出力し、プログラムは正常終了します。（異常終了ではありません）

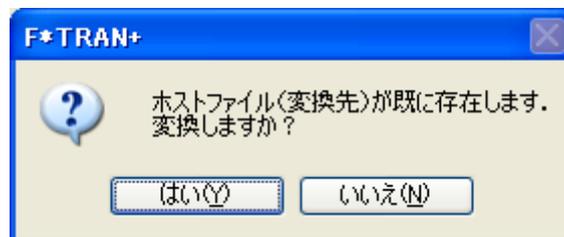
・Win→ホスト変換時のメッセージ

「ホストファイルが既に存在する為、変換できません。」

◇「問合わせる」指定時

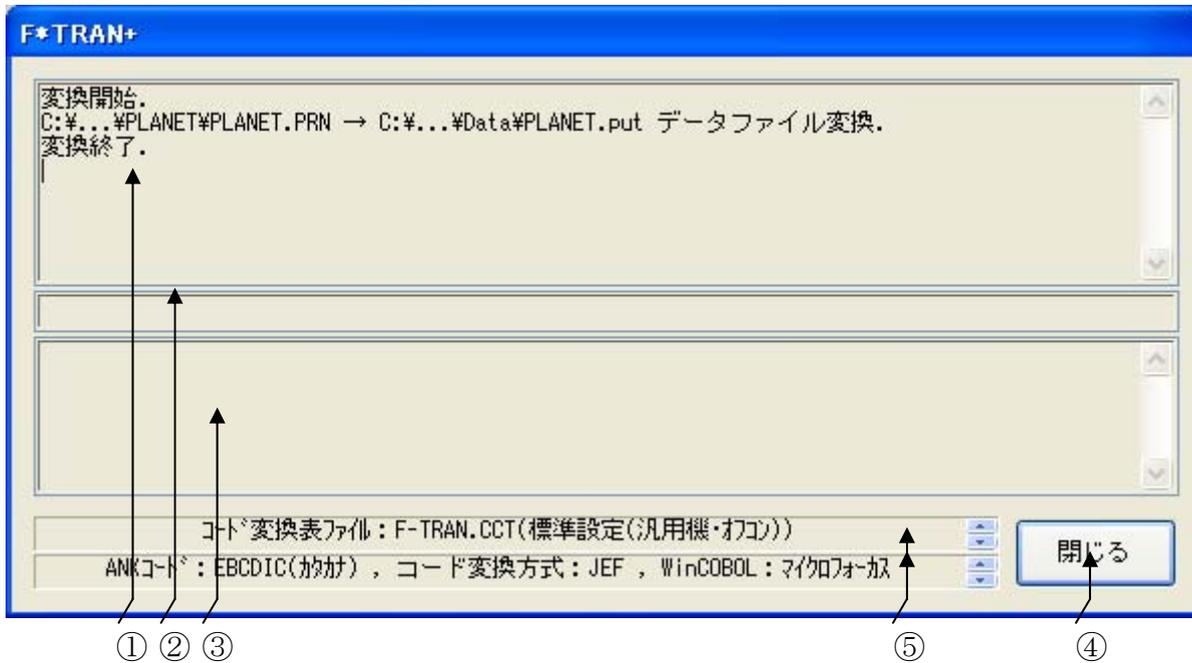
出力先ファイルと同名のファイルが同一フォルダ上に存在した場合は、以下の問い合わせダイアログを表示して、ユーザの指示を待ちます。

【Win→ホスト変換時の問い合わせ画面】



■変換時の実行ウィンドウ

変換ボタンをクリックすると、Win→ホストファイル変換が始まり、つぎの実行ウィンドウが開きます。



①変換中のメッセージを表示するメッセージフィールドです。

変換が始まると、“**変換開始.**”というメッセージ表示につづいて、

“**Windows**ファイル名 → **ホストファイル名** ~**ファイル変換.**”が表示され、

変換が正常終了すると、“**変換終了.**”と表示されます。

②変換中の進行状況（進行度%出力）を表示するフィールドです。

③変換中のエラーメッセージを表示するフィールドです。

④変換中は「中止 (Q)」と表示されています。

変換中にこのボタンをクリックすると、変換を途中で中止します。

変換が終了したら、その変換結果を確認して、“閉じる”ボタンをクリックします。

⑤おもにコード変換に関する現在の設定が表示されています。

■注意事項

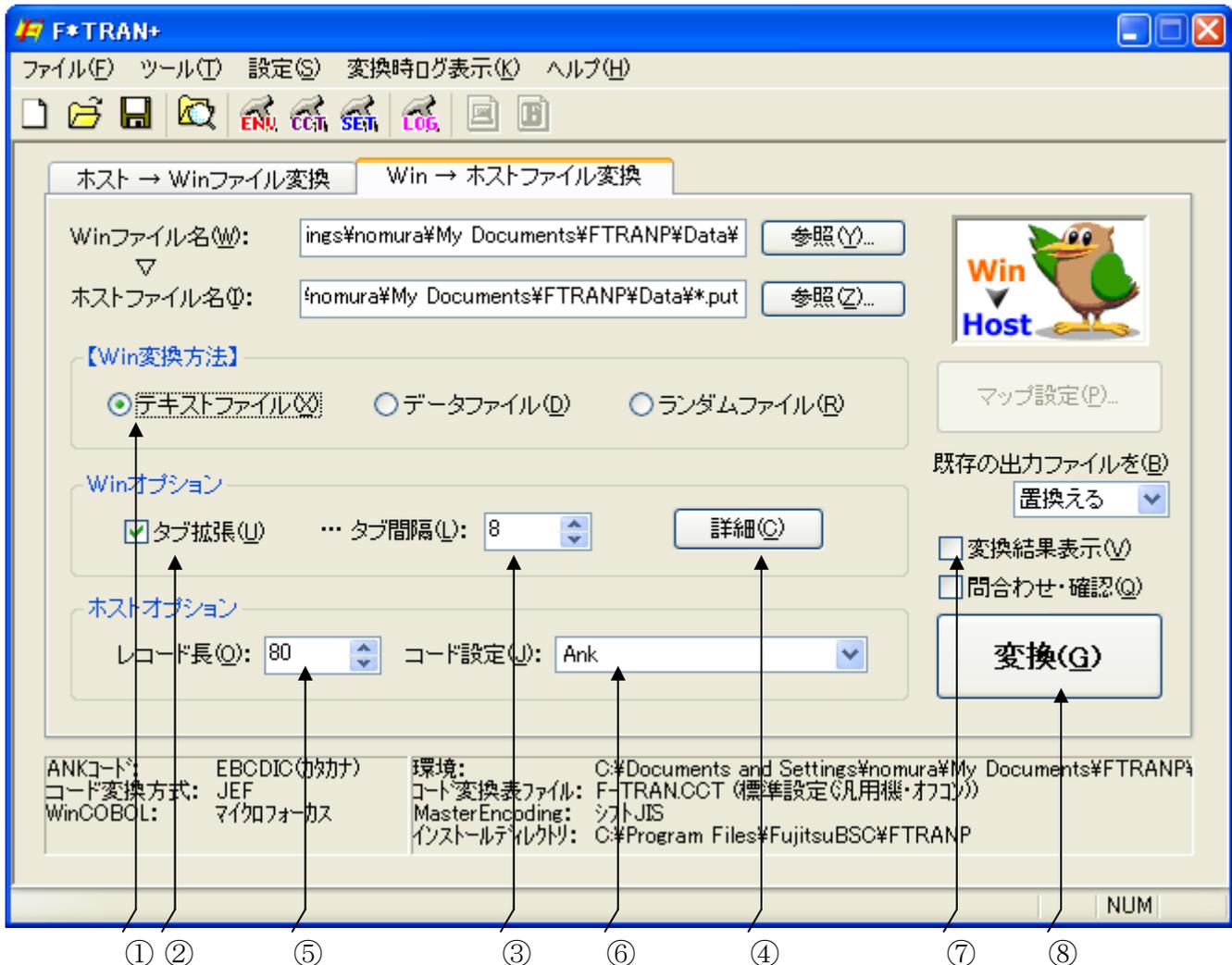
同名ファイルは置換する（デフォルト）

すでに同じ名前のホストファイルがある場合、デフォルトでは自動的に元のファイルを削除し、新たに変換したファイルで置き替えます。この動作は画面上からの指定により変更可能です。詳細は「●既存の出力ファイルの置き換え指定」の説明を参照してください。

4.7 Win→ホストテキストファイル変換

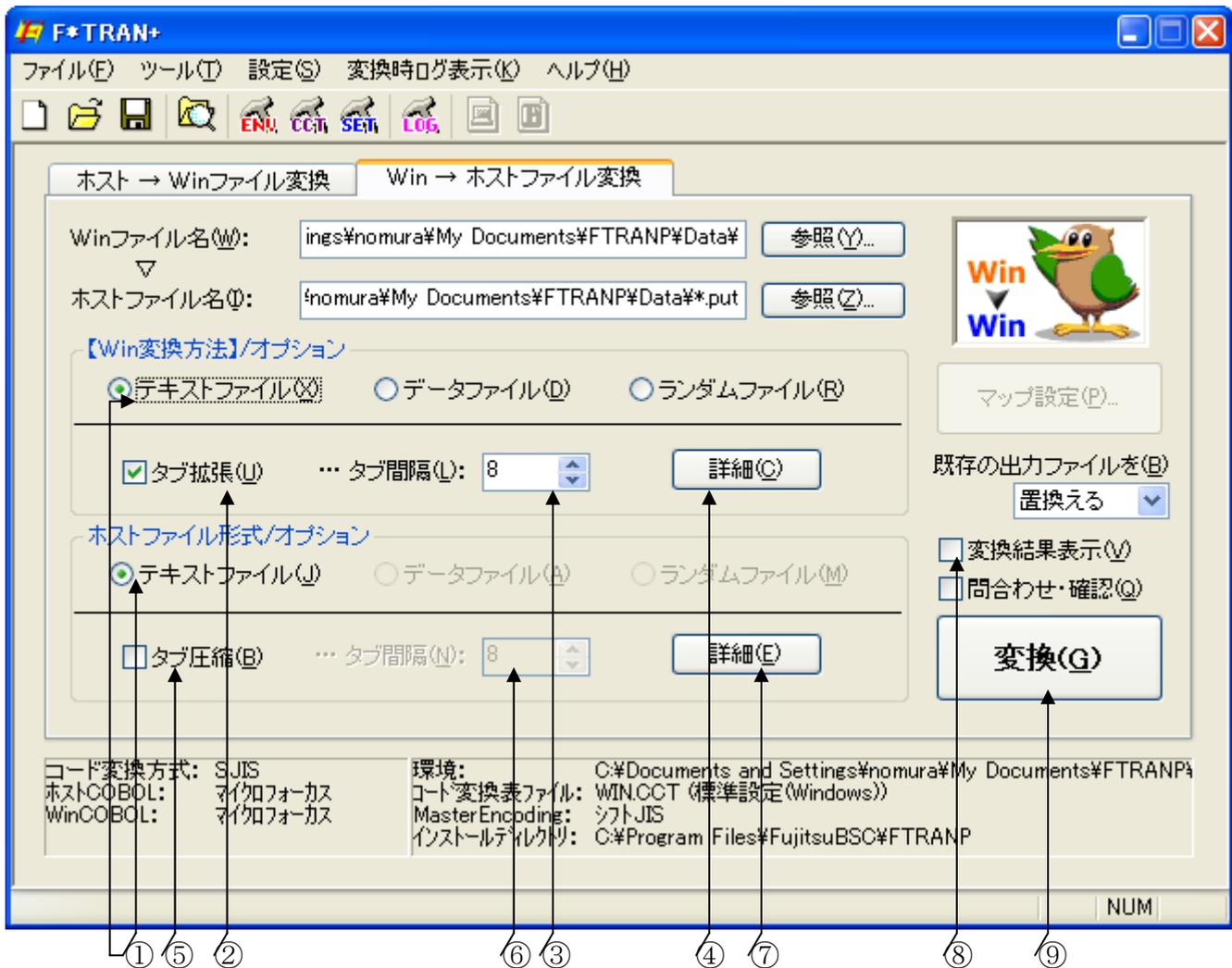
Win→ホストテキストファイル変換の操作方法について説明します。

●ホストが汎用機・オフコンの場合



- ①変換方法のテキストファイル (X) をクリックし、Win→ホストテキストファイル変換を選択します。
- ②タブ拡張の有無を指定するオプションのチェックボックスです。
- ③②でタブ拡張のチェックをONにしたとき、タブ間隔を指定できます。
- ④Winファイルの詳細オプションボタンです。EOF検査の選択、Windows側エンコーディングの設定ができます。
- ⑤ホストファイルのレコード長を指定します。
- ⑥ANK変換かAnk・漢字(KI/KO付)混在変換かを指定するオプションです。
- ⑦変換後に、変換先のファイルの中身を表示ウィンドウに出力するチェックボックスです。
- ⑧変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

●ホストがUnix、Windowsの場合



- ① Win変換方法のテキストファイル (X) をクリックし、ホスト→Winテキストファイル変換を選択します。ホストファイル形式は、テキストファイル (J) 固定になります。
- ② タブ拡張の有無を指定するオプションのチェックボックスです。
- ③ ②でタブ拡張のチェックをONにしたとき、タブ間隔を指定できます。
- ④ Winファイルの詳細オプションボタンです。EOF検査の選択、Windows側エンコーディングの設定ができます。
- ⑤ タブ圧縮の有無を指定するオプションのチェックボックスです。
- ⑥ ⑤でタブ圧縮のチェックをONにしたとき、タブ間隔を指定できます。
- ⑦ ホストファイルの詳細オプションボタンです。EOF付加の選択ができます。
ホストがWindowsの場合のみ有効です。ホストがUnixの場合はありません。
- ⑧ 変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑨ 変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

■オプションの指定

●タブ拡張のオプション

タブ拡張の有無と、タブ拡張するときのタブ間隔を指定します。タブ拡張とは、TAB (09H) をつぎのタブ位置の直前までの連続スペースに展開することです。

タブ拡張のチェックボックスを、

ONにすると、タブ拡張する指定

OFFにすると、タブ拡張しない指定 となります。

タブ拡張指定がONであれば、タブ拡張します。タブ間隔は、2～255の範囲で指定し、それがレコードのおわりまで繰り返し適用されます。**タブ間隔のデフォルトは標準のタブ間隔(8桁きざみ)になります。**これがデフォルトです。

タブ拡張指定がOFFであれば、タブ拡張はしません。

●EOF検査指定

EOFコード(1AH)を検査するか否かを指定します。

EOF検査のチェックボックスを、

ONにすると、EOFコードを検査する指定

OFFにすると、EOFコードを検査しない指定 となります。

EOF検査指定がONであれば、EOFコードを検査し、EOFコードが現れたら変換を終了します。こちらがデフォルトです。

EOF検査指定がOFFであれば、EOFコードを検査しません。単なるデータとして扱います。

●ホストファイル・レコード長指定 <ホストが汎用機・オフコンの場合のみ>

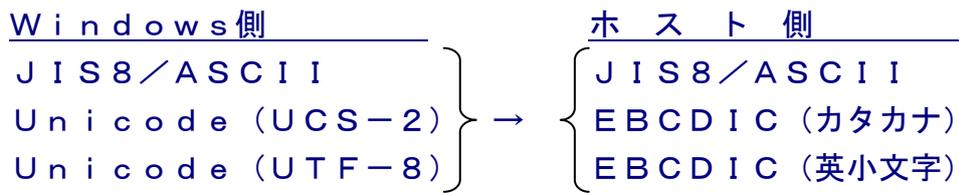
ホストファイルのレコード長を1～32767の範囲の10進数で指定します。デフォルトは、**80バイト**です。レコード長の指定が間違っていると正しいデータ変換が行われませんので、この指定は極めて重要です。

●コード設定オプション <ホストが汎用機・オフコンの場合のみ>

ホストファイル側のコード系（ANK変換かAnk・漢字（KI/KO付）混在変換か）を指定します。

Ank指定

Ank指定すると、すべてANKデータとして変換します。これがデフォルトです。



の9通りの変換が可能です。あらかじめ、変換設定のANKコード設定でホストファイル側のコード系を設定しておかなければいけません（ふつう、セットアップ時に1回だけ行います）。漢字がまじっているときは、つぎのAnk・漢字（KI/KO付）混在指定を使ってください。

Ank・漢字（KI/KO付）混在指定

ANK・漢字混在で、KI/KOつきに変換するとき、この指定をします。あらかじめ、変換設定の漢字変換方式設定で適切な漢字変換方式の割り当てをしておかなければいけません（ふつう、セットアップ時に1回だけ行います）。

●タブ圧縮のオプション <ホストがUnix、Windowsの場合のみ>

タブ圧縮の有無と、タブ圧縮するときのタブ間隔を指定します。タブ圧縮とは、タブ位置の直前までつづく2個以上の連続スペースを、TAB (09H) に置き替えることです。

タブ圧縮のチェックボックスを、

ONにすると、**タブ圧縮する指定**

OFFにすると、**タブ圧縮しない指定** となります。

タブ圧縮指定がONであれば、タブ圧縮します。タブ間隔は、2～255の範囲で指定し、それがレコードのおわりまで繰り返し適用されます。**タブ間隔のデフォルトは標準のタブ間隔(8桁きざみ)**になります。

タブ圧縮は、ソースプログラムなど空白部分が多いファイルの、変換後のファイル容量を減らすのに効果的です。

なお、文字列定数中のスペースまでTABに変換してしまうことを避けるため、アポストロフィ (') か引用符 (") が見つかったら、その行についてはそこでタブ圧縮を打ち切ります。

タブ圧縮指定がOFFであれば、タブ圧縮はしません。これがデフォルトです。

●EOF付加指定 <ホストがWindowsの場合のみ>

EOFコード (1AH) の扱いを指定します。

EOF付加のチェックボックスを、

ONにすると、**EOFコードをつける指定**

OFFにすると、**EOFコードをつけない指定** となります。

EOF付加指定がONであれば、WindowsファイルのおわりにEOFコードをつけます。現在では少なくなりましたが、テキストファイルのおわりにEOFコードがついていないとエラーにするソフトがあります。その場合にも対処するための機能です。

EOF付加指定がOFFであれば、EOFコードはつけません。これがふつうだと思ってください。こちらがデフォルトです。

●エンコーディング

Windows側のエンコーディングを以下の4通りから選択します。

Master Encodingに従う (詳細は「4.12 環境設定」を参照)
シフトJIS
Unicode (UCS-2)
Unicode (UTF-8)

Unicodeの場合は以下の選択も可能です。

BOMつき
BOMなし
BOM自動

『BOM自動』の指定ではUCS-2ではBOMを検査し、UTF-8ではBOMを検査しません。

さらにUCS-2の場合は以下の選択も可能です。

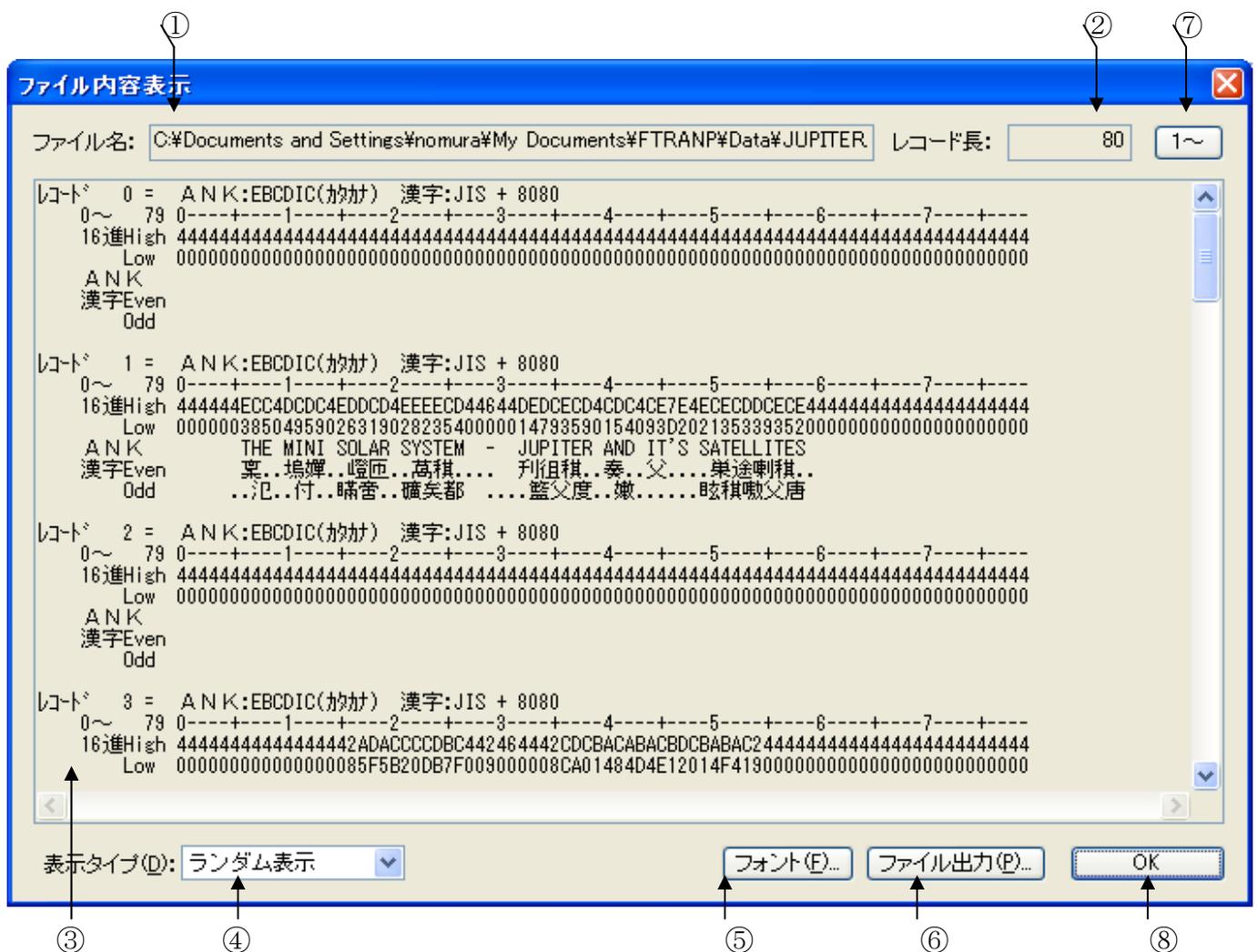
Little Endian
Big Endian

■Win→ホストテキストファイル変換の実行

変換ボタンをクリックすると、Win→ホストテキストファイル変換が始まります。

●ファイル内容表示ウインドウ

変換結果表示 (V) のチェックボックスをONにし、Win→ホストファイル変換を実行すると、実行ウインドウを閉じた後につぎのファイル内容表示ウインドウが開き、変換先のファイルの中身を確認することができます。



- ①変換先のファイル名が表示されます。
- ②変換先のファイルのレコード長が表示されます。(ホストが汎用機・オフコンのみ)
- ③変換先のファイルの中身が、④で選択されている表示タイプで出力されます。
- ④③の表示タイプをランダム表示、テキスト表示、テキストのみ表示に切り替えます。
- ⑤フォント (F) ボタンをクリックして、③の出力文字フォントを切り替えることができます。

- ⑥ファイル出力 (P) ボタンをクリックして、③の内容をファイルに保存できます。
- ⑦桁位置のガイド表示を、0 起点 (先頭を0として数える: デフォルト)、
1 起点 (先頭を1として数える) のどちらかに切り替えることができます。
- ⑧OKボタンをクリックすると、ファイル内容表示ウインドウを閉じます。

ファイル内容表示は変換結果のプレビュー表示機能のため、表示できる容量に制限があります。全ての内容が表示されている場合はデータの最後に、

----- ファイルの最後まで表示しました. -----

全ての内容が表示できなかった場合には、

----- ファイルの途中で表示を切りました. -----

と表示されます。

■注意事項

漢字があるときはコード設定オプションをAnk・漢字(KI/KO付)混在にすることを忘れずに

漢字が入っているときは、あらかじめデータ交換の相手のシステムに合わせて、

変換設定の漢字変換方式設定で、適当な漢字変換方式を割り当てておく

のを忘れないでください。また、ホストが汎用機・オフコンの場合、変換のとき、

コード設定オプションをAnk・漢字(KI/KO付)混在にするのを忘れがち

なので注意してください。

タブ拡張について

ふつう、タブ拡張は必須です。多くの場合、ホストファイルを受け取るシステムのほうでタブをサポートしていないか、サポートしていてもタブ間隔の設定が異なっています。

あふれた分は捨てられる

Windows ファイルの1行が、変換後ホストファイルの1レコードに入り切れないときは、あふれた分が切り捨てられます。そのときはもっと大きなレコード長を指定して再変換してください。とくに、ホストが汎用機・オフコンの場合、Ank・漢字(KI/KO付)混在変換をしたとき、KI/KOが挿入されて、変換後1行の長さが増えることに注意してください。

改行コードがないと、1件だけの変換になる

Windows ファイルの各レコードの末尾に改行コードがついていないと、先頭の1件だけ変換されます。この場合は、Win→ホストランダムファイル変換を使うべきです。

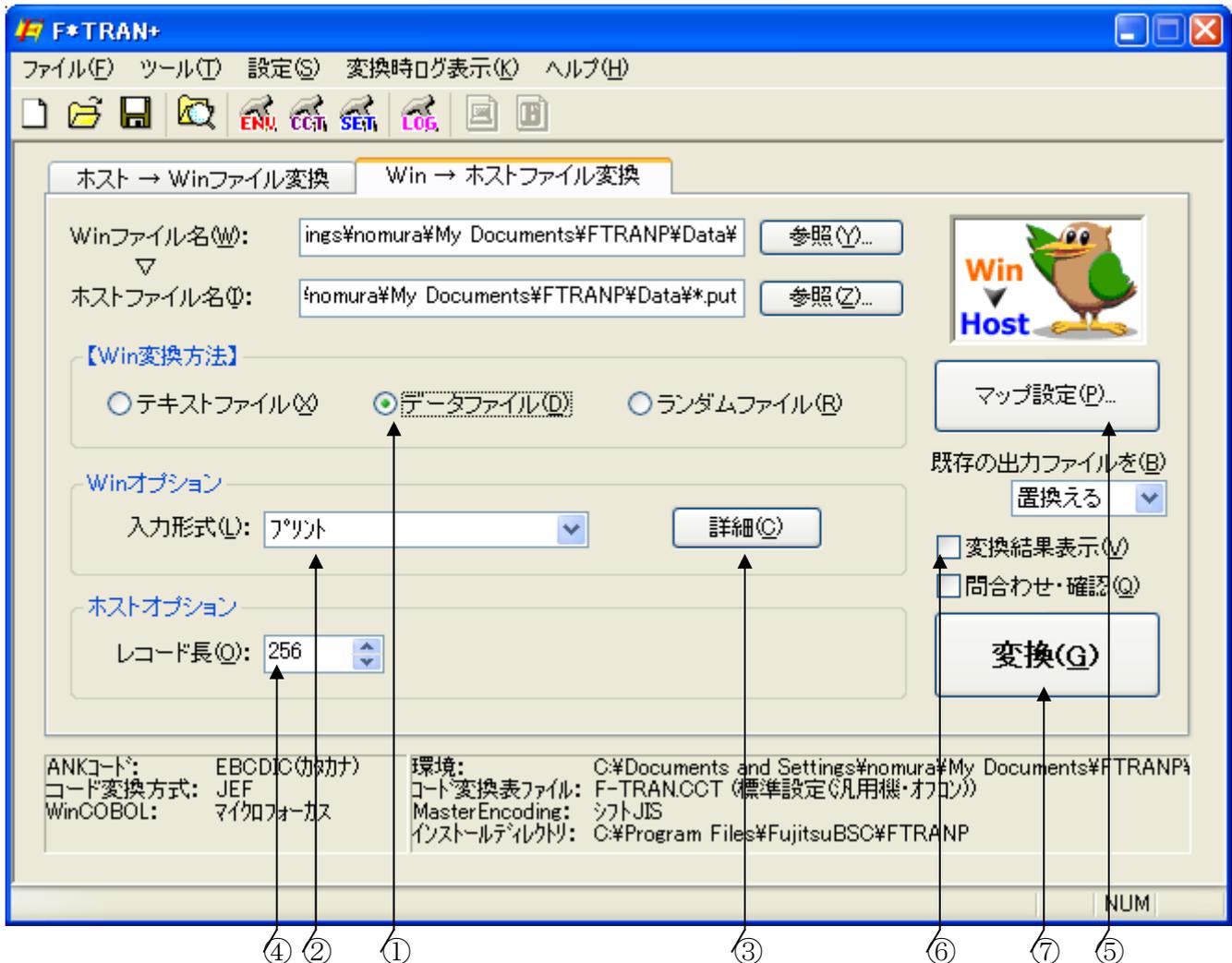
その他の注意事項

「Win→ホストファイル変換」の節を参照してください。

4.8 Win→ホストデータファイル変換

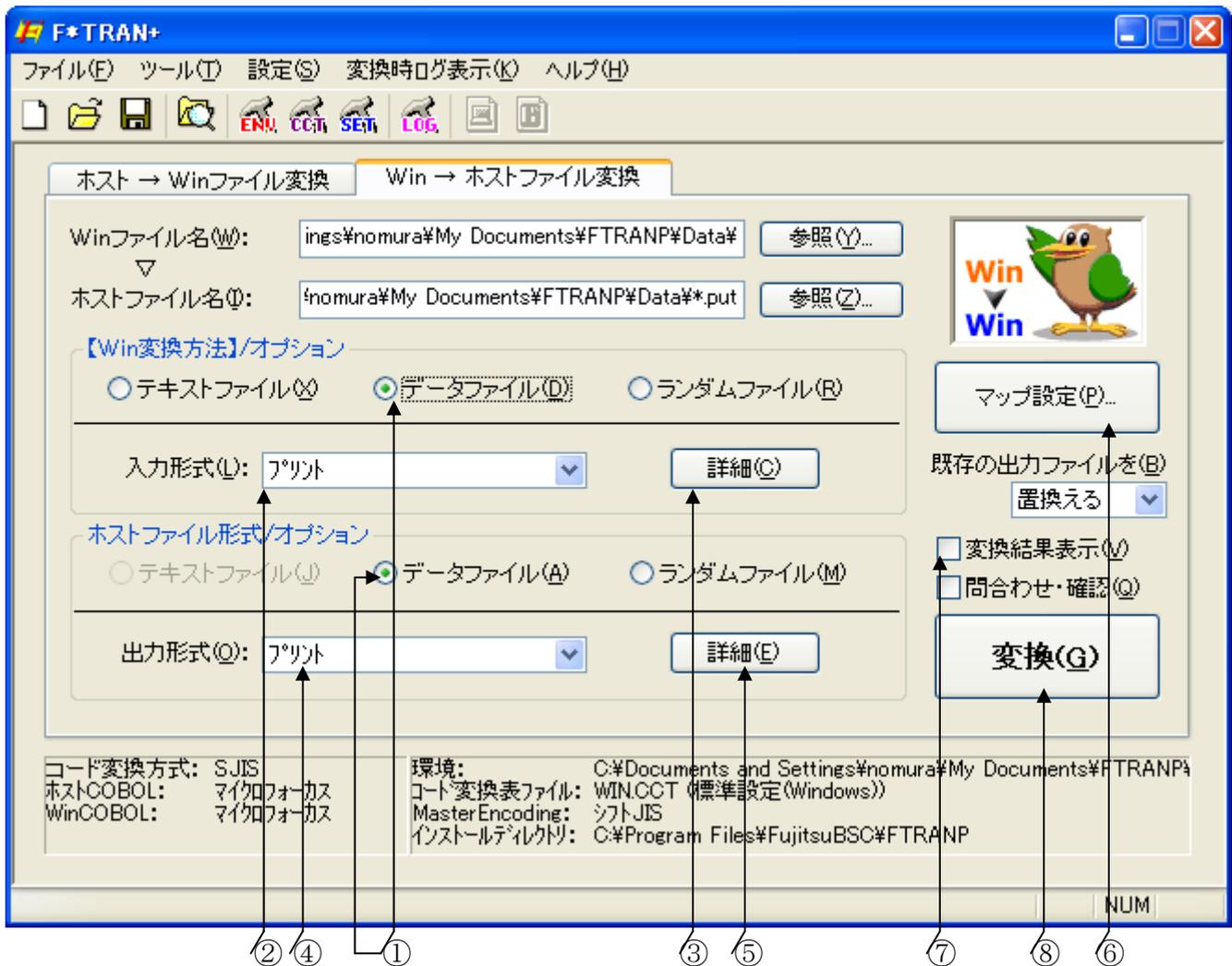
Win→ホストデータファイル変換の操作方法について説明します。

●ホストが汎用機・オフコンの場合

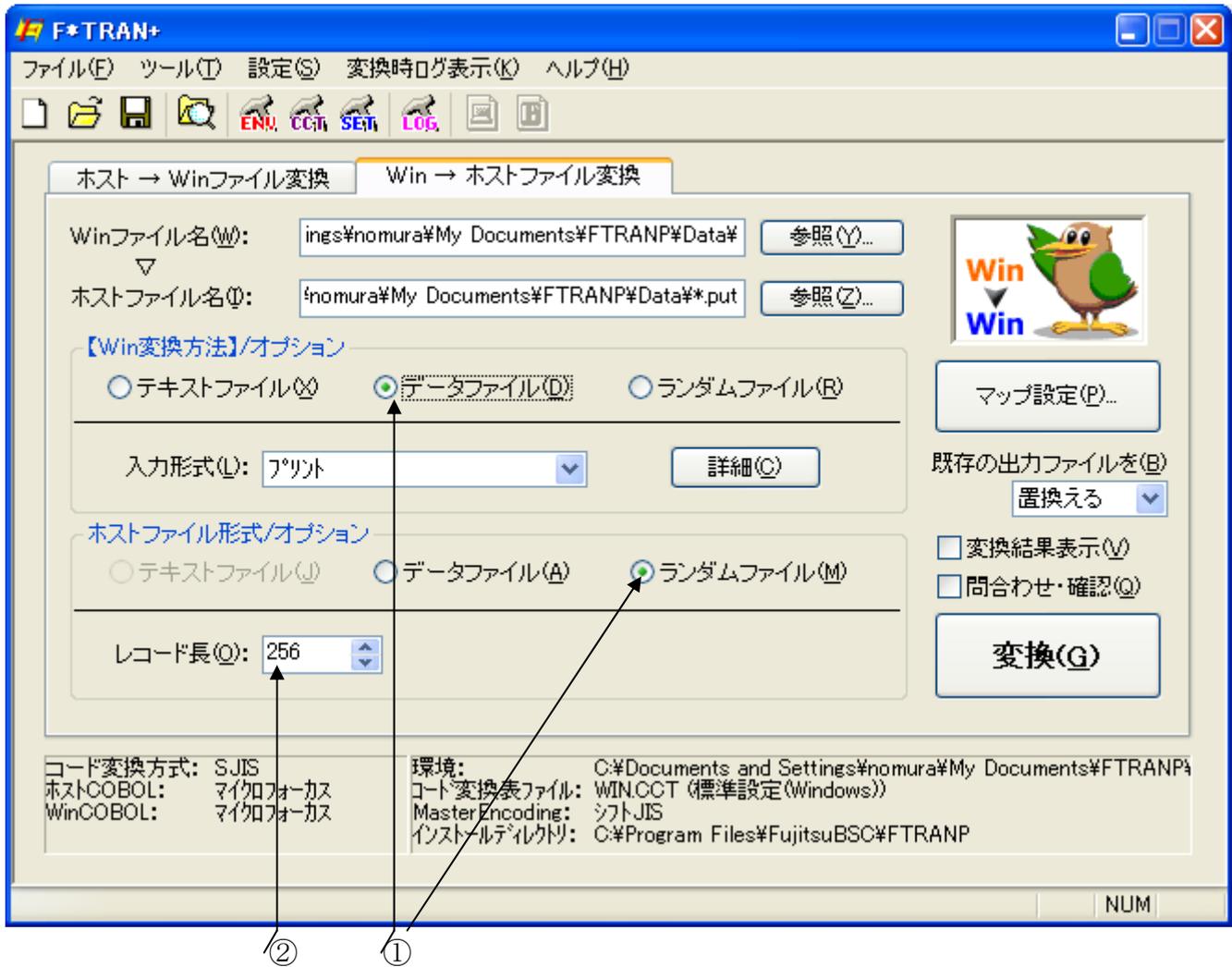


- ①変換方法のデータファイル (D) をクリックし、Win→ホストデータファイル変換を選択します。
- ②入力ファイル形式を指定するオプションです。
- ③Winファイルの詳細オプションボタンです。EOF検査、タブ拡張、空行無視の選択、Windows側エンコーディングの設定ができます。
- ④ホストファイルのレコード長を指定します。
- ⑤マップ設定 (レコードレイアウト設定) をするオプションボタンです。
- ⑥変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑦変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

●ホストがUnix、Windowsの場合



- ① Win変換方法のデータファイル (D) をクリックし、Win→hostデータファイル変換を選択します。hostファイル形式は、データファイル (A)、ランダムファイル (M) の選択ができますが、上の例はhostファイル形式がデータファイル (A) です。
- ② 入力ファイル形式を指定するオプションです。
- ③ Winファイルの詳細オプションボタンです。EOF検査、タブ拡張、空行無視の指定、Windows側エンコーディングの設定ができます。
- ④ 出力ファイル形式を指定するオプションです。
- ⑤ hostファイルの詳細オプションボタンです。EOF付加の選択ができます。
ホストがWindowsの場合のみ有効です。ホストがUnixの場合はありません。
- ⑥ マップ設定 (レコードレイアウト設定) をするオプションボタンです。
- ⑦ 変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑧ 変換を実行するボタンです。このボタンをクリックすると、変換が始まります。



①Win変換方法のデータファイル (D) をクリックし、Win→ホストデータファイル変換を選択します。ホストファイル形式は、データファイル (A)、ランダムファイル (M) の選択ができますが、上の例はホストファイル形式がランダムファイル (M) です。

②ホストファイルのレコード長を指定します。

■オプションの指定

●入力形式オプション

変換するWindowsファイルがプリント形式かデリミタ形式か、デリミタ形式ならコンマ区切り・タブ区切り・スペース区切りのどれなのかを指定します。

プリント ----- プリント形式を変換する
 デリミタ (コンマ) ----- デリミタ形式 (コンマ) を変換
 デリミタ (タブ) ----- デリミタ形式 (タブ) を変換
 デリミタ (スペース) ---- デリミタ形式 (スペース) を変換

プリントを指定すると、デリミタ (区切り文字) なしのプリント形式のファイル (固定長テキストファイル [SDF形式]。ただし、最終項目だけ可変長でもよい) を変換できます。

デリミタを指定すると、デリミタ形式 (区切り文字つき) のテキストファイルを変換できます。区切り文字の種類によって、さらに細かい形式が決まります。デリミタは、

コンマ区切り形式 (CSV形式、K3形式)
 タブ区切り形式 (TAB=09H)
 スペース区切り形式 (SP=20H)

の3つのなかから指定できます。

後述のマップ設定オプションで、デリミタ検出=コンマ (,) を指定したところに、上記の区切り文字があるとみなされます。

●EOF検査指定

EOFコード (1AH) を検査するか否かを指定します。

EOF検査のチェックボックスを、

ONにすると、EOFコードを検査する指定
 OFFにすると、EOFコードを検査しない指定

となります。

EOF検査指定がONであれば、EOFコードを検査し、EOFコードが現れたら変換を終了します。こちらがデフォルトです。

EOF検査指定がOFFであれば、EOFコードを検査しません。単なるデータとして扱います。

●タブ拡張のオプション

タブ拡張の有無と、タブ拡張するときのタブ間隔を指定します。タブ拡張とは、TAB (09H) をつぎのタブ位置の直前までの連続スペースに展開することです。

タブ拡張のチェックボックスを、

ONにすると、タブ拡張する指定

OFFにすると、タブ拡張しない指定 となります。

タブ拡張指定がONであれば、タブ拡張します。タブ間隔は、2～255の範囲で指定し、それがレコードのおわりまで繰り返し適用されます。**タブ間隔のデフォルトは標準のタブ間隔(8桁きざみ)**になります。これがデフォルトです。

タブ拡張指定がOFFであれば、タブ拡張はしません。

●空行無視のオプション

空行を無視するかどうかを指定します。

ONにすると、空行を無視する指定

OFFにすると、空行を無視しない指定 となります。

●引用符はずし

変換するWindowsファイルがデリミタ形式の場合には、項目が引用符でくくられていることがあります。その引用符の種類や扱い方を指定します。なお、この設定はデリミタ形式からの変換の場合のみ有効です。

(1) 引用符の種類

変換元のWindowsファイルで使われている引用符の種類を指定します。

なし 引用符が使われていない時の指定。(もしくは引用符をはずさない指定)

単一引用符 単一引用符「' (半角)」をはずす指定。

2重引用符 2重引用符「" (半角)」をはずす指定。

◆注意 ---- 引用符の種類指定は種類のみ

Windowsファイルに単一引用符(')でくくられた項目と2重引用符(")でくくられた項目が混在していても、どちらか片方の引用符しかはずすことができません。

(2) 引用符項目の取得方法

データの中に引用符を含むのに全体が引用符でくくられていない

,3.5" FDD,

のような例外的な形式のデータの扱いは、ソフトによって解釈がまちまちです。そこで、F*TRAN+では項目取り出し方式を複数用意しておくことで例外的データに対処しています。

【旧F*TRAN風】

- ・ 引用符は必ず対にして使う
- ・ 引用符対は、項目内のどこに何回現れてもよい
- ・ 一对の引用符の中には区切り文字が使われてもよい
- ・ 対の外では、区切り文字をデータとして使えない
- ・ 引用符が2連化してあってもデータとしては認識しない
- ・ 結果的に、すべての引用符が消える

例) ,AAA" B,B" CCC" " DDD, → AAAB, BCCCDDD

【新F*TRAN風】

- ・ 基本的には旧F*TRAN風と同様である
- ・ ただし、2連化した引用符をデータとしての引用符として認識する

例) ,AAA" B,B" CCC" " DDD, → AAAB, BCCC" DDD

【MS-Excel風】

- ◎ 引用符で始まる項目
 - ・ 対になる引用符までは、区切り文字が使われてもよい
 - ・ 対になる引用符までは、2連化した引用符をデータとして扱う
 - ・ 対になる引用符以降は、区切り文字が現れるまで引用符もデータとして扱う
 - ・ 項目内改行はサポートしない
- ◎ 普通の文字で始まる項目
 - ・ 2文字目以降の引用符は、単なるデータとして扱う
 - ・ 区切り文字をデータとして使うことはできない

例) ,3.5" FDD, →

3.5" FDD

【MS-Access風】

- ◎ 引用符で始まる項目
 - ・ 対になる引用符までは、区切り文字が使われてもよい
 - ・ 対になる引用符までは、2連化した引用符をデータとして扱う
 - ・ 対になる引用符以降は、区切り文字が現れるまで削除する
 - ・ 項目内改行はサポートしない
- ◎ 普通の文字で始まる項目
 - ・ 区切り文字をデータとして使うことはできない
 - ・ 2文字目以降に引用符が現れると、それ以降区切り文字までを削除する

例) ,3.5" FDD, →

3.5

◆注意 ---- MS-Excel風とMS-Access風について

MS-Excel風とMS-Access風では、引用符は項目先頭と項目の終わりで各1つずつ存在するのを正常系とし、それ以外については例外系として扱う方式です。両者の違いは、例外系データの扱い方に限られます。入力側データが正常系でありさえすれば、両者に違いは生じないわけです。本マニュアルでのサンプルの例はあくまで例外系のものと認識してください。

●ホストファイル・レコード長指定 <ホストファイルが固定長の場合のみ>

ホストファイルのレコード長を1～32767の範囲の10進数で指定します。デフォルトは、**256バイト**です。レコード長の指定が間違っていると正しいデータ変換が行われませんので、この指定は極めて重要です。

●出力形式オプション <ホストファイルが可変長の場合のみ>

データとしてのテキストファイルにはいくつもの形式があるので、どの形式にするかを指定します。つぎの、7種類の中から選択します。

プリント	-----	プリント形式に変換する
デリミタ (コンマ ・ 圧縮)	----	デリミタ形式 (コンマ) に変換、圧縮あり
デリミタ (タブ ・ 圧縮)	----	デリミタ形式 (タブ) に変換、圧縮あり
デリミタ (スペース ・ 圧縮)	----	デリミタ形式 (スペース) に変換、圧縮あり
デリミタ (コンマ ・ 非圧縮)	----	デリミタ形式 (コンマ) に変換、圧縮なし
デリミタ (タブ ・ 非圧縮)	----	デリミタ形式 (タブ) に変換、圧縮なし
デリミタ (スペース ・ 非圧縮)	----	デリミタ形式 (スペース) に変換、圧縮なし

プリントを指定すると、デリミタ (区切り文字) なしで変換され、プリント形式 (固定長のテキストファイル [SDF形式]) になります。これがデフォルトです。

デリミタを指定すると、デリミタ形式 (区切り文字付きのテキストファイル) に変換されます。デリミタの種類によってさらに細かい形式が決まります。デリミタは、

コンマ区切り形式 (CSV形式、K3形式)

タブ区切り形式 (TAB=09H)

スペース区切り形式 (SP=20H)

の3つのなかから指定できます。

通常は、コンマ区切り (CSV) 形式への変換が一般的です。さらに、

変換後に圧縮をかける (可変長になる。デフォルト)

変換後に圧縮しない (固定長のまま)

を指定できます。

圧縮をかけると本来不要なスペース、つまり、

レコードの先頭・末尾のスペース

デリミタの前後のスペース

引用符の前のスペース

が削除されます。

圧縮をかける機能があるのは、つぎのような理由があるためです。

変換後のWindowsファイルの容量を、大幅に減らすことができる

一部の市販ソフトでは、不要なスペースがあるとうまくデータを読み込めない

●EOF付加指定 <ホストがWindowsで、可変長ファイルの場合のみ>

EOFコード（1AH）の扱いを指定します。

EOF付加のチェックボックスを、

ONにすると、EOFコードをつける指定

OFFにすると、EOFコードをつけない指定 となります。

EOF付加指定がONであれば、WindowsファイルのおわりにEOFコードをつけます。現在では少なくなりましたが、テキストファイルのおわりにEOFコードがついていないとエラーにするソフトがあります。その場合にも対処するための機能です。

EOF付加指定がOFFであれば、EOFコードはつけません。これがふつうだと思ってください。こちらがデフォルトです。

●エンコーディング

Windows側のエンコーディングを以下の4通りから選択します。

Master Encodingに従う（詳細は「4.12 環境設定」を参照）

シフトJIS

Unicode (UCS-2)

Unicode (UTF-8)

Unicodeの場合は以下の選択も可能です。

BOMつき

BOMなし

BOM自動

『BOM自動』の指定ではUCS-2ではBOMを検査し、UTF-8ではBOMを検査しません。

さらにUCS-2の場合は以下の選択も可能です。

Little Endian

Big Endian

●引用符くくり

(1) 引用符の種類

この設定はホストがUnix/Windowsで出力形式がデリミタ形式に指定されているときのみ有効です。引用符の種類それぞれの意味を説明します。

- なし 引用符でくくらずに変換内容を出力します。
- 単一引用符 単一引用符「'（半角）」をくくる。
- 2重引用符 2重引用符「"（半角）」をくくる指定。

「なし」の指定の場合は、Map文で引用符くくりの指定がしてあっても無視します。

◆注意 ---- Map文の引用符記号は、常に二重引用符「"」

「引用符の種類」については「単一引用符」指定時も、Map文において引用符くくりを使う記号は、2重引用符「"（半角）」ですので注意してください。

◆注意 ---- 引用符の種類指定は一種類のみ

単一引用符（'）と2重引用符（"）の両方を混在してくくることはできません。必ずどちらか一方でくくられます。

(2) データとしての引用符の扱い

- 削除 項目のデータとしての引用符は削除する指定。
- 空白化 項目のデータとしての引用符は半角スペースに変換する指定。
- 2連化 項目のデータとしての引用符は、2連化して出力する指定。

2連化を指定した場合、変換後データの項目長が変換前と変わりますが、それについては「2連化時の優先処理」で指定します。

◆注意 ---- 「データとしての引用符の扱い」のデータ内引用符は「引用符の種類」に依存

「データとしての引用符の扱い」ではデータ内容として引用符が存在した時の扱いを指定しますが、その「引用符」というのは「引用符の種類」で指定したものに限りません。

変換処理で使用する引用符として、2重引用符を指定した場合は、単一引用符は空白化などの処理の対象になることはありません。

(3) 2連化時の優先順位

引用符の2連化によって入力データ長に対する変換後データ長が膨れ上がった場合の処理を指定します。

出力幅優先 出力幅を優先して、データの後方部分を切り捨てる指定。

データ優先 変換後の長さそのままを出力する指定。

「2連化時の優先順位」は「データとしての引用符の扱い」において「2連化」を指定していない場合には有効になりません。

◆注意 ---- 「2連化時の優先順位」にてデータ優先をした時の出力幅について

「2連化時の優先順位」で「データ優先」を指定すると、M a p 文で出力幅指定を省略した時だけでなく出力幅指定をしていたとしてもそれを無視して変換します。下は変換処理時の実際の出力幅です。

出力幅指定をした場合	…	M a p 文指定の出力幅+引用符の長さ
出力幅指定を省略した場合	…	M a p 文指定の入力幅+引用符の長さ

● マップ設定オプション

このマップ設定オプションで、項目別の細かい変換方法を指示します。本来なら、自動的に項目を認識して変換ができると便利です。しかし、ホストの、とくにCOBOLのデータには、データ自身に桁数や小数点位置の判断に必要な情報が含まれていない、という特性があります。そのため自動変換は原理的に不可能なのです。

マップ設定のボタンをクリックすると、つぎのウインドウが開きます。



マップ設定には、

- 簡易設定 —— 設定ウインドウに従って、必要な値を入力するだけで設定ができます。通常は、こちらの機能だけで間に合う場合がほとんどです。
- 詳細設定 —— フリーフォーマットで、マップ設定を記述するエディタモードです。簡易設定だけでは設定できない特殊な設定をする場合、あるいは、既存のパラメータファイル等を流用したい場合等に使用します。マップ設定の文の書き方を覚えなくては設定できません。

の2通りの方法があります。

◆注意 ---- **簡易設定と詳細設定の組み合わせ**

簡易設定で設定した値は、詳細設定で読み込むことができるテキスト形式に直されます。よって、簡易設定をした後の項目に、詳細設定で特殊設定をつけ足すこともできます。

◆注意 ---- **詳細設定→簡易設定の順で設定した場合の制約**

詳細設定→簡易設定の順で設定した場合は、原形が変わってしまう場合があります。これは、詳細設定で設定した項目が簡易設定で読み込まれることにより、スリム化されてしまうからです。ただし、原形が変わってしまっても、設定内容が変わるわけではありません。

◆注意 ---- **入力幅、出力幅の指定について**

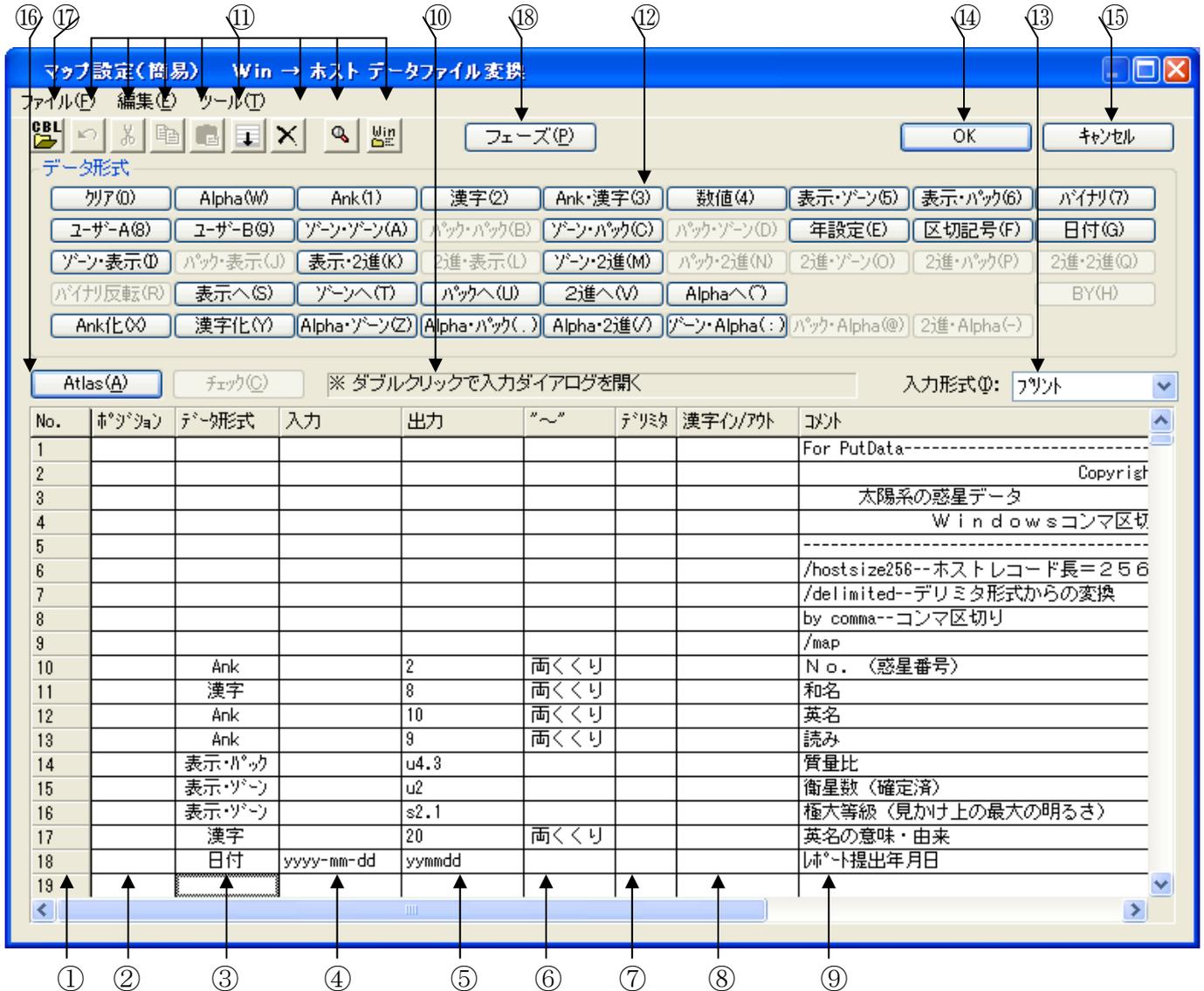
入力側や出力側の文字コードによって、入力幅、および出力幅の指定には注意が必要です。

たとえばANKデータを変換する場合、入力がWindowsでエンコーディングがUCS-2の場合は文字数×2の入力幅を指定する必要があります。

出力幅が省略された場合の省略値は入力コードと出力コードから入力幅に指定された値を最小として自動的に計算して求めています。ただし、漢字とANKが混ざった項目に対して漢字項目として変換した場合（ANK文字が全角化される）や、汎用機・オフコンの半角カナをANK変換でUTF-8に変換する場合など、明確に出力幅を指定しないと出力データが途中で切れてしまう場合もあります。

文字コードとしてUnicodeを指定する場合や半角カナ文字がある場合などは、入力幅や出力幅の指定には十分注意する必要があります。

<マップ設定（簡易）ウィンドウ>



- ①No. は、1～2000です。ここは非入力項目で、行選択の時にクリックします。
- ②項目の桁位置等を入力します。通常、項目を組み替えるとき以外は入力しません。
ダブルクリックすると、入力サブウィンドウが開きます。
- ③項目のデータ形式を指定します。ダブルクリックすると、入力サブウィンドウが開きます。
- ④項目の入力幅を入力します。通常、デリミタ形式ファイルを変換する場合は省略します。
Ank／漢字／Ank・漢字／バイナリの場合、定数を指定することもできます。
Alphaへ／表示へ／ゾーンへ／パックへ／2進への場合、数値定数かシステム変数を入力します。
- ⑤項目の出力幅を入力します。プリント形式ファイルで、出力幅＝入力幅の場合は省略します。
データ形式がゾーン／パック／2進指定の場合、ピクチャまたは2進ピクチャで指定します。
- ⑥デリミタ形式から変換するときに、項目の引用符をはずす指定ができます。ホストがUnix、Windowsの場合、デリミタ形式の項目を引用符でくくる指定にもなります。

- ⑦デリミタ形式から変換するとき、デリミタ検出を無効にする指定ができます。
ホストがUnix、Windowsの場合、デリミタ形式に変換するときデリミタ挿入を無効にする指定にもなります。
- ⑧ホストが汎用機・オフコンの場合は、項目の前後に漢字イン／アウトを挿入する指定です。
ホストがUnix、Windowsの場合は、改行コードを挿入する指定です。
- ⑨コメントの入力ができます。
- ⑩セルポインタの入力ガイダンスメッセージが出力されるフィールドです。
誤入力をした場合のエラーメッセージも出力されます。

⑪は編集 (E)、ツール (T) のメニューの機能がボタン化されています。



直前の編集作業が無効になり、元に戻ります。



1行または複数行を選択 (①の操作) をした後に、このボタンをクリックすると、選択した項目の内容が切り取られ、カットバッファに入ります。



1行または複数行を選択 (①の操作) をした後に、このボタンをクリックすると、選択した項目の内容がカットバッファに入ります。



カットバッファに入った内容を、セルポインタがある行へ貼りつけます。
すでに、セルポインタ以降の行に設定項目がある場合は、挿入になります。



セルポインタがある行以降が1行ずつ下がります。通常は、行挿入をする場合に使用します。



選択した項目の内容が削除されます。



変換プレビュー (変換前／変換後のデータ確認) ができます。



Winファイルエディタが起動します。

Windowsファイルの中身を見ること (編集も可能) ができます。

- ⑫セルポインタが選択項目にある場合に、選択肢のボタンが表示されるエリアです。
- ⑬入力形式オプションの指定 (プリント、デリミタ3種類) を変更できます。
- ⑭設定が完了したら、OKボタンをクリックします。
- ⑮キャンセルボタンをクリックすると、すべての入力／編集作業が無効になります。
- ⑯マルチレコードの設定ができるAtlasモードに切り替えるボタンです。*1
- ⑰COBOLのソースファイル (Copy句) を読み込むボタンです。*1
- ⑱フェーズボタンをクリックすると、フェーズを設定するウィンドウが開きます。*1

*1) 詳細は、マルチレコード編のマニュアルを参照してください。

◆注意 ----- **マップ未設定時のデータ変換動作**

マップ未設定の場合は、すべてをANKデータとみなしてデータ変換を行います。

◆注意 ----- **デリミタ検出について**

デリミタ形式のファイルを変換するときは、なんらかの方法で項目分けをしなければいけません。コンマを使うと行頭からデリミタ、デリミタからデリミタ、デリミタから行末までを項目として認識します。なお、引用符でくくられた項目は、そのなかにデリミタがあっても単なる文字データとして扱われます。

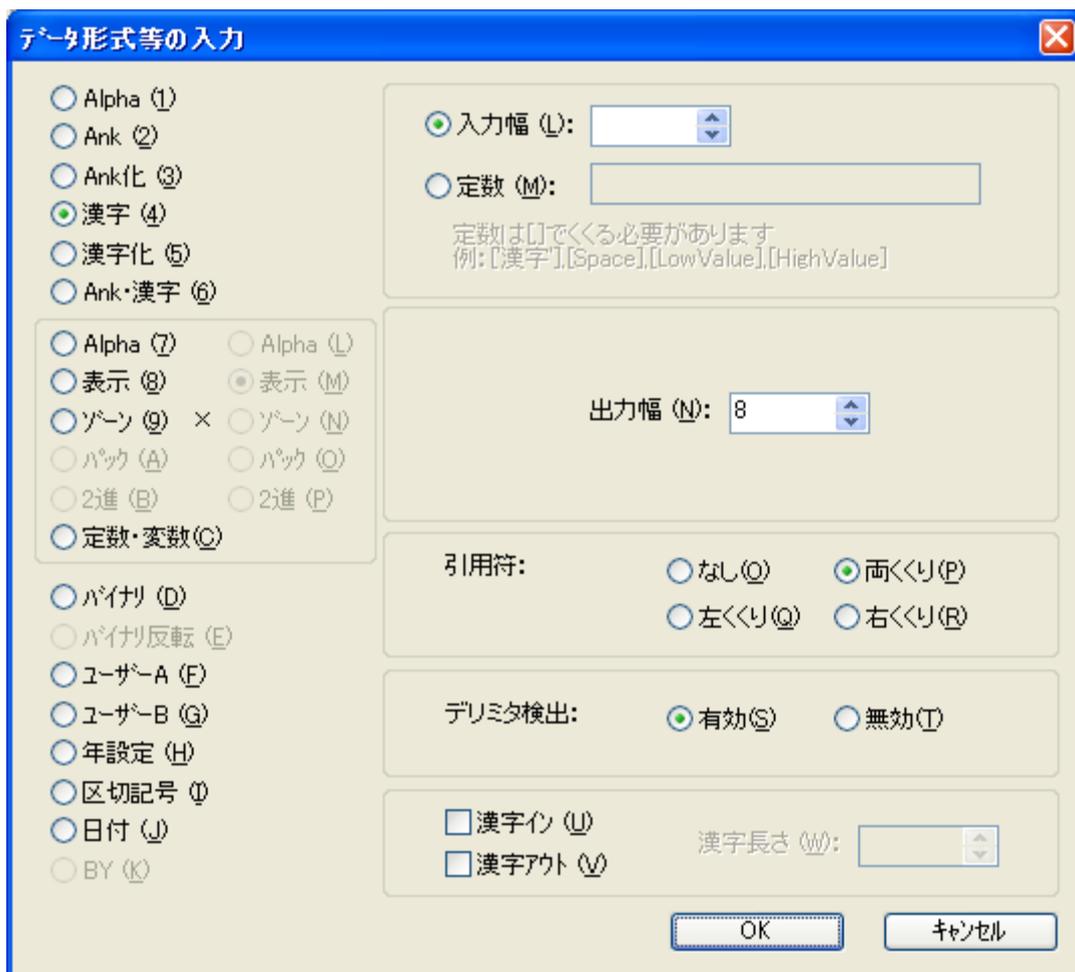
マップ設定（簡易）で項目を設定し、デリミタ形式のファイルを変換する場合、変換元ファイルの項目の後にデリミタ（区切り文字）があることを前提に変換します（最終項目は除く）。検出される区切り文字は指定された入力形式により、コンマ、タブ、スペースの3つのいずれかになります。なお、プリント形式への変換のときは、デリミタ検出の指定は無効になります。また、デリミタ検出に設定したが、ある項目についてはデリミタ検出をしたくない場合は、「デリミタ」欄を「無効」と設定します。また、簡易設定後に、詳細設定を行い修正することもできます。

<マップ設定（簡易）サブウィンドウ>

ポジションの欄をダブルクリックすると、つぎのサブウィンドウが開きます。入出力桁位置や入出力スキップの設定を簡便に行うことができます。



データ形式の欄をダブルクリックすると、つぎのサブウィンドウが開きます。データ形式から引用符はずし（ホストが汎用機・オフコンの場合、漢字項目は漢字イン／アウト）までの設定を簡便に行うことができます。

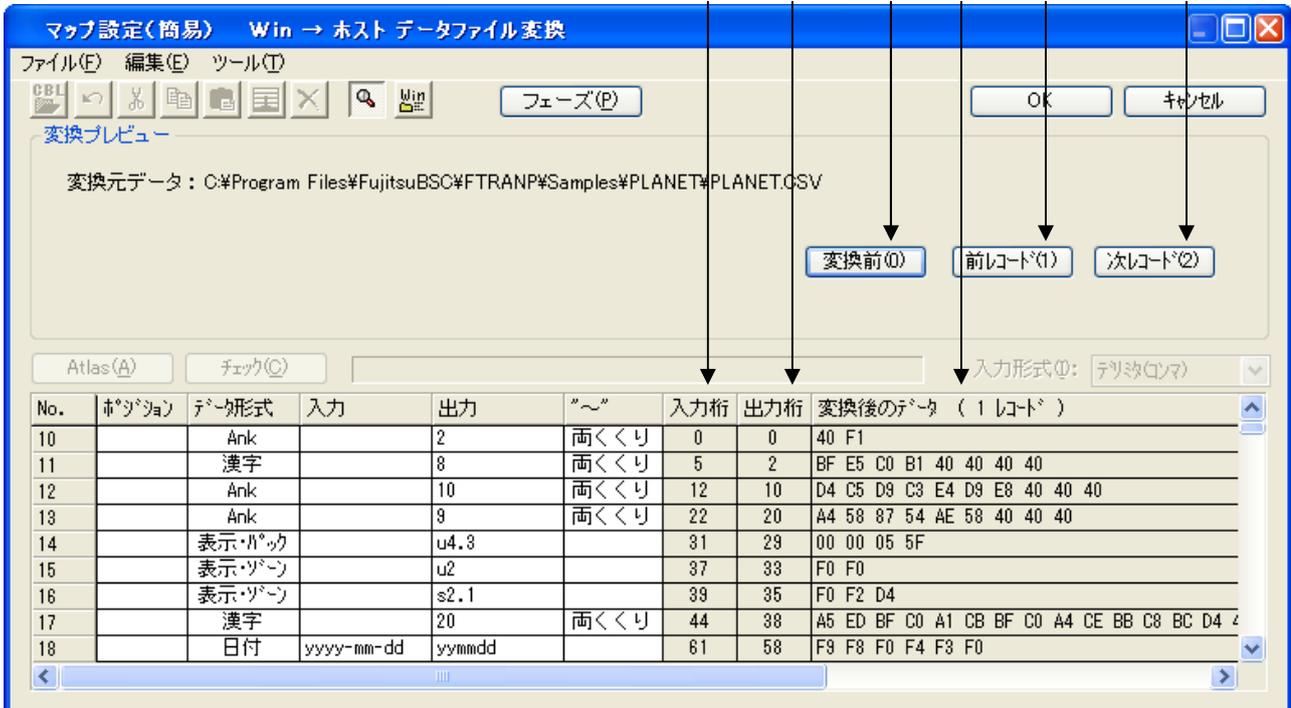


<マップ設定（簡易）・変換プレビュー機能>

変換プレビューボタンをクリックすると、マップ設定（簡易）のウィンドウはつぎのような状態になり、変換前／変換後のデータを確認することができます。



↑変換前データ表示 変換後データ表示↓ ③ ③ ① ④ ② ③



- ①変換前／変換後のデータ表示の切り替えを行うボタンです。④の表示が変わります。
 ②前レコード／次レコード表示のためのボタンです。
 ③ここには、入力レコード／出力レコードのポジション（桁）が表示されます。
 ④データ表示は、変換前が文字、変換後が16進、未定義個所が文字になります。

プリント形式とデリミタ形式では指定の方法と考え方がかなり違うので、この2つを必要に応じて分けて説明することにします。

デリミタ形式の場合には、Windows側（入力項目）が可変長で、ホスト側（出力項目）が固定長だという点に留意して読み進めてください。

(1) ポジション（桁位置）の指定

プリント形式の場合

変換対象にするWindows側（入力）の桁位置や、変換結果を書き込むホスト側（出力）の桁位置を、別の任意の位置に移動できます。現在、処理対象にしている桁位置を、この指定で強制的に変更できます。この機能を利用すると、項目の組み替えなどが簡単に実現できます。

入力桁位置は、ふつう10進数で指定します。式による指定もでき、そのなかでは、

. **Windows（入力）側の現在の桁位置** という特殊変数が使えます。

たとえば、. **-40** と指定すれば、
今の入力桁位置から40バイト戻る という意味になります。

出力桁位置を移動することもできます。ふつう、**: 10進数** で桁位置を指定します。10進数の代わりに式による指定もでき、そのなかでは、

\$ **ホスト側（出力）の末尾**
. **ホスト側（出力）の現在の桁位置** という特殊変数が使えます。

デリミタ形式の場合

変換結果を書き込むホスト側（出力）の桁位置を、別の任意の位置に移動できます。現在、処理対象にしている桁位置を、この指定で強制的に変更できます。この指定を利用すると、項目の組み替えなどが簡単に実現できます。

デリミタ形式からの変換の場合、出力桁位置の移動だけが有効です。入力桁位置の指定はできません。出力桁位置は、**: 10進数** で指定します。式による指定もでき、そのなかでは、

\$ **ホスト側（出力）の末尾**
. **ホスト側（出力）の現在の桁位置** という特殊変数が使えます。

たとえば、**: \$-8** と指定すれば、
ホストレコード末尾の8バイト前に出力先を移動する という意味になります。

◆注意 ---- 先頭を0桁目とする

F*TRAN+では、レコードの先頭を0桁目として数えます。

(2) ポジション (デリミタ位置) の指定

変換対象にするWindows側 (入力) データのデリミタ形式の項目位置を、別の任意の項目位置に移動できます。現在、処理対象にしている項目位置を、この機能で強制的に変更できます。この機能を利用すると、デリミタ形式の項目の組み替えなどが簡単に実現できます。

デリミタ位置は、 **@@入力項目番号** で指定します。式による指定もでき、そのなかでは、

- . Windows側 (入力) の現在の項目位置
 - \$ Windows側 (入力) の入力全体の項目数
 - * Windows側 (入力) の残りの項目数
- という特殊変数が使えます。

たとえば、

- @@. - 2** と指定すれば、 **今の入力項目から2項目戻る**
- @@\$ - 2** と指定すれば、 **Windows側 (入力) レコード末尾から2項目に位置する**

という意味になります。

◆注意 ---- 先頭を0項目目とする

F*TRAN+では、レコードの先頭項目を0項目目として数えます。

(3) ポジション（入カスキップ）の指定

プリント形式の場合

Windows側（入力）レコードに不要な項目があるとき、それをスキップして変換できません。

スキップする幅は、 **^バイト数** で指定します。

たとえば、3バイト分スキップしたいなら、 **^3** と指定します。

バイト数は省略でき、省略すると1バイトとみされるので、

^3 は **^^^** と指定したのと同じです。

スキップする幅は式による指定もでき、そのなかでは、

*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

デリミタ形式の場合

デリミタ形式の場合には、出力幅に0を指定すると項目スキップできます。

(4) ポジション（出カスキップ）の指定

入カスキップ（プリント形式）とは逆に、ホスト側（出力）に何桁かスペースを作ることできます。空項目を作るのがおもな用途です。

スキップする幅は、 **__バイト数** で指定します。

たとえば、3バイト分スキップしたいなら、 **__3** と指定します。

バイト数は省略でき、省略すると1バイトとみされるので、

__3 は **_____** と指定したのと同じです。

(5) データ形式

[Alpha]

Windows側のデータに関係なく1バイトごとにANK項目として変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。出力側のコード体系に関係なく、ANKコードの設定に従って1バイトデータに変換されて出力します。

プリント形式の場合

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- * **Windows側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。項目長を変更するときは、出力幅を10進のバイト数で指定します。出力幅は式による指定もでき、式のなかでは、

- \$ **ホスト側（出力）のレコード長**
- * **ホスト側（出力）の残りバイト数** という特殊変数が使えます。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のANK項目のおわりにスペースが詰められます。

デリミタ形式の場合

ふつう、入力幅は省略し、出力幅だけを10進のバイト数で指定します。出力幅は式による指定もでき、式のなかでは

- * **ホスト側（出力）の残りバイト数** という特殊変数が使えます。

プリント形式・デリミタ形式共通

出力幅は省略できます。出力幅を省略すると、「正確な出力に必要な幅」とみなします。出力幅の指定は、プリント形式の場合、デリミタ形式の場合を参照してください。

[Ank]

ANK項目を変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。

プリント形式の場合

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

出力幅は式による指定もでき、式のなかでは、

\$ ホスト側（出力）のレコード長

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のANK項目のおわりにスペースが詰められます。

デリミタ形式の場合

ふつう、入力幅は省略し、出力幅だけを10進のバイト数で指定します。出力幅は式による指定もでき、式のなかでは

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

プリント形式・デリミタ形式共通

入力幅の代わりに、ANK形式の定数を指定することもできます。その場合、指定した定数はWindows側（入力）レコードにあったかのようにみなされ、その後コード変換されます。

入力値として、つぎの定数が使えます。

文字列定数

文字列はANK形式

汎用定数

Space、LowValue、HighValue

出力幅は省略できます。出力幅の省略値は「**見た目の文字数×2**」になります。指定された定数は内部的にUCS-2コードなので、出力コードがUCS-2以外の場合は明示的に出力幅を指定する必要があります。出力幅の指定は、プリント形式の場合、デリミタ形式の場合を参照してください。

[漢字]

漢字項目を変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。

プリント形式の場合

入力幅は、10進のバイト数で指定します（漢字の文字数ではありません）。式による指定もでき、そのなかでは、

*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。

項目長を変更したいときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、漢字項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、ホスト側の漢字項目のおわりに漢字変換方式に設定されている漢字スペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

\$ ホスト側（出力）のレコード長

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

デリミタ形式の場合

ふつう入力幅は省略し、出力幅だけを10進のバイト数で指定します。出力幅は式による指定もでき、そのなかでは、

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

プリント形式・デリミタ形式共通

入力幅の代わりに、漢字形式の定数を指定することもできます。その場合、指定した定数はWindows側（入力）レコードにあったかのようにみなされ、その後コード変換されます。

入力値として、つぎの定数が使えます。

文字列定数	文字列は漢字形式
汎用定数	Space

出力幅は省略できます。出力幅を省略すると、「正確な出力に必要な幅」とみなします。出力幅の指定は、プリント形式の場合、デリミタ形式の場合を参照してください。

[Ank・漢字]

ANK・漢字まじり項目を変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。ホストが汎用機・オフコンの場合、変換後、漢字文字列の前後にはKI/KOが挿入されます。この、KI/KO挿入のタイミングも漢字変換方式の設定で決まります。そのため、オーバーフローの危険性があるので、出力幅の指定には注意しなければいけません。

プリント形式の場合

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

ふつうは、KI/KOが挿入されてデータ長が長くなる分を加算した出力幅を指定します。オーバーフローすると、ANK・漢字まじり項目のおわりのほうが切り捨てられます。ただし、漢字モードのままおわることはありません。また、KOが途中で切れることもありません。

出力幅は式による指定もでき、そのなかでは、

\$ ホスト側（出力）のレコード長

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

デリミタ形式の場合

ふつう、入力幅は省略し、出力幅だけを10進のバイト数で指定します。出力幅は式による指定もでき、式のなかでは、

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

プリント形式・デリミタ形式共通

入力幅の代わりに、ANK・漢字まじり形式の定数を指定することもできます。その場合、指定した定数はWindows側（入力）レコードにあったかのようにみなされ、その後コード変換されます。入力値として、つぎの定数が使えます。

文字列定数	文字列はANK・漢字まじり形式
汎用定数	Space、LowValue、HighValue

出力幅は省略できます。出力幅を省略すると、「正確な出力に必要な幅」とみなします。出力幅の指定は、プリント形式の場合、デリミタ形式の場合を参照してください。

[数値]

文字形式の数値項目どうしの変換をします。

入力幅は、バイト数で指定します。省略すると、入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

数値変換は、Ank項目変換と後述の表示・ゾーン変換の中間的なものです。Ank項目変換と比較すると、

文字形式数値しか通さない

入力幅を省略すると15桁とみなす

右詰めになる

などの点が異なります。

「文字形式数値しか通さない」というのは、具体的には、

＋、－、0～9、ピリオド(.)、E、e、D、d

しか変換しないで、これら以外の文字は捨ててしまうということです。たとえば、通貨記号(¥/\$)や位取りのコンマ(,)などは削除されるので、リストファイルから入力データファイルを作るときなどに役立ちます。

[表示・ゾーン]

Windowsの文字形式数値項目を、ホストのCOBOLのゾーン形式数値項目に変換します。

入力幅は、バイト数で指定します。省略すると、入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**－123.4** という数字を5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が省略値を超えるときは、

入力幅=40、ピクチャ=u15.2 のように、ダミーの入力幅を指定してください。

[表示・パック]

Windowsの文字形式数値項目を、ホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。

入力幅は、バイト数で指定します。省略すると入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**123.4** という数字を3バイトの符号つきパック形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。**123.4** という数字を3バイトのBCD形式の項目に記録するとすれば、ピクチャは **b5.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が省略値を超えるときは、

入力幅=40、ピクチャ=u15.2 のように、ダミーの入力幅を指定してください。

[バイナリ]

バイナリ変換（無変換）は、「コード変換を一切しない」という変換方法です。通常、Win→ホストデータファイル変換では使用しません。

プリント形式の場合

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

出力幅は式による指定もでき、式のなかでは、

\$ ホスト側（出力）のレコード長

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

項目長を縮めると、バイナリ項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のバイナリ項目のおわりに、ホストが汎用機・オフコンの場合はNUL（00H）が、ホストがUnix/Windowsの場合はスペース（20H）が詰められます。

デリミタ形式の場合

ふつう、入力幅は省略し、出力幅だけを10進のバイト数で指定します。出力幅は式による指定もでき、式のなかでは

* **ホスト側（出力）の残りバイト数** という特殊変数が使えます。

プリント形式・デリミタ形式共通

入力幅の代わりに、バイナリ形式の定数を指定することもできます。その場合、指定した定数はWindows側（入力）レコードにあったかのようにみなされ、コード変換されずにそのまま出力されます。

入力値として、つぎの定数が使えます。

16進列定数、汎用定数（LowValue、HighValue）

出力幅は省略できます。出力幅を省略すると、「正確な出力に必要な幅」とみなします。出力幅の指定は、プリント形式の場合、デリミタ形式の場合を参照してください。

[ユーザーA]、[ユーザーB]

ユーザーA/B変換は、利用者独自のバイト単位の変換処理が必要なときに、ユーザA/B変換表を書き替えて利用します。ユーザーA/B変換には、Alpha項目変換の説明がほとんどそのまま当てはまります。

プリント形式の場合

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

* **Windows側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ユーザーA/B項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、Windows側のユーザーA/B項目のおわりにスペース（20H/40H）が詰められます。

出力幅は式による指定もでき、式のなかでは、

\$ **ホスト側（出力）のレコード長**

* **ホスト側（出力）の残りバイト数** という特殊変数が使えます。

デリミタ形式の場合

ふつう、入力幅は省略し、出力幅だけを10進のバイト数で指定します。出力幅は式による指定もでき、式のなかでは、

* **ホスト側（出力）の残りバイト数** という特殊変数が使えます。

[ゾーン・ゾーン]

Windows COBOLのゾーン形式数値項目を、ホストのCOBOLのゾーン形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**—123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、出力ピクチャは **s4.1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s4.1という指定は、入力ピクチャ=s4.1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s4.1、出力ピクチャ=省略という指定は、出力ピクチャ=s4.1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[ゾーン・パック]

Windows COBOLのゾーン形式数値項目を、ホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**—123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、出力ピクチャは **s4.1** と指定します。BCD形式なら、**b4.1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s4.1という指定は、入力ピクチャ=s4.1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s4.1、出力ピクチャ=省略という指定は、出力ピクチャ=s4.1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[年設定]

日付データ項目を変換する際の、年の2桁（yy）と4桁（yyyy）の変換方式を設定します。Wnn、または、Snnの形式で設定します。Wnnの“W”はウィンドウ方式を、Snnの“S”はシフト方式を意味し、“nn”は00～99の数字で指定します。

入力側に **W30** と指定すれば、入力データの年を1930～2029年とみなし、出力側に **S25** と指定すれば、出力データの年下2桁を、-25します。

また、シフト方式（“Snn”指定）では、つぎの特殊指定ができます。

SShowa は、“S25”の指定と同じ（昭和通年方式）

SHeisei は、“S88”の指定と同じ（平成通年方式）

年設定は日付項目変換が実行された時に適用になり、複数の年設定がなされている場合は、日付項目変換の直前の年設定が有効になります。

年設定がない場合の日付項目変換のデフォルトは、入力／出力ともに **W30** となり、入出力データの年を1930～2029年とみなします。

[区切記号]

日付データ項目を出力する際の日付区切り記号を、つぎの3つの中から設定します。入力に、SLASH、または、HYPHEN、または、PERIODと入力します。

<u>指 定 文 字</u>	<u>日付区切り記号</u>	<u>デ ー タ 例</u>
SLASH	／（スラッシュ）	1998／12／31
HYPHEN	－（ハイフン）	1998－12－31
PERIOD	．（ピリオド）	1998．12．31

日付区切り設定は日付項目変換が実行された時に適用になり、複数の日付区切り設定がなされている場合は、日付項目変換の直前の日付区切り設定が有効になります。

日付区切り設定がない場合の日付項目変換のデフォルトは、**SLASH** となり、日付区切り記号を“／”にします。

[日付]

日付データ項目を変換します。

入力と出力は、必ず日付マスクの形式で指定し、省略はできません。日付マスクの指定方法については、「操作の基礎」の章で説明しました。たとえば、

入力側に、 `yyyy-mm-dd`
 出力側に、 `yy m m d d` のように指定すると、

コード変換後に、入力側10桁の日付データ項目を、出力側6桁の日付データ項目に編集します。その際に、年設定、日付区切り設定が適用になります。

日付マスクにはオプションが指定可能です。入力、または出力コードがUCS-2の時、-D i s p指定では1桁2バイト単位で処理します。それ以外の場合は1桁1バイト単位で処理します。これが省略値です。-A l p h a指定ではコードに関係なく1桁1バイト（ゾーン形式）で処理します。

[ゾーン・表示]

Windows COBOLのゾーン形式数値項目を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、ピクチャは `s4.1` と指定します。なお、ピクチャは省略できません。

出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。たとえば、ピクチャ=`s4.1`、出力幅=省略という指定で出力コードがUCS-2以外の場合、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[表示・2進]

Windowsの文字形式数値項目を、ホストの2進形式整数・小数項目に変換します。

入力幅は、バイト数で指定します。省略すると、入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

まとめると、プリント形式からの変換の場合は

入力幅=10、2進キャスト/ピクチャ=i4s4.1 のような指定になり、

デリミタ形式からの変換の場合は

入力幅=省略、2進キャスト/ピクチャ=i4s4.1 のような指定になるのがふつうです。ただし、入力幅が省略値を超えるときは、

入力幅=40、2進キャスト/ピクチャ=i8u15.2 のように、ダミーの入力幅を指定します。

[ゾーン・2進]

WindowsのCOBOLのゾーン形式数値項目を、ホストの2進形式整数・小数項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、入力ピクチャは **s4.1** と指定します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

入力ピクチャも2進キャストにつづくピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、入力ピクチャ=省略、2進キャスト/ピクチャ=i4s4.1という指定は、入力ピクチャ=s4.1を指定したのと同じ意味になります。

[表示へ]

入力した数値定数、またはシステム変数の値を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

入力値は省略できません。

出力幅は省略できます。出力幅を省略すると、「入力した値を正確に出力するために必要な幅」で出力されます。たとえば、入力値 = [-123.4]、出力幅 = 省略という指定は出力幅 = 1 (符号) + 3 (整数部桁数) + 1 (小数点) + 1 (小数部桁数) = 6 を指定したのと同じ意味になります。(出力コードが UCS-2 の場合は 12)

[ゾーンへ]

入力した数値定数、システム変数の値を、出力ピクチャに従ってホストの COBOL のゾーン形式数値項目に変換します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、-123.4 という定数を入力値として指定し、5 バイトの符号つきゾーン形式の項目に記録するとすれば、出力ピクチャは **s4.1** と指定します。

入力値、出力ピクチャともに省略できません。

[パックへ]

入力した数値定数、システム変数の値を、出力ピクチャに従ってホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という定数を入力値として指定し、3バイトの符号つきパック形式の項目に記録するとすれば、出力ピクチャは **s4.1** と指定します。

入力値、出力ピクチャともに省略できません

[2進へ]

入力した数値定数、システム変数の値を、出力ピクチャに従ってホストの2進形式整数・小数項目に変換します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という定数を入力値として指定し、4バイトの符号つき2進形式の項目に記録するとすれば、出力2進キャスト／ピクチャは **i4s4.1** と指定します。

入力値、出力2進キャスト／ピクチャともに省略できません。

[Alpha^]

入力した数値定数、またはシステム変数の値を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

入力値は省略できません。

出力幅は省略できます。出力幅を省略すると、「入力した値を正確に出力するために必要な幅」で出力されます。たとえば、入力値 = [-123.4]、出力幅 = 省略という指定は出力幅 = 1 (符号) + 3 (整数部桁数) + 1 (小数点) + 1 (小数部桁数) = 6 を指定したのと同じ意味になります。

[Ank化]

Windows側（入力）文字列をできる限りANK（半角）に変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。

プリント形式の場合

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

出力幅は式による指定もでき、式のなかでは、

\$ ホスト側（出力）のレコード長

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のANK項目のおわりにスペースが詰められます。

デリミタ形式の場合

ふつう、入力幅は省略し、出力幅だけを10進のバイト数で指定します。出力幅は式による指定もでき、式のなかでは

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

[漢字化]

Windows側（入力）文字列をできる限り漢字（全角）に変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。

プリント形式の場合

入力幅は、10進のバイト数で指定します（漢字の文字数ではありません）。式による指定もでき、そのなかでは、

*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。

項目長を変更したいときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、漢字項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、ホスト側の漢字項目のおわりに漢字変換方式に設定されている漢字スペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

\$ ホスト側（出力）のレコード長

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

デリミタ形式の場合

ふつう入力幅は省略し、出力幅だけを10進のバイト数で指定します。出力幅は式による指定もでき、そのなかでは、

*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

[Alpha・ゾーン]

Windowsの文字形式数値項目を、ホストのCOBOLのゾーン形式数値項目に変換します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が15バイトを超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**123.4** という数字を5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が15バイトを超えるときは、

入力幅=20、ピクチャ=u15.2 のように、ダミーの入力幅を指定してください。

[Alpha・パック]

Windowsの文字形式数値項目を、ホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が15バイトを超えることは少ないので、省略するほうがふつうです。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**123.4** という数字を3バイトの符号つきパック形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。**123.4** という数字を3バイトのBCD形式の項目に記録するとすれば、ピクチャは **b5.1** と指定します。なお、ピクチャは省略できません。

まとめると、プリント形式からの変換の場合は、

入力幅=8、ピクチャ=s4.1 のような指定になり、

デリミタ形式からの変換の場合は、

入力幅=省略、ピクチャ=s4.1 のような指定になるのがふつうです。

ただし、入力幅が15バイトを超えるときは、

入力幅=20、ピクチャ=u15.2 のように、ダミーの入力幅を指定してください。

[Alpha・2進]

Windowsの文字形式数値項目を、ホストの2進形式整数・小数項目に変換します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が15バイトを超えることは少ないので、省略するほうがふつうです。

2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト／ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

まとめると、プリント形式からの変換の場合は

入力幅=10、2進キャスト／ピクチャ=i4s4.1 のような指定になり、

デリミタ形式からの変換の場合は

入力幅=省略、2進キャスト／ピクチャ=i4s4.1 のような指定になるのがふつうです。ただし、入力幅が15バイトを超えるときは、

入力幅=20、2進キャスト／ピクチャ=i8u15.2 のように、ダミーの入力幅を指定します。

[ゾーン・Alpha]

Windows COBOLのゾーン形式数値項目を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする

+整数部桁数

+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値が使われます。たとえば、ピクチャ=**s4.1**、出力幅=省略という指定は、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

(6) 引用符はずし／くくりの指定

デリミタ形式からの変換の場合には、項目が引用符（“ ” や ‘ ’）でくくられていることがあります。その引用符は無視します。また、引用符でくくったなかにデリミタがあっても、ただの文字データとして扱います。

ホストがU n i x、W i n d o w s の場合、デリミタ形式に変換するときは、変換後の文字列を引用符でくくることができます。ただし、プリント形式への変換のときは、この引用符くくりの機能は無効になります。

引用符にはつぎの3種類があります。

両くくり	一つの項目を単独でくくる
左くくり	複数の項目を一つにくくるときに、引用符の始まりを表す
右くくり	複数の項目を一つにくくるときに、引用符の終わりを表す

市販ソフトの入力用には、文字項目だけを引用符でくくることが多いのですが、引用符を使わないソフトもあれば、すべての項目を引用符でくくるソフトもあります。

◆参考...

たとえば、デリミタ形式のうちK3フォーマットと呼ばれるものは、

各項目はコンマで区切る	
数値はそのまま（位取りのコンマ不可）	
文字列は引用符でくくる	というルールになっています。

(7) デリミタ検出／挿入の有無の指定

デリミタ形式からの変換の場合には、デリミタ検出を無効にすることができます。デリミタ検出を無効にすると、その次の項目と合わせて1項目とみなします。

ホストがUnix、Windowsの場合、デリミタ形式に変換するときは、デリミタ挿入を無効にすることができます。デリミタ挿入を無効にすると、その次の項目と合わせて1項目とみなします。ただしプリント形式への変換のときは、このデリミタ有無の指定の機能は無効になります。

たとえばつぎのように、デリミタ形式から変換するときに、ある項目 (**Ank : 20**) の前に定数 (**Data=**) を挿入するときなどに、デリミタ検出を無効にします。

```

      :
Ank  ['Data='] &
Ank  : 20
      :
```

(8) 漢字イン／アウトの指定

ホストが汎用機・オフコンの場合、項目の前後に、漢字イン (KI)・漢字アウト (KO) を挿入することができます。

これは、漢字項目に漢字イン (KI)・漢字アウト (KO) を指定するのが、ふつうの使い方です。この指定とAnk・漢字まじり項目変換は、似ているところもありますが別物です。ご注意ください。

1 : 漢字イン、3 : 漢字イン+アウトを指定した場合、さらに10進数数値 (0~255の範囲) を入力することもできます。東芝方式のANK・漢字項目化するために「長さバイト」を付加する場合に指定します。

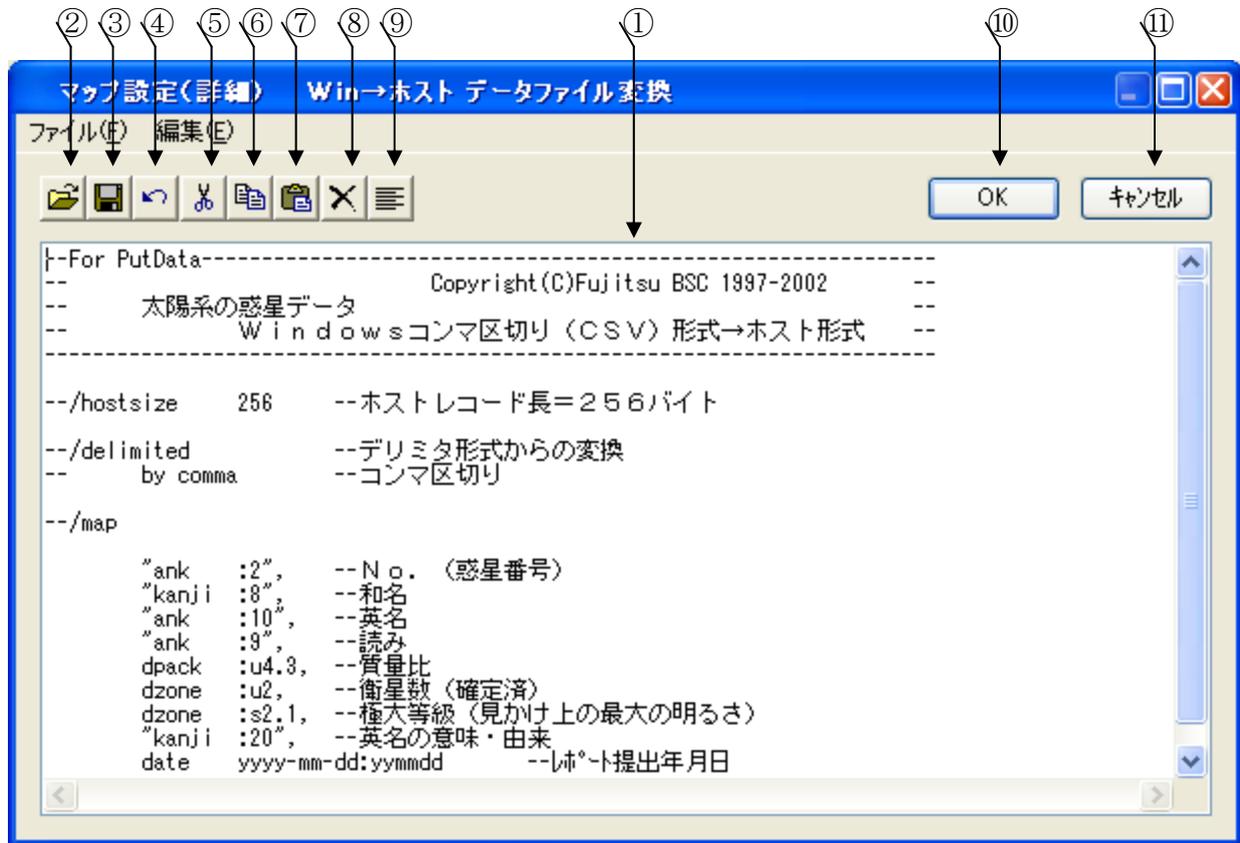
(9) 改行コード挿入の指定

ホストがUnix、Windowsの場合、任意のところに改行コードを挿入する (改行コードで項目を区切る) こともできます。改行コード挿入には、つぎの2種類があります。

通常 出力がプリント形式データファイルの時は、指定が無効になる
強制 出力がプリント形式データファイルの時も、指定が有効になる

出力がデータファイルのときに、自動的にレコード末尾に改行コードが付加される機能とは別のものですから、混同しないでください。

<マップ設定（詳細）ウィンドウ>



①マップ設定の記述をするエディットボックスです。

②～⑨はファイル（F）、編集（E）のメニューの機能がボタン化されています。

②既存のパラメータファイル等を読みます。

③①の内容をパラメータファイルへ保存します。

④直前の編集作業が無効になり、元に戻ります。

⑤選択した文字列が切り取られ、カットバッファに入ります。

⑥選択した文字列がカットバッファに入ります。

⑦⑤／⑥の操作でカットバッファに入った内容を、カーソルがある位置へ貼りつけます。

カーソルがある位置以降に文字列があれば、挿入になります。

⑧選択した文字列が削除されます。

⑨①の文字列をすべて選択状態にします。

⑩設定が完了したら、OKボタンをクリックします。

⑪キャンセルボタンをクリックすると、すべての入力／編集作業が無効になります。

◆注意 --- マップ設定（詳細）の記述内容について

マップ設定（詳細）のエディットボックスの中に記述する内容は、レコードレイアウト（/map を含まない /map 以降）のみの記述となります。既存のパラメータファイルを読み込むと、それ以外の記述（/delimited～、/map 等）は自動的にコメントになります。記述に誤りがある状態でOKボタンをクリックすると、エラーとなる場合があります。修正してから再度OKボタンをクリックしてください。/map 以降の文法については、「コマンド編」の「コマンド型の実行」の章を参照してください。

例) マップ設定（詳細）にパラメータファイル“PNPUTCSV.P”を読み込む

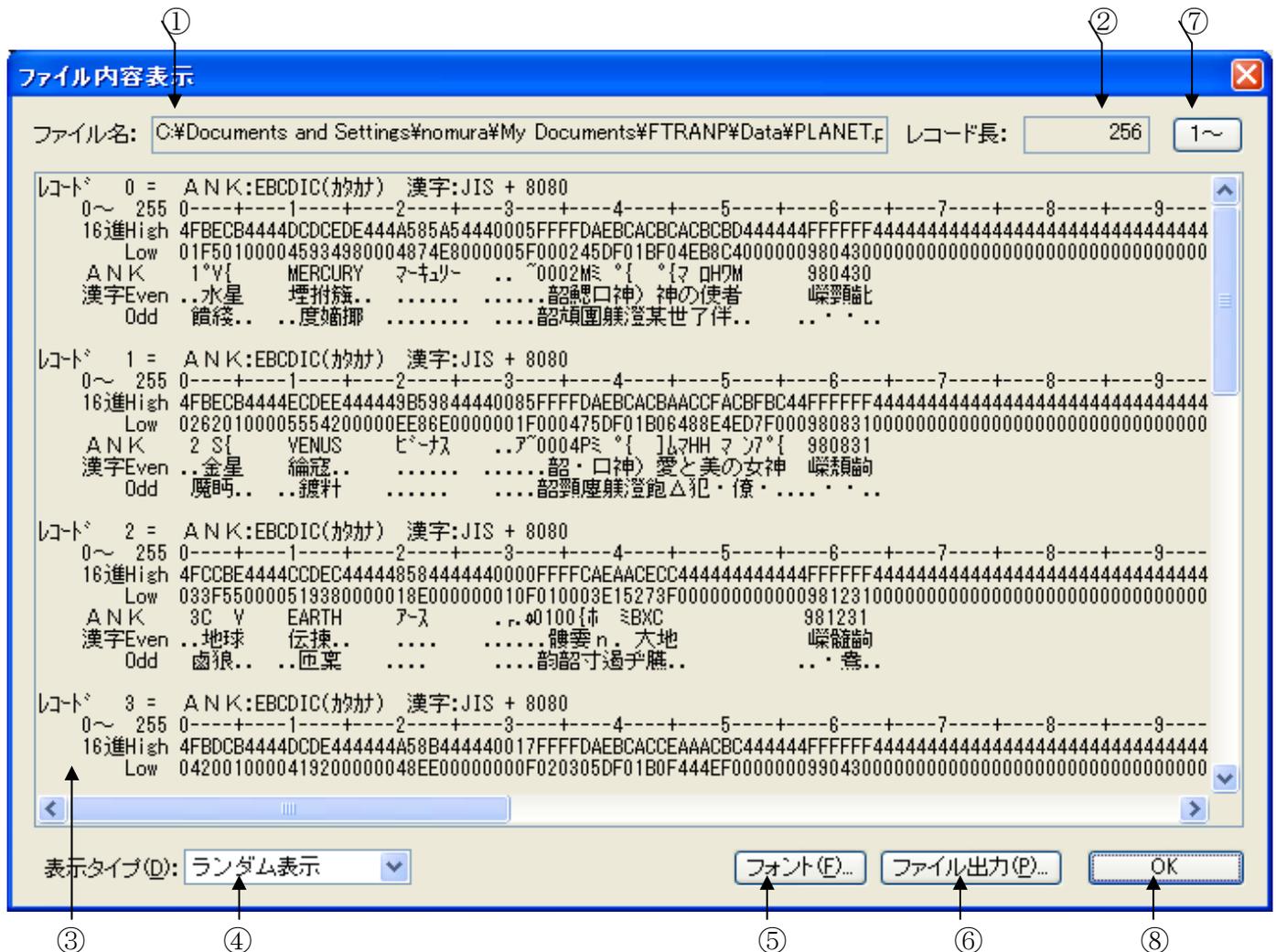
<pre> --- --/hostsize 256 --ホストレコード長=256バイト --/delimited --デリミタ形式からの変換 -- by comma --コンマ区切り --/map "ank :2", --No. (惑星番号) "kanji :8", --和名 "ank :10", --英名 "ank :9", --読み dpack :u4.3, --質量比 dzone :u2, --衛星数 (確定済) dzone :s2.1, --極大等級 (見かけ上の最大の明るさ) "kanji :20", --英名の意味・由来 date yyyy-mm-dd:yymmdd --レポート提出年月日 </pre>	<p>コメント になる行</p> <p>変換時に、 使われる行</p>
--	---

■Win→ホストデータファイル変換の実行

変換ボタンをクリックすると、Win→ホストデータファイル変換が始まります。

●ファイル内容表示ウインドウ

変換結果表示 (V) のチェックボックスをONにし、Win→ホストファイル変換を実行すると、実行ウインドウを閉じた後に下記のファイル内容表示ウインドウが開き、変換先のファイルの中身を確認することができます。



- ①変換先のファイル名が表示されます。
- ②変換先のファイルのレコード長が表示されます。(ホストが汎用機・オフコンのみ)
- ③変換先のファイルの中身が、③で選択されている表示タイプで出力されます。
- ④③の表示タイプをランダム表示、テキスト表示、テキストのみ表示に切り替えます。
- ⑤フォント (F) ボタンをクリックして、③の出力文字フォントを切り替えることができます。
- ⑥ファイル出力 (P) ボタンをクリックして、③の内容をファイルに保存できます。
- ⑦桁位置のガイド表示を、0 起点 (先頭を 0 として数える：デフォルト)、1 起点 (先頭を 1 として数える) のどちらかに切り替えることができます。
- ⑧OK ボタンをクリックすると、ファイル内容表示ウインドウを閉じます。

ファイル内容表示は変換結果のプレビュー表示機能のため、表示できる容量に制限があります。全ての内容が表示されている場合はデータの最後に、

----- ファイルの最後まで表示しました. -----

全ての内容が表示できなかった場合には、

----- ファイルの途中で表示を切りました. -----

と表示されます。

■注意事項

漢字があるときの注意

漢字項目変換やA n k・漢字まじり項目変換を行うときは、あらかじめ

変換設定の漢字変換方式設定で、適当な漢字変換方式を割り当てておく

のを忘れないでください。また、入力幅、出力幅は漢字データについても

バイト単位で指定

します。漢字の文字数ではないことに注意してください。

改行コードがないと、1件だけの変換になる

W i n d o w s ファイルの各レコードの末尾に改行コードがついていないと、先頭の1件だけ変換されます。この場合は、W i n →ホストランダムファイル変換を使うべきです。

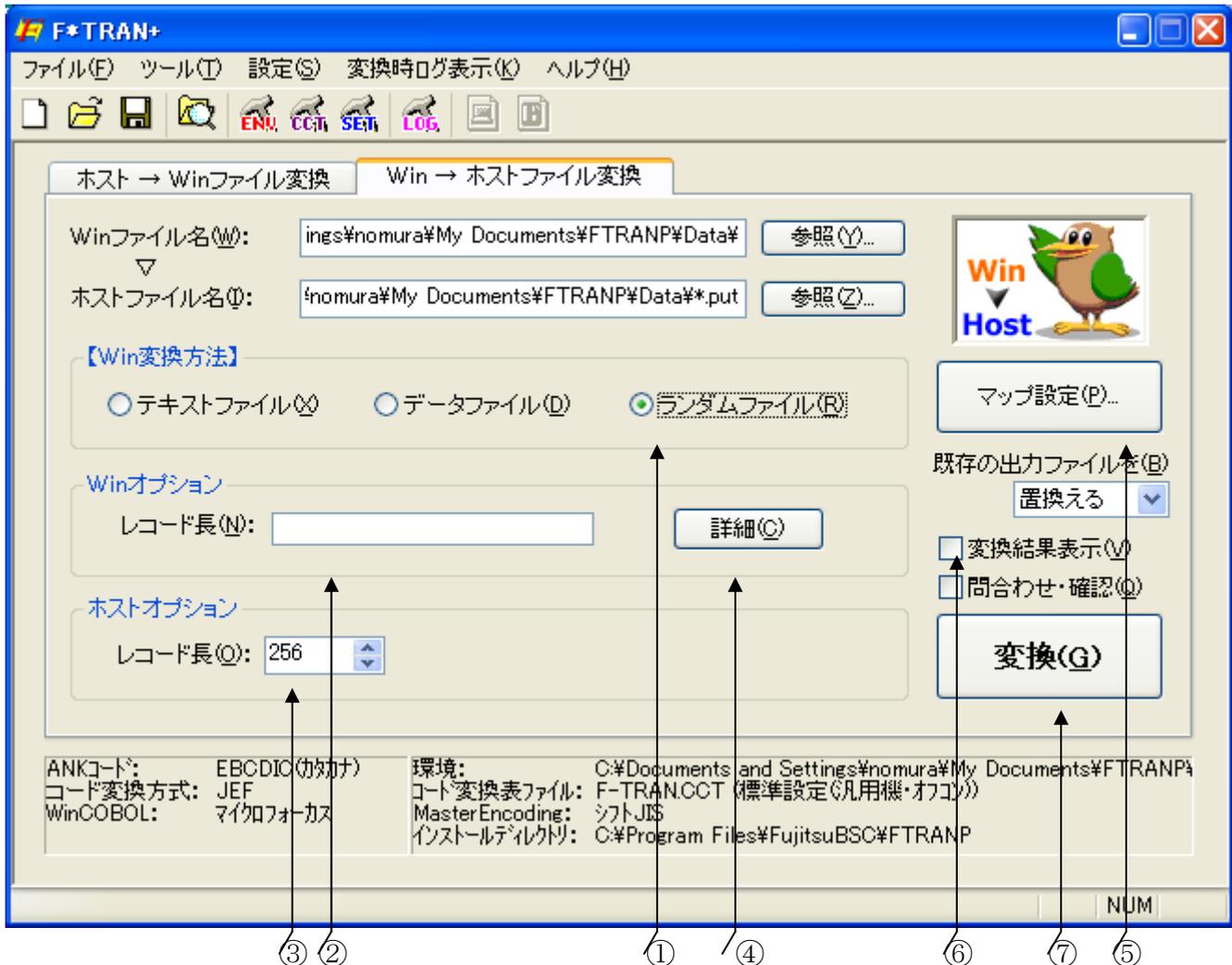
その他の注意事項

「W i n →ホストファイル変換」の節を参照してください。

4.9 Win→ホストランダムファイル変換

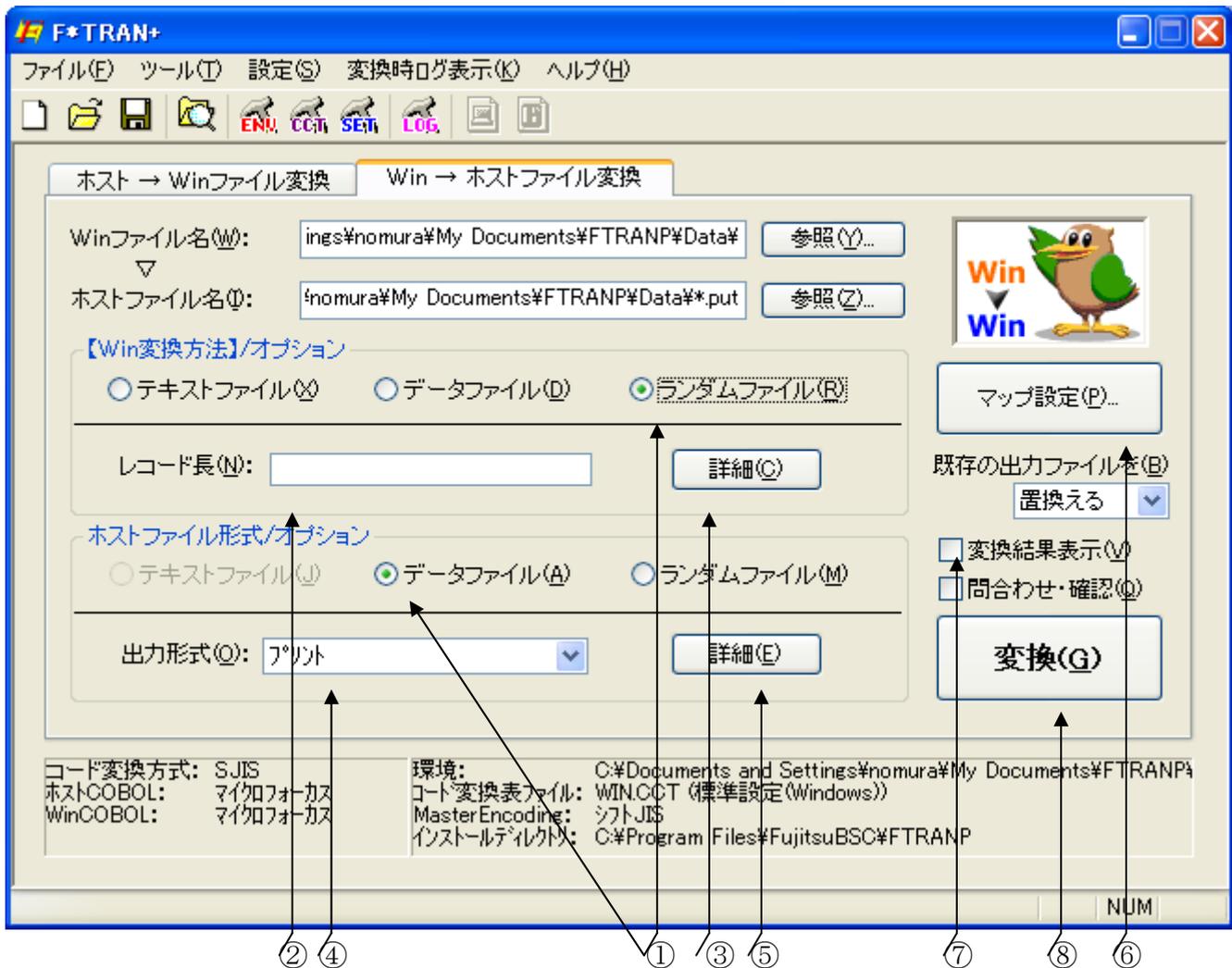
Win→ホストランダムファイル変換の操作方法について説明します。

●ホストが汎用機・オフコンの場合

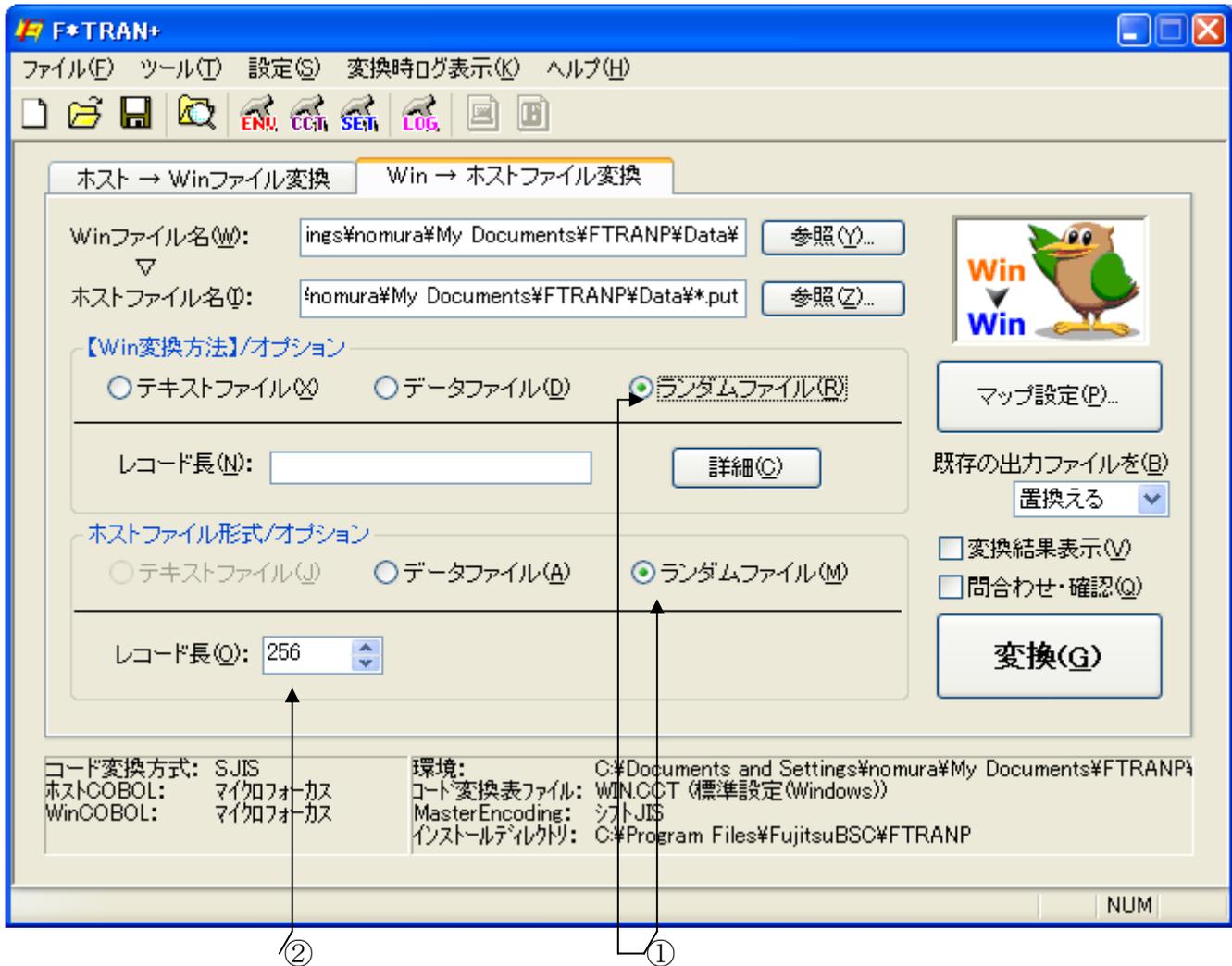


- ①変換方法のランダムファイル (R) をクリックし、Win→ホストランダムファイル変換を選択します。
- ②入力レコード長を指定するオプションです。
- ③ホストファイルのレコード長を指定します。
- ④Winファイルの詳細オプションボタンです。Windows側エンコーディングの設定ができます。
- ⑤マップ設定 (レコードレイアウト設定) をするオプションボタンです。
- ⑥変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑦変換を実行するボタンです。このボタンをクリックすると、変換が始まります。

●ホストがUnix、Windowsの場合



- ① Win変換方法のランダムファイル (R) をクリックし、Win→ホストランダムファイル変換を選択します。ホストファイル形式は、データファイル (A)、ランダムファイル (M) の選択ができますが、上の例はホストファイル形式がデータファイル (A) です。
- ② 入力レコード長を指定するオプションです。
- ③ Winファイルの詳細オプションボタンです。Windows側エンコーディングの設定ができます。
- ④ 出力ファイル形式を指定するオプションです。
- ⑤ ホストファイルの詳細オプションボタンです。EOF付加の選択ができます。
ホストがWindowsの場合のみ有効です。ホストがUnixの場合はありません。
- ⑥ マップ設定 (レコードレイアウト設定) をするオプションボタンです。
- ⑦ 変換後に、変換先のファイルの中身を表示ウインドウに出力するチェックボックスです。
- ⑧ 変換を実行するボタンです。このボタンをクリックすると、変換が始まります。



① Win変換方法のランダムファイル (R) をクリックし、Win→ホストランダムファイル変換を選択します。ホストファイル形式は、データファイル (A)、ランダムファイル (M) の選択ができますが、上の例はホストファイル形式がランダムファイル (M) です。

② ホストファイルのレコード長を指定します。

■オプションの指定

●入力レコード長オプション

Windows側（入力）のレコード長を指定します。Windowsレコード長は、通常、10進数で指定します。式による指定もでき、そのなかでは、

\$ ホスト側（出力）のレコード長 という特殊変数が使用できます。

ホストファイルのレコード長を基準にして、Windowsファイルのレコード長を決めることができます。たとえば、

\$+2 と指定すれば、

「ホストファイルのレコード長に+2したものをWindows側のレコード長にせよ」という意味になります。

入力レコード長オプションでWindowsレコード長を指定できることは、Win→ホストランダムファイル変換の性格をよく反映しています。

ゾーン形式とパック形式の変換、日付データの変換をサポートしているため、レコード長を増減する機能がないと困るからです。また、不要な項目がいくつもあるときや、予備領域が多すぎるときはレコード長を縮めたいものです。

◆注意 ---- Windows自身には「レコード長」の概念がない

Windows自体にはファイルごとに登録されたレコード長というものはありません。そのため、かなり自由が利きます。アプリケーションがファイルを扱う論理的な単位を、ここではWindows側のレコード長、あるいはWindowsレコード長と呼んでいます。

●ホストファイル・レコード長指定 <ホストファイルが固定長の場合のみ>

ホストファイルのレコード長を1～32767の範囲の10進数で指定します。デフォルトは、**256バイト**です。レコード長の指定が間違っていると正しいデータ変換が行われませんので、この指定は極めて重要です。

●出力形式オプション <ホストファイルが可変長の場合のみ>

データとしてのテキストファイルにはいくつもの形式があるので、どの形式にするかを指定します。つぎの、7種類の中から選択します。

プリント	-----	プリント形式に変換する
デリミタ (コンマ	・ 圧縮)	---- デリミタ形式 (コンマ) に変換、圧縮あり
デリミタ (タブ	・ 圧縮)	---- デリミタ形式 (タブ) に変換、圧縮あり
デリミタ (スペース	・ 圧縮)	---- デリミタ形式 (スペース) に変換、圧縮あり
デリミタ (コンマ	・ 非圧縮)	---- デリミタ形式 (コンマ) に変換、圧縮なし
デリミタ (タブ	・ 非圧縮)	---- デリミタ形式 (タブ) に変換、圧縮なし
デリミタ (スペース	・ 非圧縮)	---- デリミタ形式 (スペース) に変換、圧縮なし

プリントを指定すると、デリミタ (区切り文字) なしで変換され、プリント形式 (固定長のテキストファイル [SDF形式]) になります。これがデフォルトです。

デリミタを指定すると、デリミタ形式 (区切り文字付きのテキストファイル) に変換されます。デリミタの種類によってさらに細かい形式が決まります。デリミタは、

コンマ区切り形式 (CSV形式、K3形式)

タブ区切り形式 (TAB=09H)

スペース区切り形式 (SP=20H)

の3つのなかから指定できます。

通常は、コンマ区切り (CSV) 形式への変換が一般的です。さらに、

変換後に圧縮をかける (可変長になる。デフォルト)

変換後に圧縮しない (固定長のまま)

を指定できます。

を指定できます。圧縮をかけると本来不要なスペース、つまり、

レコードの先頭・末尾のスペース

デリミタの前後のスペース

引用符の前のスペース

が削除されます。

圧縮をかける機能があるのは、つぎのような理由があるためです。

変換後のWindowsファイルの容量を、大幅に減らすことができる
一部の市販ソフトでは、不要なスペースがあるとうまくデータを読み込めない

●EOF付加指定 <ホストがWindowsで、可変長ファイルの場合のみ>

EOFコード（1AH）の扱いを指定します。

EOF付加のチェックボックスを、

ONにすると、EOFコードをつける指定

OFFにすると、EOFコードをつけない指定 となります。

EOF付加指定がONであれば、WindowsファイルのおわりにEOFコードをつけます。現在では少なくなりましたが、テキストファイルのおわりにEOFコードがついていないとエラーにするソフトがあります。その場合にも対処するための機能です。

EOF付加指定がOFFであれば、EOFコードはつけません。これがふつうだと思ってください。こちらがデフォルトです。

●エンコーディング

Windows側のエンコーディングを以下の4通りから選択します。

Master Encodingに従う（詳細は「4.12 環境設定」を参照）

シフトJIS

Unicode (UCS-2)

Unicode (UTF-8)

Unicodeの場合は以下の選択も可能です。

BOMつき

BOMなし

BOM自動

『BOM自動』の指定ではUCS-2ではBOMを検査し、UTF-8ではBOMを検査しません。

さらにUCS-2の場合は以下の選択も可能です。

`Little Endian`
`Big Endian`

●引用符くくり

(1) 引用符の種類

この設定はホストがUnix/Windowsで出力形式がデリミタ形式に指定されているときのみ有効です。引用符の種類それぞれの意味を説明します。

- なし 引用符でくくらずに変換内容を出力します。
- 単一引用符 単一引用符「'」（半角）」をくくる。
- 2重引用符 2重引用符「"」（半角）」をくくる指定。

「なし」の指定の場合は、Map文で引用符くくりの指定がしてあっても無視します。

◆注意 ---- Map文の引用符記号は、常に二重引用符「"」

「引用符の種類」については「単一引用符」指定時も、Map文において引用符くくりを使う記号は、2重引用符「"」（半角）」ですので注意してください。

◆注意 ---- 引用符の種類指定は一種類のみ

単一引用符（'）と2重引用符（"）の両方を混在してくくることはできません。必ずどちらか一方でくくられます。

(2) データとしての引用符の扱い

- 削除 項目のデータとしての引用符は削除する指定。
- 空白化 項目のデータとしての引用符は半角スペースに変換する指定。
- 2連化 項目のデータとしての引用符は、2連化して出力する指定。

2連化を指定した場合、変換後データの項目長が変換前と変わりますが、それについては「2連化時の優先処理」で指定します。

◆注意 ---- 「データとしての引用符の扱い」のデータ内引用符は「引用符の種類」に依存

「データとしての引用符の扱い」ではデータ内容として引用符が存在した時の扱いを指定しますが、その「引用符」というのは「引用符の種類」で指定したものに限りません。

変換処理で使用する引用符として、2重引用符を指定した場合は、単一引用符は空白化などの処理の対象になることはありません。

(3) 2連化時の優先順位

引用符の2連化によって入力データ長に対する変換後データ長が膨れ上がった場合の処理を指定します。

出力幅優先 出力幅を優先して、データの後方部分を切り捨てる指定。

データ優先 変換後の長さそのままを出力する指定。

「2連化時の優先順位」は「データとしての引用符の扱い」において「2連化」を指定していない場合には有効になりません。

◆注意 ---- 「2連化時の優先順位」にてデータ優先をした時の出力幅について

「2連化時の優先順位」で「データ優先」を指定すると、M a p 文で出力幅指定を省略した時だけでなく出力幅指定をしていたとしてもそれを無視して変換します。下は変換処理時の実際の出力幅です。

出力幅指定をした場合 … M a p 文指定の出力幅+引用符の長さ

出力幅指定を省略した場合 … M a p 文指定の入力幅+引用符の長さ

●マップ設定オプション

このマップ設定オプションで、項目別の細かい変換方法を指示します。本来なら、自動的に項目を認識して変換ができると便利です。しかし、ホストの、とくにCOBOLのデータには、データ自身に桁数や小数点位置の判断に必要な情報が含まれていない、という特性があります。そのため自動変換は原理的に不可能なのです。

マップ設定のボタンをクリックすると、つぎのウインドウが開きます。



マップ設定には、

- 簡易設定 —— 設定ウインドウに従って、必要な値を入力するだけで設定ができます。通常は、こちらの機能だけで間に合う場合がほとんどです。
- 詳細設定 —— フリーフォーマットで、マップ設定を記述するエディタモードです。簡易設定だけでは設定できない特殊な設定をする場合、あるいは、既存のパラメータファイル等を流用したい場合等に使用します。マップ設定の文の書き方を覚えなくては設定できません。

の2通りの方法があります。

◆注意 ---- **簡易設定と詳細設定の組み合わせ**

簡易設定で設定した値は、詳細設定で読み込むことができるテキスト形式に直されます。よって、簡易設定をした後の項目に、詳細設定で特殊設定をつけ足すこともできます。

◆注意 ---- **詳細設定→簡易設定の順で設定した場合の制約**

詳細設定→簡易設定の順で設定した場合は、原形が変わってしまう場合があります。これは、詳細設定で設定した項目が簡易設定で読み込まれることにより、スリム化されてしまうからです。ただし、原形が変わってしまっても、設定内容が変わるわけではありません。

◆注意 ---- **入力幅、出力幅の指定について**

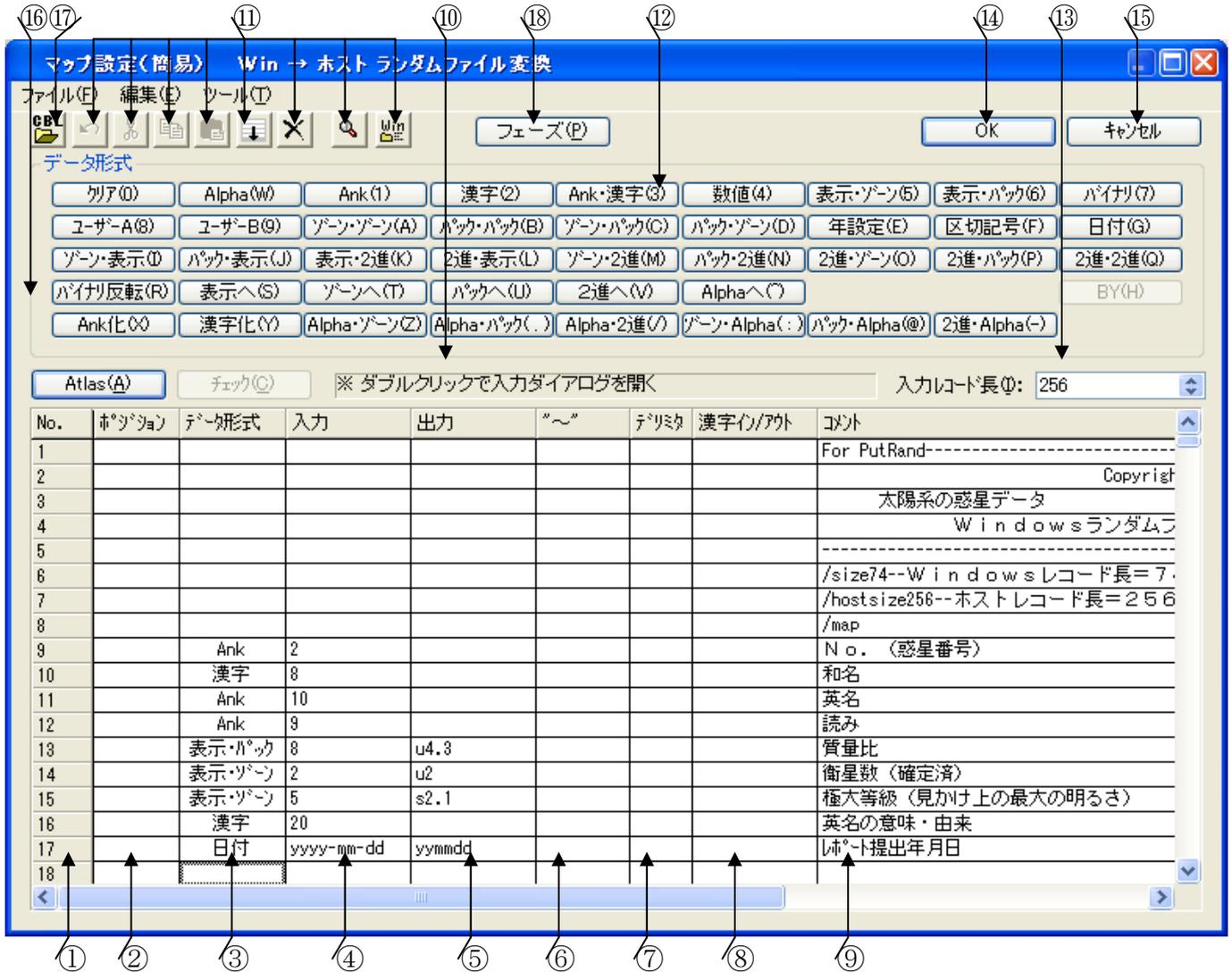
入力側や出力側の文字コードによって、入力幅、および出力幅の指定には注意が必要です。

たとえばANKデータを変換する場合、入力がWindowsでエンコーディングがUCS-2の場合は文字数×2の入力幅を指定する必要があります。

出力幅が省略された場合の省略値は入力コードと出力コードから入力幅に指定された値を最小として自動的に計算して求めています。ただし、漢字とANKが混ざった項目に対して漢字項目として変換した場合（ANK文字が全角化される）や、汎用機・オフコンの半角カナをANK変換でUTF-8に変換する場合など、明確に出力幅を指定しないと出力データが途中で切れてしまう場合もあります。

文字コードとしてUnicodeを指定する場合や半角カナ文字がある場合などは、入力幅や出力幅の指定には十分注意する必要があります。

<マップ設定（簡易）ウィンドウ>



- ①No. は、1～2000です。ここは非入力項目で、行選択の時にクリックします。
- ②項目の桁位置等を入力します。通常、項目を組み替えるとき以外は入力しません。
ダブルクリックすると、入力サブウィンドウが開きます。
- ③項目のデータ形式を指定します。
ダブルクリックすると、入力サブウィンドウが開きます。
- ④項目の入力幅を入力します。
Ank / 漢字 / Ank・漢字 / バイナリの場合、定数を指定することもできます。
Alphaへ / 表示へ / ゾーンへ / パックへ / 2進への場合、数値定数かシステム変数を入力します。
- ⑤項目の出力幅を入力します。出力幅＝入力幅の場合は省略します。
データ形式がゾーン / パック / 2進指定の場合、ピクチャまたは2進ピクチャで指定します。

- ⑥ホストがUnix、Windowsの場合、デリミタ形式に変換するときに項目を引用符でくくる指定ができます。
- ⑦ホストがUnix、Windowsの場合、デリミタ形式に変換するときにデリミタ挿入を無効にする指定ができます。
- ⑧ホストが汎用機・オフコンの場合は、項目の前後に漢字イン／アウトを挿入する指定です。ホストがUnix、Windowsの場合は、改行コードを挿入する指定です。
- ⑨コメントの入力ができます。
- ⑩セルポインタの入力ガイダンスメッセージが出力されるフィールドです。誤入力をした場合のエラーメッセージも出力されます。
- ⑪は編集 (E)、ツール (T) のメニューの機能がボタン化されています。



直前の編集作業が無効になり、元に戻ります。



1行または複数行を選択 (①の操作) をした後に、このボタンをクリックすると、選択した項目の内容が切り取られ、カットバッファに入ります。



1行または複数行を選択 (①の操作) をした後に、このボタンをクリックすると、選択した項目の内容がカットバッファに入ります。



カットバッファに入った内容を、セルポインタがある行へ貼りつけます。すでに、セルポインタ以降の行に設定項目がある場合は、挿入になります。



セルポインタがある行以降が1行ずつ下がります。通常は行挿入をする場合に使用します。



選択した項目の内容が削除されます。



変換プレビュー (変換前／変換後のデータ確認) ができます。



Winファイルエディタが起動します。

Windowsファイルの中身を見ること (編集も可能) ができます。

- ⑫セルポインタが選択項目にある場合に、選択肢のボタンが表示されるエリアです。
- ⑬入力レコード長オプションの指定を変更できます。
- ⑭設定が完了したら、OKボタンをクリックします。
- ⑮キャンセルボタンをクリックすると、すべての入力／編集作業が無効になります。
- ⑯マルチレコードの設定ができるAtlasモードに切り替えるボタンです。*1
- ⑰COBOLのソースファイル (Copy句) を読み込むボタンです。*1
- ⑱フェーズボタンをクリックすると、フェーズを設定するウィンドウが開きます。*1

*1) 詳細は、マルチレコード編のマニュアルを参照してください。

◆注意 —— マップ未設定時のデータ変換動作

マップ未設定の場合は、すべてをANKデータとみなしてデータ変換を行います。

<マップ設定（簡易）サブウィンドウ>

ポジションの欄をダブルクリックすると、つぎのサブウィンドウが開きます。入出力桁位置や入出力スキップの設定を簡便に行うことができます。

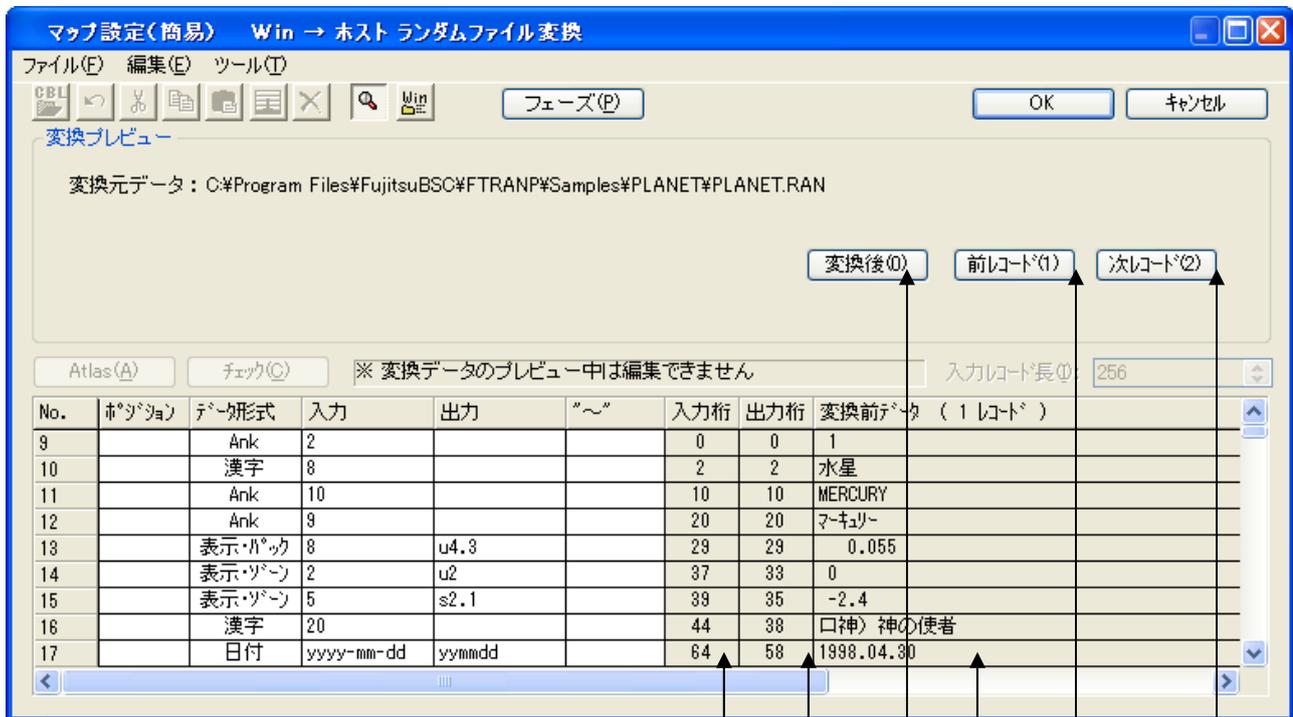


データ形式の欄をダブルクリックすると、つぎのサブウィンドウが開きます。データ形式から引用符はずし（漢字項目は漢字イン／アウト）までの設定を簡便に行うことができます。

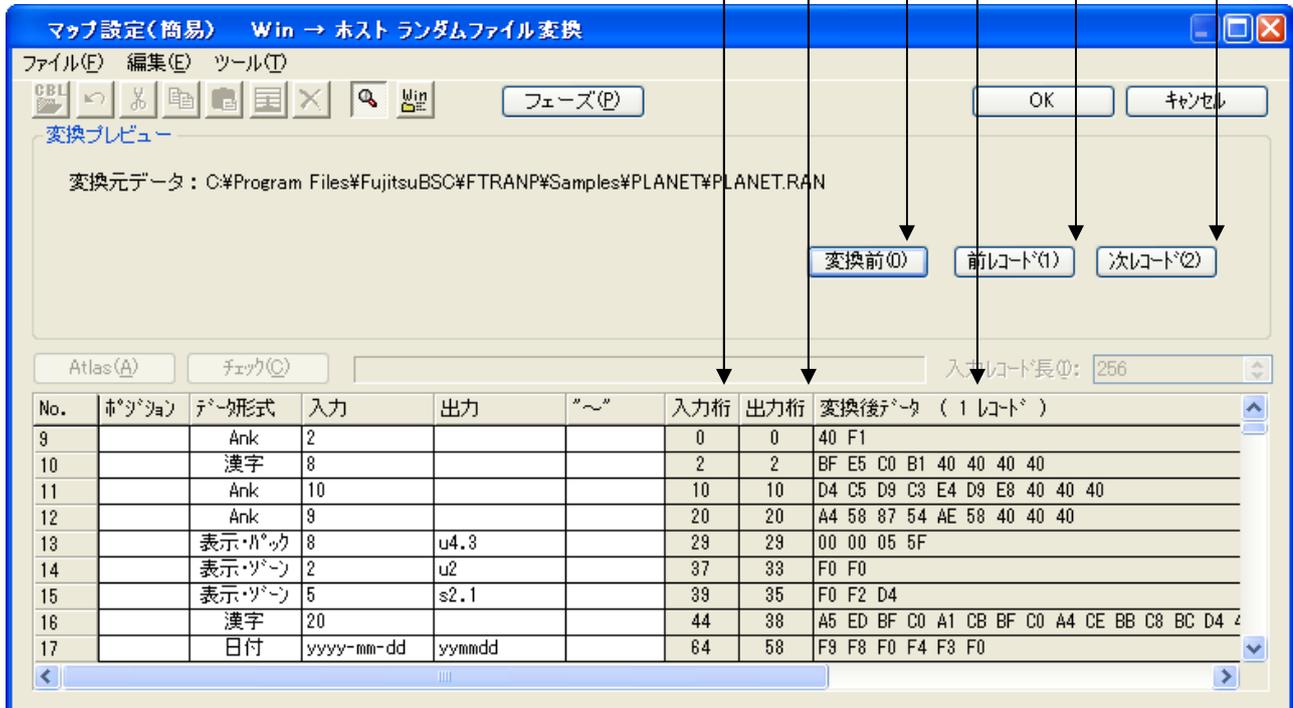


<マップ設定（簡易）・変換プレビュー機能>

変換プレビューボタンをクリックすると、マップ設定（簡易）のウィンドウはつぎのような状態になり、変換前／変換後のデータを確認することができます。



↑変換前データ表示 変換後データ表示↓ ③ ③ ① ④ ② ③



- ①変換前／変換後のデータ表示の切り替えを行うボタンです。④の表示が変わります。
- ②前レコード／次レコード表示のためのボタンです。
- ③ここには、入力レコード／出力レコードのポジション（桁）が表示されます。
- ④データ表示は、変換前が文字、変換後が16進、未定義個所が文字になります。

(1) ポジション（桁位置）の指定

変換対象にするWindows側（入力）の桁位置や、変換結果を書き込むホスト側（出力）の桁位置を、別の任意の位置に移動できます。現在、処理対象にしている桁位置を、この指定で強制的に変更できます。この機能を利用すると、項目の組み替えなどが簡単に実現できます。

入力桁位置は、ふつう10進数で指定します。式による指定もでき、そのなかでは、

\$ Windows側（入力）のレコード長
. Windows側（入力）の現在の桁位置 という特殊変数が使えます。

たとえば、

. -40 と指定すれば、 **今の入力桁位置から40バイト戻る**

という意味になります。

出力桁位置を移動することもできます。ふつう、

: 10進数

で桁位置を指定します。10進数の代わりに式による指定もでき、そのなかでは、

\$ ホスト側（出力）のレコード長
. ホスト側（出力）の現在の桁位置 という特殊変数が使えます。

たとえば、

:. -40 と指定すれば、 **現在の出力桁位置から40バイトバックせよ**

という意味になります。

◆注意 ---- 先頭を0桁目とする

F*TRAN+では、レコードの先頭を0桁目として数えます。

(2) ポジション（入力スキップ）の指定

Windows側（入力）レコードに不要な項目があるとき、それをスキップして変換できません。

スキップする幅は、 `^バイト数` で指定します。

たとえば、3バイト分スキップしたいなら、 `^3` と指定します。

バイト数は省略でき、省略すると1バイトとみされるので、

`^3` は `^^^` と指定したのと同じです。

スキップする幅は式による指定もでき、そのなかでは、

* `Windows側（入力）の残りバイト数` という特殊変数が使えます。

(3) ポジション（出力スキップ）の指定

入力スキップとは逆に、ホスト側（出力）に何桁か空きを作ることもできます。

スキップする幅は、 `__バイト数` で指定します。

たとえば、3バイト分スキップしたいなら、 `__3` と指定します。

バイト数は省略でき、省略すると1バイトとみされるので、

`__3` は `_____` と指定したのと同じです。

スキップする幅は式による指定もでき、そのなかでは、

* `ホスト側（出力）の残りバイト数` という特殊変数が使えます。

(4) データ形式

[Alpha]

Windows側のデータに関係なく1バイトごとにANK項目として変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。出力側のコード体系に関係なく、ANKコードの設定に従って1バイトデータに変換されて出力します。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ Windows側（入力）のレコード長
- * Windows側（入力）の残りバイト数 という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。入力に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のANK項目のおわりにスペース（20H/40H）が詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ ホスト側（出力）のレコード長
- * ホスト側（出力）の残りバイト数 という特殊変数が使えます。

[Ank]

ANK項目を変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

\$ Windows側（入力）のレコード長
*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

また、入力幅の代わりに、ANK形式の定数を指定することもできます。その場合、指定した定数はWindows側（入力）レコードにあったかのようにみなされ、その後コード変換されます。

入力値として、つぎの定数が使えます。

文字列定数 **文字列はANK形式**
汎用定数 **Space、LowValue、HighValue**

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。入力に定数を指定した場合は、出力幅の省略値は「**見た目の文字数×2**」になります。指定された定数は内部的にUCS-2コードなので、出力コードがUCS-2以外の場合は明示的に出力幅を指定する必要があります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のANK項目のおわりにスペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

\$ ホスト側（出力）のレコード長
*** ホスト側（出力）の残りバイト数** という特殊変数が使えます。

[漢字]

漢字項目を変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します（漢字の文字数ではありません）。式による指定もでき、そのなかでは、

\$ Windows側（入力）のレコード長
*** Windows側（入力）の残りバイト数** という特殊変数が使えます。

また、入力幅の代わりに、漢字形式の定数を指定することもできます。その場合、指定した定数はWindows側（入力）レコードにあったかのようにみなされ、その後コード変換されます。

入力値として、つぎの定数が使えます。

文字列定数 **文字列は漢字形式**
汎用定数 **Space**

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。入力に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、漢字項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、ホスト側の漢字項目のおわりに漢字変換方式に設定されている漢字スペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

\$ **ホスト側（出力）のレコード長**
***** **ホスト側（出力）の残りバイト数** という特殊変数が使えます。

[Ank・漢字]

ANK・漢字まじり項目を変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。ホストが汎用機・オフコンの場合、変換後、漢字文字列の前後にはKI/KOが挿入されます。この、KI/KO挿入のタイミングも漢字変換方式の設定できまります。そのため、オーバーフローの危険性があるので、出力幅の指定には注意しなければいけません。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

\$ **Windows側（入力）のレコード長**
***** **Windows側（入力）の残りバイト数** という特殊変数が使えます。

また、入力幅の代わりに、ANK・漢字まじり形式の定数を指定することもできます。その場合、指定した定数はWindows側（入力）レコードにあったかのようにみなされ、その後コード変換されます。

入力値として、つぎの定数が使えます。

文字列定数 **文字列はANK・漢字まじり形式**
汎用定数 **Space、LowValue、HighValue**

[表示・ゾーン]

Windowsの文字形式数値項目を、ホストのCOBOLのゾーン形式数値項目に変換します。

入力幅は、バイト数で指定します。省略すると入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、ふつうは明示的にバイト数を指定します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字が5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

[表示・パック]

Windowsの文字形式数値項目を、ホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。

入力幅は、バイト数で指定します。省略すると入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、ふつうは明示的にバイト数を指定します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字を3バイトの符号つきパック形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。**123.4** という数字を3バイトのBCD形式の項目に記録するとすれば、ピクチャは **b5.1** と指定します。なお、ピクチャは省略できません。

[バイナリ]

バイナリ変換（無変換）は、「コード変換を一切しない」という変換方法です。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ Windows側（入力）のレコード長**
- * Windows側（入力）の残りバイト数** という特殊変数が使えます。

また、入力幅の代わりに、バイナリ形式の定数を指定することもできます。その場合、指定した定数はWindows側（入力）レコードにあったかのようにみなされ、コード変換されずにそのまま出力されます。

入力値として、つぎの定数が使えます。

16進列定数、汎用定数（LowValue、HighValue）

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなされます。入力幅に定数を指定した場合は、出力幅の省略値は「正確な出力に必要な幅」になります。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、バイナリ項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のバイナリ項目のおわりに、ホストが汎用機・オフコンの場合はNUL (00H)が、ホストがUnix/Windowsの場合はスペース (20H)が詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ ホスト側 (出力) のレコード長
- * ホスト側 (出力) の残りバイト数 という特殊変数が使えます。

[ユーザーA], [ユーザーB]

ユーザーA/B変換は、利用者独自のバイト単位の変換処理が必要なときに、ユーザーA/B変換表を書き替えて利用します。ユーザーA/B変換には、Alpha項目変換の説明がほとんどそのまま当てはまります。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

- \$ Windows側 (入力) のレコード長
- * Windows側 (入力) の残りバイト数 という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ユーザーA/B項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のユーザーA/B項目のおわりにNUL (00H)が詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ ホスト側 (出力) のレコード長
- * ホスト側 (出力) の残りバイト数 という特殊変数が使えます。

[ゾーン・ゾーン]

Windows COBOLのゾーン形式数値項目を、ホストのCOBOLのゾーン形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
- 1 2 3 . 4 という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、出力ピクチャは **s 4 . 1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s 4 . 1という指定は、入力ピクチャ=s 4 . 1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s 4 . 1、出力ピクチャ=省略という指定は、出力ピクチャ=s 4 . 1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[パック・パック]

Windows COBOLのパック、またはBCD形式数値項目を、ホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
- 1 2 3 . 4 という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、出力ピクチャは **s 4 . 1** と指定します。BCD形式なら **b 5 . 1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s 4 . 1という指定は、入力ピクチャ=s 4 . 1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s 4 . 1、出力ピクチャ=省略という指定は、出力ピクチャ=s 4 . 1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[ゾーン・パック]

Windows COBOLのゾーン形式数値項目を、ホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
- 1 2 3 . 4 という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、出力ピクチャは **s 4 . 1** と指定します。BCD形式なら **b 5 . 1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s 4 . 1という指定は、入力ピクチャ=s 4 . 1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s 4 . 1、出力ピクチャ=省略という指定は、出力ピクチャ=s 4 . 1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[パック・ゾーン]

Windows COBOLのパック、またはBCD形式数値項目を、ホストのCOBOLのパック形式数値項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
- 1 2 3 . 4 という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、出力ピクチャは **s 4 . 1** と指定します。

入力ピクチャは省略できます。入力ピクチャを省略すると、「出力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=省略、出力ピクチャ=s 4 . 1という指定は、入力ピクチャ=s 4 . 1を指定したのと同じ意味になります。

出力ピクチャも省略できます。出力ピクチャを省略すると、「入力ピクチャ」と同じとみなされます。たとえば、入力ピクチャ=s 4 . 1、出力ピクチャ=省略という指定は、出力ピクチャ=s 4 . 1を指定したのと同じ意味になります。

なお、入力ピクチャと出力ピクチャを同時に省略することはできません。

[年設定]

日付データ項目を変換する際の、年の2桁（yy）と4桁（yyyy）の変換方式を設定します。Wnn、または、Snnの形式で設定します。Wnnの“W”はウィンドウ方式を、Snnの“S”はシフト方式を意味し、“nn”は00～99の数字で指定します。

入力側に **W30** と指定すれば、入力データの年を1930～2029年とみなし、出力側に **S25** と指定すれば、出力データの年下2桁を、-25します。

また、シフト方式（“Snn”指定）では、つぎの特殊指定ができます。

SShowa は、“S25”の指定と同じ（昭和通年方式）

SHeisei は、“S88”の指定と同じ（平成通年方式）

年設定は日付項目変換が実行された時に適用になり、複数の年設定がなされている場合は、日付項目変換の直前の年設定が有効になります。

年設定がない場合の日付項目変換のデフォルトは、入力／出力ともに **W30** となり、入出力データの年を1930～2029年とみなします。

[区切記号]

日付データ項目を出力する際の日付区切り記号を、つぎの3つの中から設定します。入力に、SLASH、または、HYPHEN、または、PERIODと入力します。

<u>指 定 文 字</u>	<u>日付区切り記号</u>	<u>デ ー タ 例</u>
SLASH	／（スラッシュ）	1998／12／31
HYPHEN	－（ハイフン）	1998－12－31
PERIOD	．（ピリオド）	1998．12．31

日付区切り設定は日付項目変換が実行された時に適用になり、複数の日付区切り設定がなされている場合は、日付項目変換の直前の日付区切り設定が有効になります。

日付区切り設定がない場合の日付項目変換のデフォルトは、**SLASH** となり、日付区切り記号を“／”にします。

[日付]

日付データ項目を変換します。

入力と出力は、必ず日付マスクの形式で指定し、省略はできません。日付マスクの指定方法については、「操作の基礎」の章で説明しました。たとえば、

入力側に、 `yyyy-mm-dd`
出力側に、 `yy m m d d` のように指定すると、

コード変換後に、入力側10桁の日付データ項目を、出力側6桁の日付データ項目に編集します。その際に、年設定、日付区切り設定が適用になります。

日付マスクにはオプションが指定可能です。入力、または出力コードがUCS-2の時、-D i s p指定では1桁2バイト単位で処理します。それ以外の場合は1桁1バイト単位で処理します。これが省略値です。-A l p h a指定ではコードに関係なく1桁1バイト（ゾーン形式）で処理します。

[ゾーン・表示]

Windows COBOLのゾーン形式数値項目を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、ピクチャは `s4.1` と指定します。なお、ピクチャは省略できません。

出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。たとえば、ピクチャ=`s4.1`、出力幅=省略という指定で出力コードがUCS-2以外の場合、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[パック・表示]

Windows COBOLのパック、またはBCD形式数値項目を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

パック形式では、整数部桁数+小数部桁数を奇数にしておくのが通例です。整数部の最上位桁に意味があるのかないのかは、半々の割合です。出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。例を示すと、ピクチャ=s4.1、出力幅=省略という指定で出力コードがUCS-2以外の場合、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[表示・2進]

Windowsの文字形式数値項目を、ホストの2進形式整数・小数項目に変換します。

入力幅は、バイト数で指定します。省略すると、入力コードがUCS-2以外の場合は15バイト、UCS-2の場合は30バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が省略値を超えることは少ないので、省略するほうがふつうです。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

[2進・表示]

Windowsの2進形式整数・小数項目を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。

2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト／ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

出力幅は省略できます。出力幅を省略すると、ピクチャに従って出力されます。

出力幅、ピクチャ共に省略すると、

入力のバイト数	1	2	3	4	5	6	7	8
出力幅	3	5	8	10	13	15	17	18

の要領で2進キャストから自動的に計算された値にさらにコードによって、出力がUCS-2の場合、計算された値×2、UCS-2以外の場合は、計算された桁数そのままが使われます。例を示すと、入力2進キャスト=i4s、出力幅=省略という指定は、出力コードがUCS-2の場合、出力幅=20という指定と同じです。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[ゾーン・2進]

WindowsのCOBOLのゾーン形式数値項目を、ホストの2進形式整数・小数項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、入力ピクチャは **s4.1** と指定します。

2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト／ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

入力ピクチャも2進キャストにつづくピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、入力ピクチャ=省略、2進キャスト／ピクチャ=i4s4.1という指定は、入力ピクチャ=s4.1を指定したのと同じ意味になります。

[パック・2進]

Windows COBOLのパック、またはBCD形式数値項目を、ホストの2進形式整数・小数項目に変換します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
- 1 2 3 . 4 という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、
入力ピクチャは **s 4. 1** と指定します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
- 1 2 3 . 4 という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2
進キャスト/ピクチャは **i 4 s 4. 1** と指定します。なお、2進キャストは省略できません。

入力ピクチャも2進キャストにつづくピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、入力ピクチャ=省略、
2進キャスト/ピクチャ=**i 4 s 4. 1**という指定は、入力ピクチャ=**s 4. 1**を指定したのと同じ意味になります。

[2進・ゾーン]

Windowsの2進形式整数・小数項目を、ホストのCOBOLのゾーン形式数値項目に変換します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
- 1 2 3 . 4 という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2
進キャスト/ピクチャは **i 4 s 4. 1** と指定します。なお、2進キャストは省略できません。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
- 1 2 3 . 4 という数字を5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは **s 4. 1** と指定します。

2進キャストにつづくピクチャも出力ピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、2進キャスト/ピクチャ=**i 4 s 4. 1**、出力ピクチャ=省略という指定は、出力ピクチャ=**s 4. 1**を指定したのと同じ意味になります。

[2進・パック]

Windowsの2進形式整数・小数項目を、ホストのCOBOLのパック、またはBCD形式数値項目に変換します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を3バイトの符号つきパック形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。

2進キャストにつづくピクチャも出力ピクチャも省略できますが、両方同時には省略できません。一方を省略すると、指定した方の値で補って実行されます。たとえば、2進キャスト/ピクチャ=**i4s4.1**、出力ピクチャ=省略という指定は、出力ピクチャ=**s4.1**を指定したのと同じ意味になります。

[2進・2進]

Windowsの2進形式整数・小数項目を、ホストの2進形式整数・小数項目項目に変換します。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という数字を4バイトの符号つき2進形式の項目とする場合、2進キャスト/ピクチャは **i4s4.1** と指定します。

2進キャスト/ピクチャは入力または出力のどちらかを省略できます。入力を省略すると「出力と同じ」とみなされ、出力を省略すると「入力と同じ」とみなされます。なお、入力2進キャスト/ピクチャと出力2進キャスト/ピクチャを同時に省略することはできません。

[バイナリ反転]

バイナリ反転は、「コード変換を一切せずに、幅分のデータをバイト単位で左右反転する」という変換方法です。2進数値データ(整数、小数、実数)は、ホスト側が正順であるのに対して、Windows側が逆順であることが多いので、2進数値データの内容をそのままバイト単位で左右反転する場合等に使用します。2進・2進変換のような加工機能はありませんが、その分だけ処理が高速です。

幅は、10進のバイト数で指定します。たとえば4バイトの入力データが、16進表現で、**01 AB CD EF** であれば、出力データは **EF CD AB 01** になります。また、幅の代わりに16進列定数を指定することもできます。その場合、指定した16進列定数はWindows側(入力)レコードにあったかのようにみなされ、左右反転されます。

[表示へ]

入力した数値定数、またはシステム変数の値を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

入力値は省略できません。

出力幅は省略できます。出力幅を省略すると、「入力した値を正確に出力するために必要な幅」で出力されます。たとえば、入力値 = `[-123.4]`、出力幅 = 省略という指定は出力幅 = 1 (符号) + 3 (整数部桁数) + 1 (小数点) + 1 (小数部桁数) = 6 を指定したのと同じ意味になります。(出力コードが UCS-2 の場合は 12)

[ゾーンへ]

入力した数値定数、システム変数の値を、出力ピクチャに従ってホストの COBOL のゾーン形式数値項目に変換します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という定数を入力値として指定し、5 バイトの符号つきゾーン形式の項目に記録するとすれば、出力ピクチャは `s4.1` と指定します。

入力値、出力ピクチャともに省略できません。

[パックへ]

入力した数値定数、システム変数の値を、出力ピクチャに従ってホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という定数を入力値として指定し、3バイトの符号つきパック形式の項目に記録するとすれば、出力ピクチャは **s4.1** と指定します。

入力値、出力ピクチャともに省略できません

[2進へ]

入力した数値定数、システム変数の値を、出力ピクチャに従ってホストの2進形式整数・小数項目に変換します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数、汎用定数 (Zero、Min、Max)**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

出力2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、**-123.4** という定数を入力値として指定し、4バイトの符号つき2進形式の項目に記録するとすれば、出力2進キャスト／ピクチャは **i4s4.1** と指定します。

入力値、出力2進キャスト／ピクチャともに省略できません。

[Alphaへ]

入力した数値定数、またはシステム変数の値を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

入力値として、つぎの数値定数、システム変数が使えます。

数値定数 **整数定数、小数定数、16進定数**
システム変数 **SysPhase、SysRecNum、SysBreak、SysReturn、SysQuit**

入力値は省略できません。

出力幅は省略できます。出力幅を省略すると、「入力した値を正確に出力するために必要な幅」で出力されます。たとえば、入力値 = [-123.4]、出力幅 = 省略という指定は出力幅 = 1 (符号) + 3 (整数部桁数) + 1 (小数点) + 1 (小数部桁数) = 6 を指定したのと同じ意味になります。

[Ank化]

Windows側(入力)文字列をできる限りANK(半角)に変換します。ホストが汎用機・オフコンの場合、どのコード変換が行われるかは、変換設定のANKコードの設定で決まります(ふつう、セットアップ時に一回だけ行います)。

入力幅は、10進のバイト数で指定します。式による指定もでき、そのなかでは、

\$ Windows側(入力)のレコード長
*** Windows側(入力)の残りバイト数** という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると「入力幅と同じ」とみなします。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、ANK項目のおわりのほうが切り捨てられます。逆に、項目長を伸ばすと、ホスト側のANK項目のおわりにスペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

\$ ホスト側(出力)のレコード長
*** ホスト側(出力)の残りバイト数** という特殊変数が使えます。

[漢字化]

Windows側（入力）文字列をできる限り漢字（全角）に変換します。どのコード変換が行われるかは、変換設定の漢字変換方式の設定で決まります（ふつう、セットアップ時に一回だけ行います）。

入力幅は、10進のバイト数で指定します（漢字の文字数ではありません）。式による指定もでき、そのなかでは、

- \$ Windows側（入力）のレコード長
- * Windows側（入力）の残りバイト数 という特殊変数が使えます。

入力幅を省略すると*を指定したとみなします。

出力幅も省略できます。出力幅を省略すると入力側のコード、および出力側のコードから適切な値を計算して求めます。

項目長を変更するときは、出力幅を10進のバイト数で指定します。

項目長を縮めると、漢字項目のおわりのほうが切り捨てられます（漢字の中央で切れることはありません）。逆に、項目長を伸ばすと、ホスト側の漢字項目のおわりに漢字変換方式に設定されている漢字スペースが詰められます。

出力幅は式による指定もでき、そのなかでは、

- \$ ホスト側（出力）のレコード長
- * ホスト側（出力）の残りバイト数 という特殊変数が使えます。

[Alpha・ゾーン]

Windowsの文字形式数値項目を、ホストのCOBOLのゾーン形式数値項目に変換します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなすので、ふつうは明示的にバイト数を指定します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字が5バイトの符号つきゾーン形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

[Alpha・パック]

Windowsの文字形式数値項目を、ホストのCOBOLのパック形式数値項目、BCD形式数値項目に変換します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなすので、ふつうは明示的にバイト数を指定します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字を3バイトの符号つきパック形式の項目に記録するとすれば、ピクチャは **s4.1** と指定します。**123.4** という数字を3バイトのBCD形式の項目に記録するとすれば、ピクチャは **b5.1** と指定します。なお、ピクチャは省略できません。

[Alpha・2進]

Windowsの文字形式数値項目を、ホストの2進形式整数・小数項目に変換します。入力側は1バイト単位で取り出されます。

入力幅は、バイト数で指定します。省略すると15バイトとみなされるので、プリント形式からの変換の場合は、ふつうは明示的に桁数を指定します。デリミタ形式からの変換の場合は、数値項目が15バイトを超えることは少ないので、省略するほうがふつうです。

2進キャスト/ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字を4バイトの符号つき2進形式の項目に記録するとすれば、2進キャスト/ピクチャは **i4s4.1** と指定します。なお、2進キャストは省略できません。

[ゾーン・Alpha]

Windows COBOLのゾーン形式数値項目を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字が5バイトの符号つきゾーン形式の項目に記録されているとすれば、
 ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値が使われます。たとえば、ピクチャ = s4.1、
 出力幅 = 省略という指定は、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[パック・Alpha]

Windows COBOLのパック、またはBCD形式数値項目を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、
-123.4 という数字が3バイトの符号つきパック形式の項目に記録されているとすれば、
 ピクチャは **s4.1** と指定します。なお、ピクチャは省略できません。

パック形式では、整数部桁数+小数部桁数を奇数にしておくのが通例です。整数部の最上位桁に意味があるのかないのかは、半々の割合です。出力幅は省略できます。出力幅を省略すると、

符号つきなら1、符号なしなら0とする
+整数部桁数
+1+小数部桁数（小数部があれば）

の要領でピクチャから自動的に計算された値が使われます。例を示すと、ピクチャ = s4.1、
 出力幅 = 省略という指定は、出力幅に7を指定したのと同じ意味になります。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

[2進・Alpha]

Windowsの2進形式整数・小数項目を、ホストの文字形式数値項目に変換します。変換結果は右詰めになります。出力側のコード体系に関係なく、1バイトデータに変換されて出力します。

2進キャスト／ピクチャの指定方法については、「操作の基礎」の章で説明しました。たとえば、`-123.4` という数字が4バイトの符号つき2進形式の項目に記録されているとすれば、2進キャスト／ピクチャは `i4s4.1` と指定します。なお、2進キャストは省略できません。

出力幅は省略できます。出力幅を省略すると、ピクチャに従って出力されます。

出力幅、ピクチャ共に省略すると、

入力のバイト数	1	2	3	4	5	6	7	8
出力幅	3	5	8	10	13	15	17	18

の要領で2進キャストから自動的に計算された値が使われます。例を示すと、入力2進キャスト = `i4s`、出力幅 = 省略という指定は、出力幅 = 10という指定と同じです。

出力幅を明示的に指定するときは、オーバーフローに注意しながら10進のバイト数で指定します。オーバーフローすると、符号や上位桁が切り捨てられるので、注意してください。

(5) 引用符くくりの指定

ホストがUnix、Windowsの場合、デリミタ形式に変換するときは、変換後の文字列を引用符でくくることができます。ただし、プリント形式への変換のときは、この引用符くくりの機能は無効になります。

引用符にはつぎの3種類があります。

両くくり	一つの項目を単独でくくる
左くくり	複数の項目を一つにくくるときに、引用符の始まりを表す
右くくり	複数の項目を一つにくくるときに、引用符の終わりを表す

市販ソフトの入力用には、文字項目だけを引用符でくくることが多いのですが、引用符を使わないソフトもあれば、すべての項目を引用符でくくるソフトもあります。

◆参考...

たとえば、デリミタ形式のうちK3フォーマットと呼ばれるものは、

各項目はコンマで区切る	
数値はそのまま（位取りのコンマ不可）	
文字列は引用符でくくる	というルールになっています。

(6) デリミタ挿入の有無の指定

ホストがUnix、Windowsの場合、デリミタ形式に変換するときは、デリミタ挿入を無効にすることができます。デリミタ挿入を無効にすると、その次の項目と合わせて1項目とみなします。ただしプリント形式への変換のときは、このデリミタ有無の指定の機能は無効になります。

たとえばつぎのように、ある項目（Ank 20）の前に定数（Data=）を挿入するときなどに、デリミタ挿入を無効にします。

```

      :
Ank   ['Data='] &
Ank   20
      :
```

(7) 漢字イン／アウトの指定

ホストが汎用機・オフコンの場合、項目の前後に、漢字イン (K I)・漢字アウト (K O) を挿入することができます。

これは、漢字項目に漢字イン (K I)・漢字アウト (K O) を指定するのが、ふつうの使い方です。この指定と A n k・漢字まじり項目変換は、似ているところもありますが別物です。ご注意ください。

1 : 漢字イン、3 : 漢字イン+アウトを指定した場合、さらに10進数数値 (0~255の範囲) を入力することもできます。東芝方式の A N K・漢字項目化するために「長さバイト」を付加する場合に指定します。

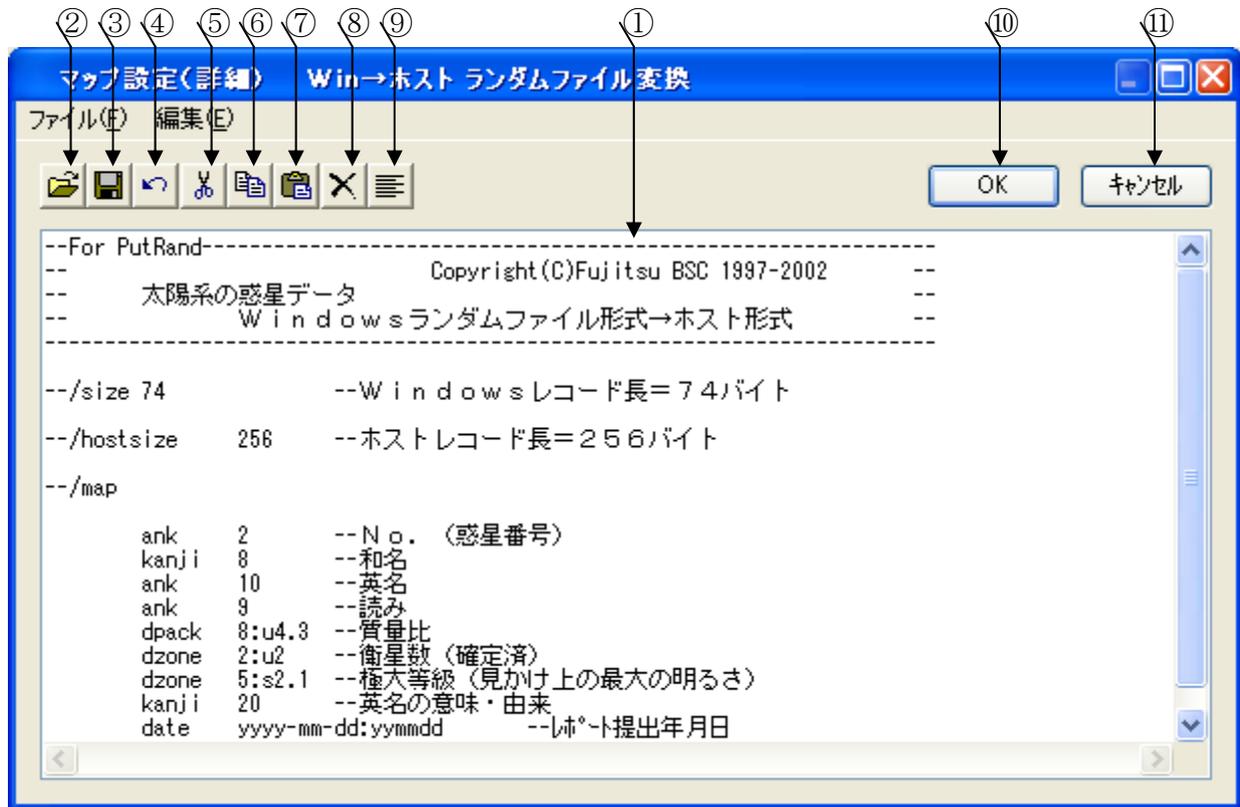
(8) 改行コード挿入の指定

ホストが U n i x、W i n d o w s の場合、任意のところに改行コードを挿入する (改行コードで項目を区切る) こともできます。改行コード挿入には、つぎの2種類があります。

通常 出力がプリント形式データファイルの時は、指定が無効になる
強制 出力がプリント形式データファイルの時も、指定が有効になる

出力がデータファイルのときに、自動的にレコード末尾に改行コードが付加される機能とは別のものですから、混同しないでください。

<マップ設定（詳細）ウィンドウ>



①マップ設定の記述をするエディットボックスです。

②～⑨はファイル (F)、編集 (E) のメニューの機能がボタン化されています。

②既存のパラメータファイル等を読みます。

③①の内容をパラメータファイルへ保存します。

④直前の編集作業が無効になり、元に戻ります。

⑤選択した文字列が切り取られ、カットバッファに入ります。

⑥選択した文字列がカットバッファに入ります。

⑦⑤／⑥の操作でカットバッファに入った内容を、カーソルがある位置へ貼りつけます。

カーソルがある位置以降に文字列があれば、挿入になります。

⑧選択した文字列が削除されます。

⑨①の文字列をすべて選択状態にします。

⑩設定が完了したら、OKボタンをクリックします。

⑪キャンセルボタンをクリックすると、すべての入力／編集作業が無効になります。

◆注意 --- マップ設定（詳細）の記述内容について

マップ設定（詳細）のエディットボックスの中に記述する内容は、レコードレイアウト（/map を含まない /map 以降）のみの記述となります。既存のパラメータファイルを読み込むと、それ以外の記述（/delimited～、/map 等）は自動的にコメントになります。記述に誤りがある状態でOKボタンをクリックすると、エラーとなる場合があります。修正してから再度OKボタンをクリックしてください。/map 以降の文法については、「コマンド編」の「コマンド型の実行」の章を参照してください。

例) マップ設定（詳細）にパラメータファイル“PNPUTRND.P”を読み込む

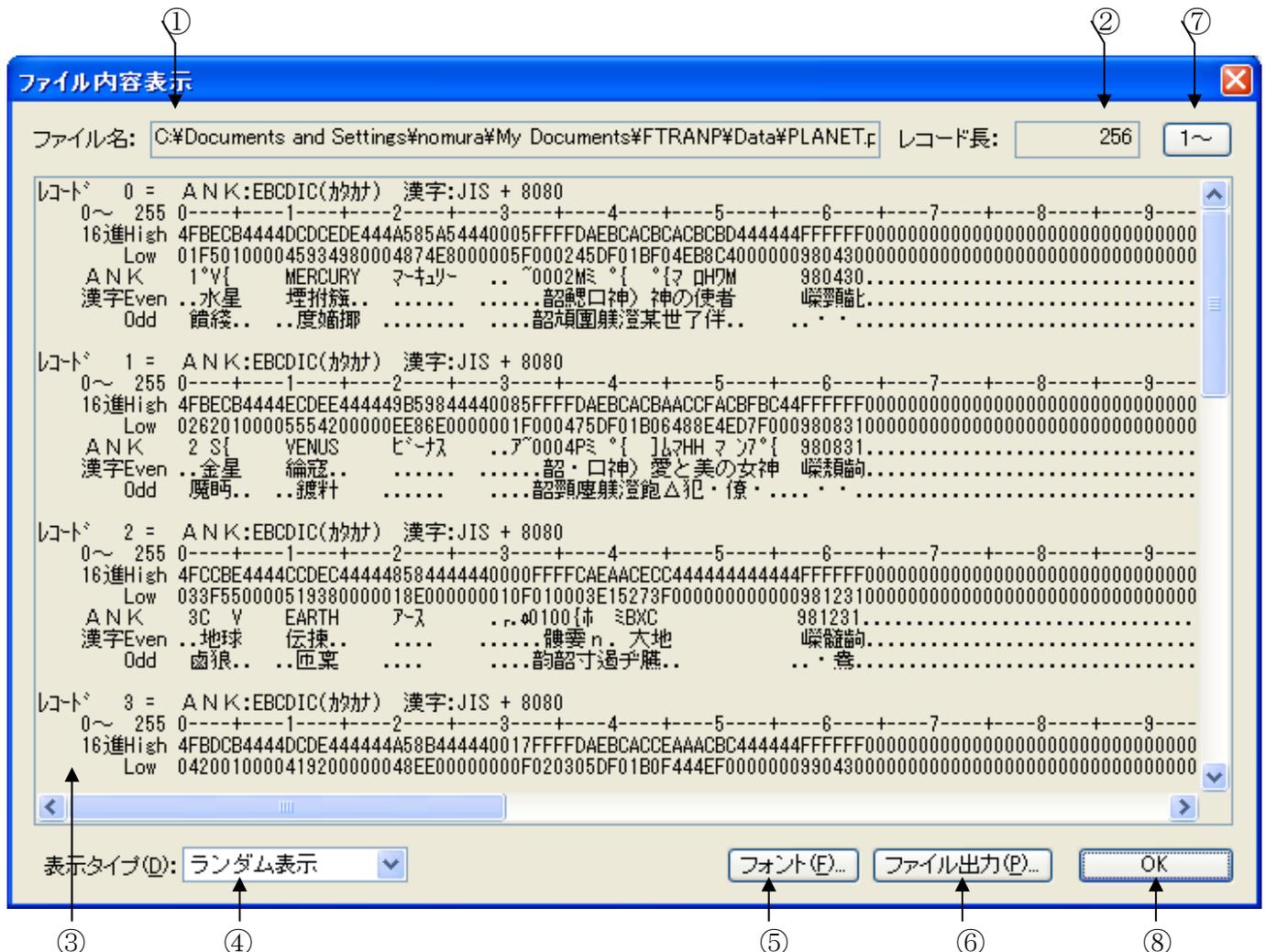
--/size 74		--Windowsレコード長=74バイト	} コメントになる行
--/hostsize 256		--ホストレコード長=256バイト	
--/map			
ank 2		--No. (惑星番号)	} 変換時に、 使われる行
kanji 8		--和名	
ank 10		--英名	
ank 9		--読み	
dpack 8:u4.3		--質量比	
dzone 2:u2		--衛星数 (確定済)	
dzone 5:s2.1		--極大等級 (見かけ上の最大の明るさ)	
kanji 20		--英名の意味・由来	
date yyyy-mm-dd:yymmdd		--レポート提出年月日	

■Win→ホストランダムファイル変換の実行

変換ボタンをクリックすると、Win→ホストランダムファイル変換が始まります。

●ファイル内容表示ウインドウ

変換結果表示 (V) のチェックボックスをONにし、Win→ホストファイル変換を実行すると、実行ウインドウを閉じた後に下記のファイル内容表示ウインドウが開き、変換先のファイルの中身を確認することができます。



- ①変換先のファイル名が表示されます。
- ②変換先のファイルのレコード長が表示されます。
- ③変換先のファイルの中身が、③で選択されている表示タイプで出力されます。
- ④③の表示タイプをランダム表示、テキスト表示、テキストのみ表示に切り替えます。
- ⑤フォント (F) ボタンをクリックして、③の出力文字フォントを切り替えることができます。
- ⑥ファイル出力 (P) ボタンをクリックして、③の内容をファイルに保存できます。
- ⑦桁位置のガイド表示を、0 起点 (先頭を 0 として数える : デフォルト)、1 起点 (先頭を 1 として数える) のどちらかに切り替えることができます。
- ⑧OK ボタンをクリックすると、ファイル内容表示ウインドウを閉じます。

ファイル内容表示は変換結果のプレビュー表示機能のため、表示できる容量に制限があります。全ての内容が表示されている場合はデータの最後に、

----- ファイルの最後まで表示しました. -----

全ての内容が表示できなかった場合には、

----- ファイルの途中で表示を切りました. -----

と表示されます。

■注意事項

漢字があるときの注意

漢字項目変換やA n k・漢字まじり項目変換を行うときは、あらかじめ

変換設定の漢字変換方式設定で、適当な漢字変換方式を割り当てておく

のを忘れないでください。また、入力幅、出力幅は漢字データについても

バイト単位で指定

します。漢字の文字数ではないことに注意してください。

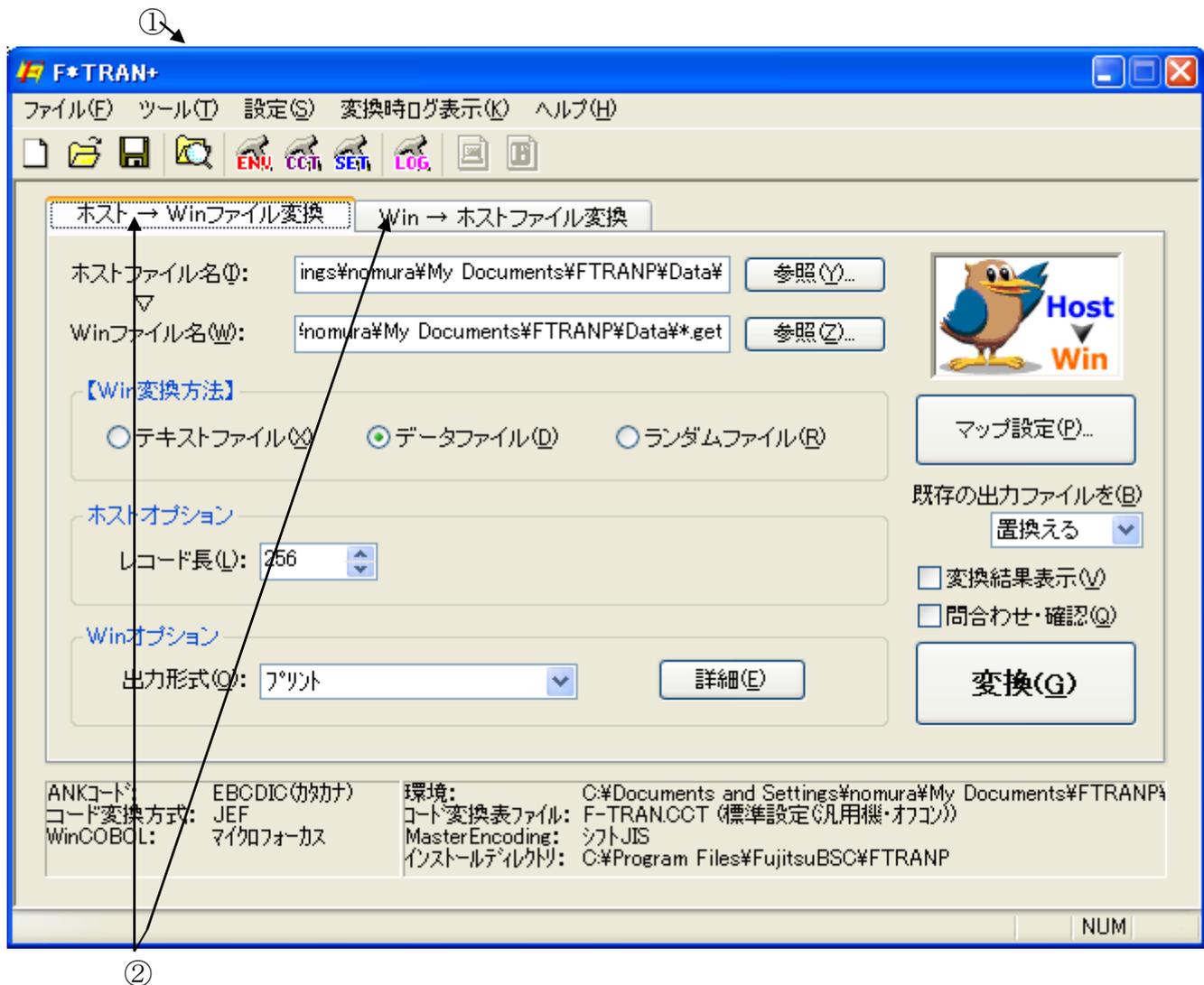
その他の注意事項

「Win→ホストファイル変換」の節を参照してください。

4.10 スクリプトファイル

ホスト→Winファイル変換、Win→ホストファイル変換の2つのメイン処理部で設定した項目は、F*TRAN+のスク립トファイルとして保存することができます。

作成されたスク립トファイルは、エクスプローラからクリックするだけで実行できます。スク립トファイルの実行については、解説で説明します。

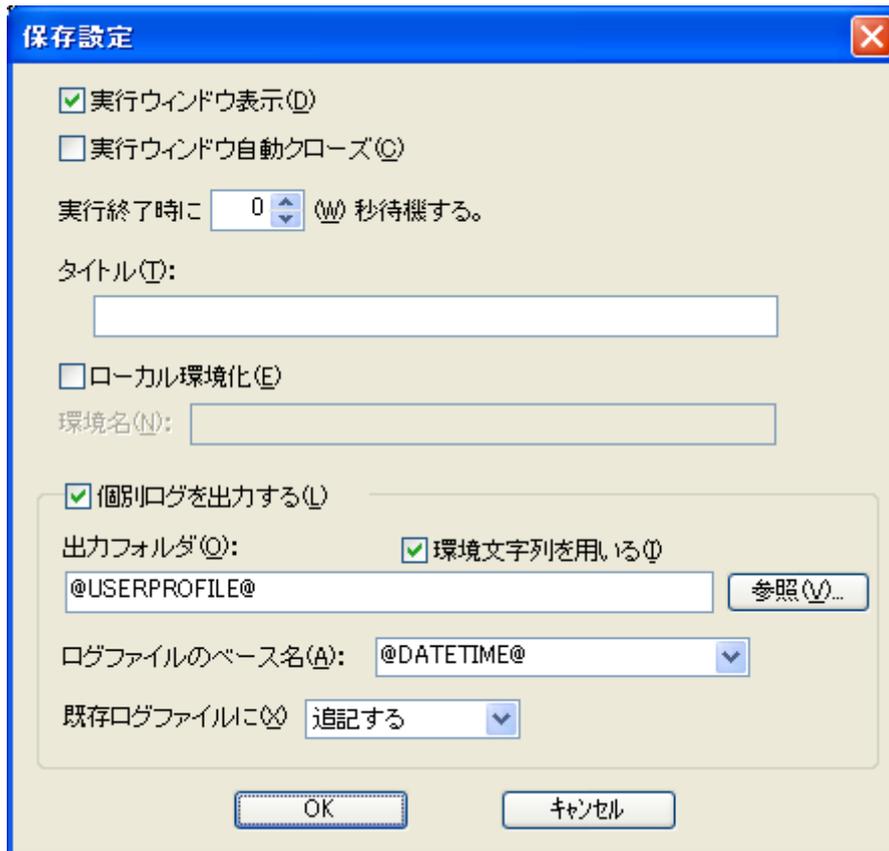


- ①読み込まれたスク립トファイル名。
- ②選択されているメイン処理部が、保存の対象となる。

[新規にスクリプトファイルに保存する場合]

②のタブコントロールをクリックし、処理（ホスト→Winファイル変換、Win→ホストファイル変換のいずれか）を選択します。必要な項目を設定し、実行が可能なら、実際に処理を行い結果を確認します。

この状態の設定を保存するなら、メニューバーのファイル（F）→名前を付けて保存（A）を実行します。ここで、つぎのウインドウが開きます。



これは、スクリプトファイルを実行する時の設定をするウインドウです。スクリプトファイルを使ってバッチなどから実行する必要がない場合は、どの設定でもかまいません。

- ①**実行ウインドウ表示をON、実行ウインドウ自動クローズをOFFで、**
実行ウインドウを表示し、処理後、実行ウインドウ開いたまま待つ
- ②**実行ウインドウ表示をON、実行ウインドウ自動クローズをONで、**
実行ウインドウを表示し、処理後、自動的に実行ウインドウ閉じて終了する

③実行ウインドウ表示をOFFで、

実行ウインドウを表示しないで処理を実行し、終了する

の3通りの設定ができます。

実行終了時の待ち時間を秒数で設定することもできます。通常は0秒のままで設定しますが、データ変換の対象となるファイルのサイズが小さすぎてアプリケーションからスクリプトファイルを実行した後の戻り値が取得できない場合は、数秒の値を設定します。

実行ウインドウ等のタイトルを変更する場合は、タイトルのエディットボックスに出力する文字列を入力します。

このウインドウで現在の環境をローカル環境化できます。作成する環境名を指定して下さい。現在の環境が指定された環境名フォルダにコピーされます。ローカル環境を作成してスクリプトファイルを出力した場合、そのスクリプトファイルでは作成したローカル環境を使用して変換が行われます。また、環境設定も保存した環境が選択されている状態になります。

●個別ログの出力設定

変換結果や変換漏れ情報などのログファイルは、ログ設定で指定されたログ出力フォルダに1日単位で出力されます。しかし、変換の度にその結果を個別に出力したい場合があります。

その場合は個別ログの出力設定をしてください。保存したスクリプトファイルの実行結果を個別ログファイルに出力することが可能となります。なお、個別ログを出力した場合、通常のログは出力されません。出力されるログファイルの形式は通常のログファイル（基本ログファイル、詳細ログファイル）と同一です。

項目	説明
個別ログを出力する	個別ログを出力するかの指定をします。この設定をオフにするとログ設定の指定に従い標準のログ出力が有効になります。
出力フォルダ	個別ログの出力フォルダのパスを指定します。 「参照」ボタンでフォルダを選択することができます。選択したフォルダが環境文字列に合致した場合は自動的に置き換わります。初期値は「@USERPROFILE@」となります。
環境文字列を用いる	出力フォルダの指定に環境文字列を使う場合、チェックをします。チェックされている場合、チェックを外すことでフルパス表記に変更することができます。
ログファイルのベース名	個別ログのベース名を指定する。ドロップダウンから@DATE@、@DATETIME@および@DATETIME-MS@を選択することが可能で、初期値は「@DATETIME@」となります。
既存ログファイルに	既存ログファイルが存在する場合、「追記する」か「上書きする」を選択します。初期値は「追記する」となります。

●出力フォルダ

個別ログファイルを出力する既存のフォルダを指定します。フォルダの指定はフルパスか、環境文字列を使った入力が可能です。フォルダの指定に環境文字列を利用すると実行環境に依存しない指定が可能です。フルパスの指定をすることで他の環境で実行すると正常に動作しなくなる場合がありますが、それを防ぐことができます。

環境文字列には以下のような文字列が利用可能です。必ずこれらの文字列から始まる記述にし、フォルダ名の途中に指定することはできません。

環境文字列	フォルダ
@USERPROFILE@	カレントユーザのプロファイルフォルダ 例：C:\Users\ユーザー名
@ALLUSERSPROFILE@	All Users のプロファイルフォルダ 例：C:\ProgramData
@FT_Personal@	カレントユーザのドキュメント（以下と同じ） 例：@USERPROFILE%\Documents
@FT_Userdata@	F*TRAN+のユーザデータフォルダ
@SystemDrive@	システムドライブレター
@環境変数名@	環境変数名を指定することが可能。OS の環境変数が利用可能です。

◆例

@USERPROFILE%\FT_LOG

カレントユーザのプロファイルフォルダ配下の FT_LOG フォルダを指定

「環境文字列を用いる」のチェックボックスは入力されたフルパスを環境文字列を使ったパスに変更します。また、チェックされている場合、チェックを外すことでフルパス表記に変更することができます。

◆参考 ---- <ユーザデータフォルダ>とは

<ユーザデータフォルダ>は、ログオンユーザの「ドキュメント」にある“FTRANP”フォルダがデフォルトで指定されています。

● ログファイルのベース名

出力されるログファイル名は以下のような構成になります。(BaseName がベース名)

基本ログ : BaseName_Basic.log

詳細ログ : BaseName_Detail.log (詳細ログは必要な場合のみ作成される。)

なお、ベース名には半角の “_” (アンダースコア) 及び、 “.” (ピリオド) は使えません。また、指定できるベース名の長さは最大 80 文字です。

ベース名には以下の置き換え文字列 (すべて半角大文字) を含めることが可能です。

置き換え文字列	説明	例
@DATE@	日付 YYYYMMDD 形式の 8 文字	20071015
@DATETIME@	日時 YYYYMMDD-HHmmSS 形式の 15 文字	20071015-130123
@DATETIME-MS@	日時 YYYYMMDD-HHmmSS-msc 形式の 19 文字 (msc はミリ秒 0~999 の十進数)	20071015-111523-807

◆ 例 ベース名を “変換ログ” とした場合

基本ログ : 変換ログ_Basic.log

詳細ログ : 変換ログ_Detail.log

ベース名を “Daily-@DATE@” とし、2008 年 1 月 29 日に実行した場合

基本ログ : Daily-20080129_Basic.log

詳細ログ : Daily-20080129_Detail.log

ログファイルは標準のログファイルの大きさと同様に 1 ファイルの最大サイズは 1 MB となり、そのサイズを超えるか何らかの原因で書込みができなかった場合、以下のように連番が追加されたファイルが作成されます。(ベース名を ftran とした場合の基本ログ)

ftran_Basic.log

ftran_Basic_000000.log …… 追加分

ftran_Basic_000001.log …… 追加分

なお、環境文字列で展開される文字列の長さはパス名を含めて 216 文字までで、それを超えるとエラーになりログ設定の指定に従って出力されます。

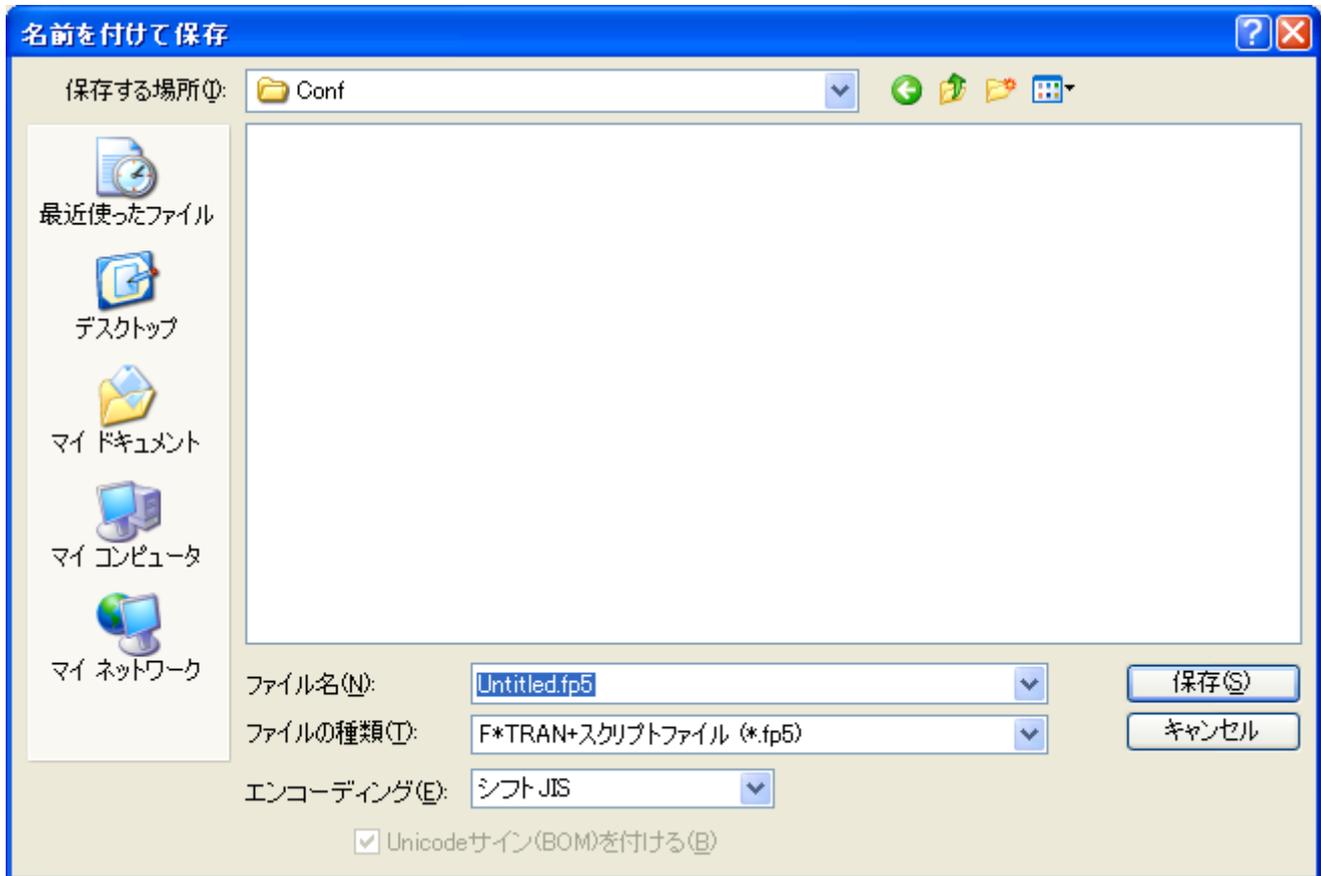
● ログ設定について

個別ログ出力時もログ設定で指定した「ログファイルの書込み時の待ち時間」と「変換漏れ情報を詳細ログに出力する」が有効になりますが、個別ログは自動削除を行いません。

個別ログの出力フォルダにログ設定でのログ出力フォルダを指定する事は推奨しません。将来ログ機能が拡張された時、動作の保証ができないためです。

設定が完了したら、OKボタンをクリックします。

つぎに、“名前を付けて保存”のウィンドウが開いたら、ファイル名を入力し、保存（S）ボタンをクリックして、スクリプトファイルへの保存がなされます。



スクリプトファイルのエンコーディングを以下の4通りから選択します。

シフトJIS

Unicode (UCS-2 LE)

Unicode (UCS-2 BE)

Unicode (UTF-8)

Unicodeの場合は以下の選択も可能です。

Unicodeサイン (BOM) を付ける

省略値は環境設定の `User Encoding` に従います。（詳細は「4.12 環境設定」を参照）

[保存したスクリプトファイルを読み込む場合]

メニューバーのファイル (F) →開く (O) を実行します。“ファイルを開く”のウインドウが開いたら、ファイル名を選択し、開く (O) ボタンをクリックします。

スクリプトファイルが読み込まれると、②のタブコントロールが切り替わり、メイン処理部の内容が保存した時の状態になります。①のタイトルバーにスクリプトファイル名が表示されます。

[変更した設定をスクリプトファイルに書き込む場合]

スクリプトファイルを読み込みます。変更する設定があれば、修正します。

メニューバーのファイル (F) →上書き保存 (S) を実行します。

“名前を付けて保存”のところで説明した保存設定のウインドウが開きます。必要があれば変更してOKボタンをクリックすると、スクリプトファイルへの上書き保存がなされます。

[今までの設定を無効にし、新規に設定をはじめめる場合]

メニューバーのファイル (F) →新規作成 (N) を実行します。実行すると、“新規に作成しますか？”と確認ウインドウが開きます。はい (Y) ボタンをクリックすれば、メイン処理部の設定が起動時の状態になります。いいえ (N) ボタンをクリックすれば、新規作成を中止できます。

以上が、ファイル・メニューの基本的な操作です。

◆注意 ---- スクリプトファイルの互換性

スクリプトファイルは、作成したF*TRANより前のバージョンのF*TRANでは使用することができません。これは、バージョンアップにより追加された機能がそれより前のバージョンのF*TRANでは使用できないことによります。

ただし、以前のバージョンで作成したスクリプトファイルは、インストールディレクトリの修正や使用されている環境の移行などの作業が必要になる場合もありますが、それよりも新しいバージョンのF*TRANで使用することができます。

◆注意 ---- インストールモードの違いによるスクリプトファイルの互換性

F*TRAN+V 5.0およびV 7.0まで利用可能であった5.0互換インストールモードで作成したスクリプトファイルとの互換性はありません。

なお、V 8.0以降は5.0互換インストールモードは使用できません。

詳細は「操作説明書 導入編」を参照してください。

■解説 — スクリプトファイルの実行

F*TRAN+のファイルメニューの機能を使って作成(保存)されたファイル(拡張子が“.FP5”のファイル)は、つぎの方法で実行することができます。

- ①コマンドプロンプトから下記の書式を入力し、実行する。
(または、バッチファイルの中に記述し、実行する)

```
[START /W[AIT]] FP FP5 ファイル名.FP5 [-H ホストファイル名] [-W Winファイル名]
```

通常は、

```
C : ¥> f p s h o r i 1 . f p 5 ↓
```

のように入力し、実行します。

バッチファイルの中で組み合わせて使用する場合は、START/Wを頭に付けます。

- ②エクスプローラ等からFP5ファイルをダブルクリックし、実行する。

F*TRAN+を正常にインストールしてある環境では、エクスプローラ等からFP5ファイルをダブルクリックするだけで、実行することができます。

◆注意 ---- FP5ファイルが実行できる範囲

FP5ファイルが実行できる範囲はつぎの通りです。

ホスト→Winファイル変換
Win→ホストファイル変換

のいずれか1処理です(組み合わせ等はできません)。

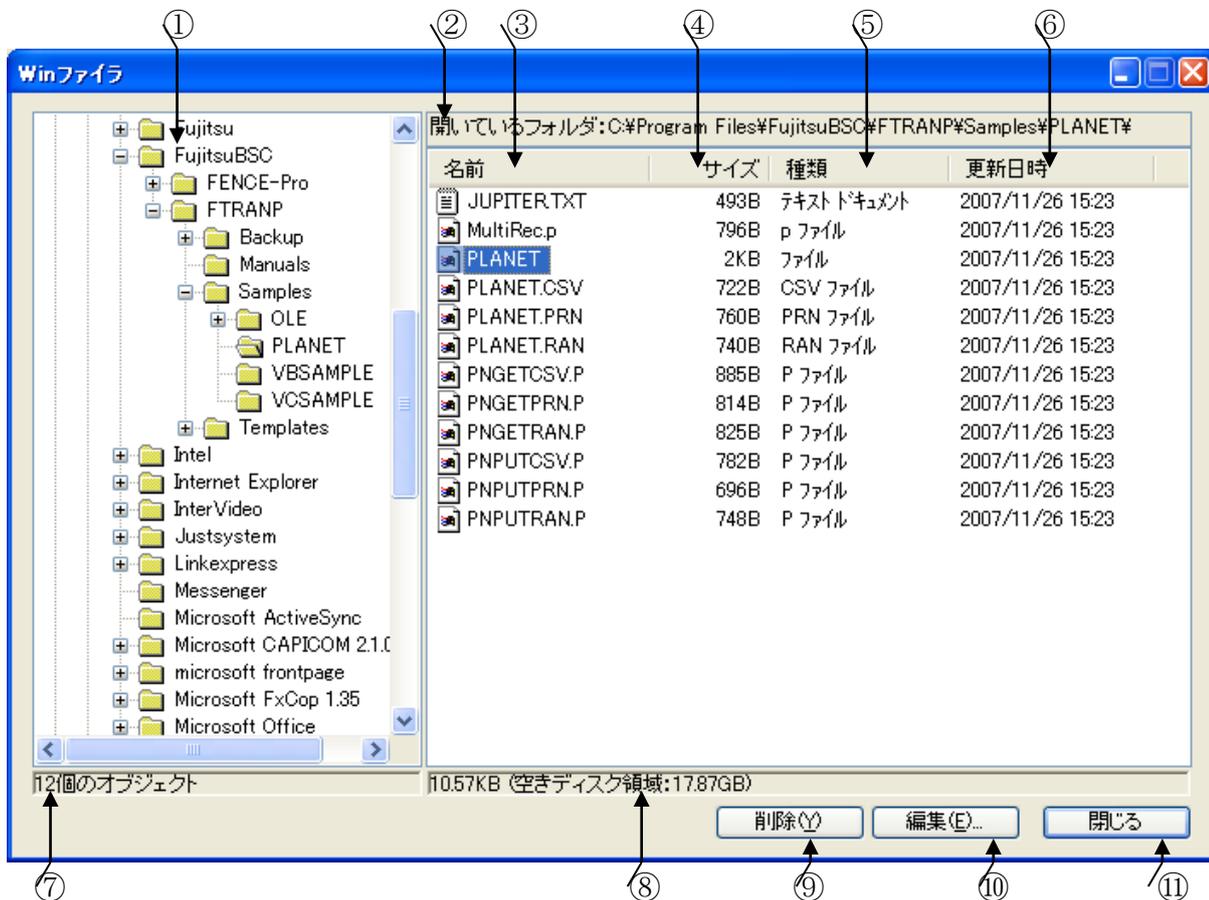
また、下記の条件のFP5ファイルを実行しようとする、正常な実行ができません。F*TRAN+のメインウィンドウが起動し、FP5ファイルを開いた状態になります。不足している項目を確定させてから実行してください。

- ・ホストファイル名が指定されていない場合
- ・Winファイル名が指定されていない場合 (Win→ホストファイル変換のみ)

-Hや-Wを使ってファイル名を指定する場合は、FP5ファイル作成時にファイル名の欄を省略してください。FP5ファイルにファイル名が指定されている場合は、そちらが優先されず。

4.11 Winファイラ

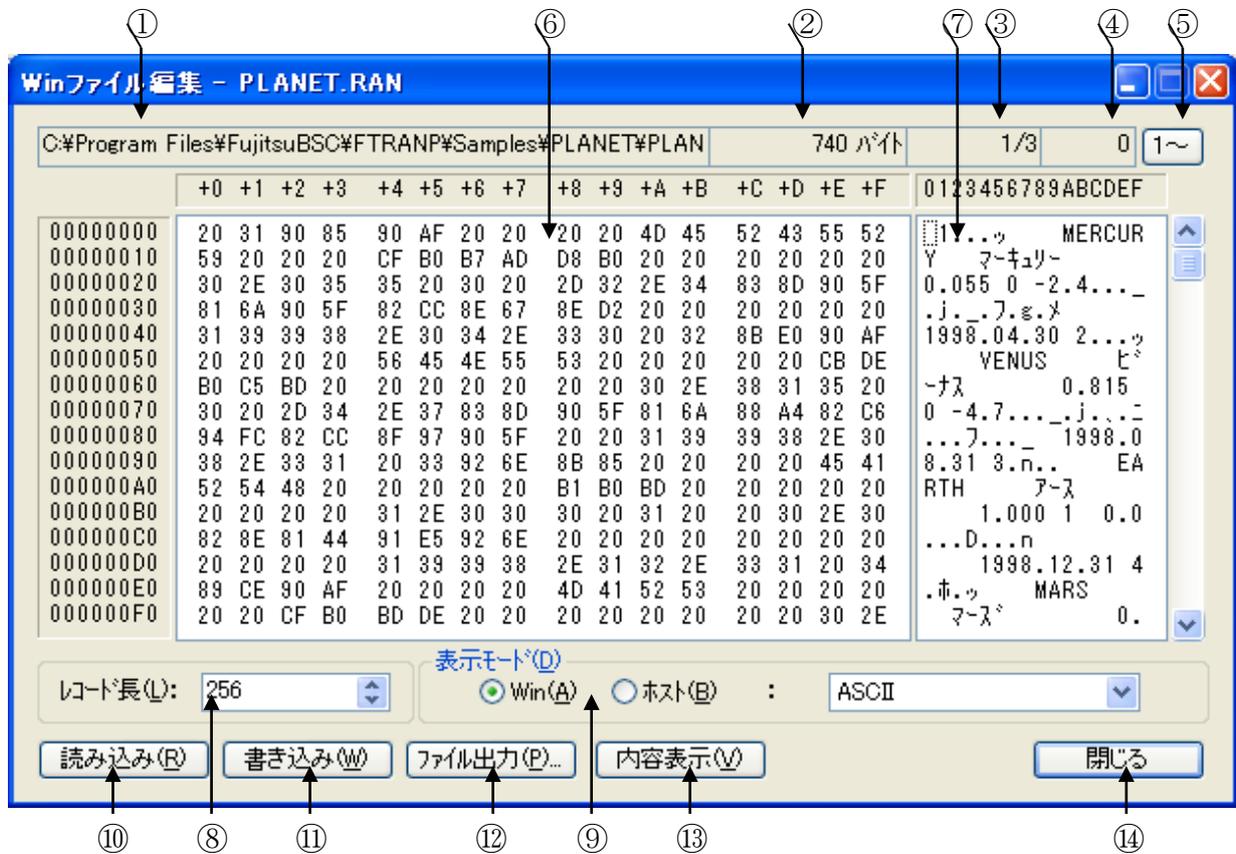
Winファイラは、Windowsのファイルを管理する機能を持っています。メニューバーのツール (T) → Winファイラ (W) を実行すると、つぎのウインドウが開きます。



- ①全ドライブのフォルダ情報が表示されるエリアです。
- ②開いているフォルダ名が表示されます。
- ③開いているフォルダ内のフォルダ名とファイル名が表示されます。
- ④開いているフォルダ内のフォルダとファイルのサイズが表示されます。
- ⑤開いているフォルダ内のフォルダとファイルの種類が表示されます。
- ⑥開いているフォルダ内のフォルダとファイルの更新日時が表示されます。
- ⑦開いているフォルダ内のオブジェクトの個数が表示されます。
- ⑧開いているフォルダ内のオブジェクトの総容量と空きディスク容量が表示されます。
- ⑨③のフォルダまたはファイルを選択し、削除 (Y) ボタンをクリックすると、
選択したフォルダまたはファイルを削除することができます。
- ⑩③のファイルを選択し、編集 (E) ボタンをクリックすると、
Winファイルエディタが開きます。
- ⑪閉じるボタンをクリックすると、Winファイラを終了し、ウインドウを閉じます。

●Winファイル編集 (Winファイルエディタ)

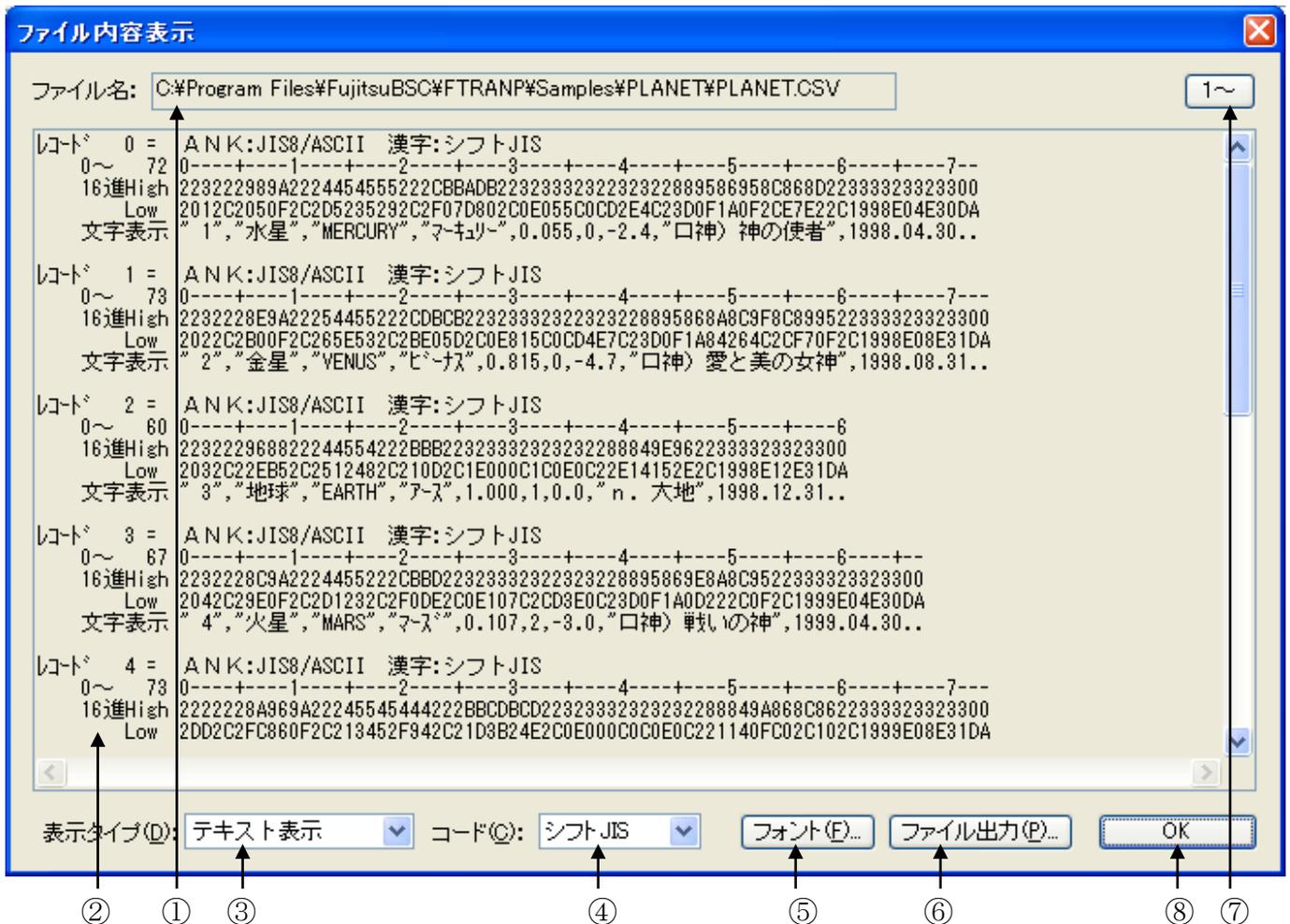
Windowsファイル (ホストファイルを含む) の中身を16進で表示し、編集できます。



- ①ファイル名が表示されます。
- ②ファイルサイズが表示されます。
- ③データのレコード番号/全体のレコード件数が表示されます。
- ④レコード内の桁位置 (先頭は0または1) が表示されます。
- ⑤桁位置を数える起点を、0起点、1起点のどちらかに切り替えることができます。
- ⑥ファイルの中身を16進表現で表示するエリアです。0～Fを入力できます。
- ⑦ファイルの中身を文字表現で表示するエリアです。Unicodeは表示できません。文字を入力できます。
- ⑧レコード長を1～32767で指定できます。デフォルトは256です。
- ⑨文字表示エリアの表示モードを切り替えます。
- ⑩読み込み (R) ボタンをクリックすると、データをディスクから読み直します。
- ⑪書き込み (W) ボタンをクリックすると、データをディスクに書き戻します。
- ⑫ファイル出力 (P) ボタンをクリックすると、表示されている画面からのデータ内容を、ファイルに出力するウィンドウが開きます。出力するレコード数とファイル名を指定して、保存 (S) ボタンをクリックすると、ファイルに出力されます。
- ⑬内容表示 (V) ボタンをクリックすると、ファイル内容表示ウィンドウが開きます。
- ⑭閉じるボタンをクリックすると、Winファイルエディタを終了します。

●ファイル内容表示ウインドウ

Winファイルエディタの表示モード (D) がWin (A) の状態で、内容表示 (V) ボタンをクリックすると、つぎのファイル内容表示ウインドウが開きます。



- ①Windows ファイル名が表示されます。
- ②Windows ファイルの中身が、③で選択されている表示タイプで出力されます。
デフォルトは、テキスト表示です。
- ③②の表示タイプを3種類の中から切り替えることができます。
- ④②の表示コードをシフトJIS、UCS-2LE、UCS-2BE、UTF-8から選択します。
- ⑤フォント (F) ボタンをクリックして、②の出力文字フォントを切り替えることができます。
- ⑥ファイル出力 (P) ボタンをクリックして、②の内容をファイルに保存できます。
- ⑦桁位置のガイド表示を、0 起点 (先頭を0として数える)、
1 起点 (先頭を1として数える) のどちらかに切り替えることができます。
- ⑧OKボタンをクリックすると、ファイル内容表示ウインドウを閉じます。

ファイル内容表示

ファイル名: C:\Program Files\FujitsuBSC\FTRANP\Samples\PLANET\PLANET.CSV レコード長: 256 1~

```

レコード 0 = ANK:JIS8/ASCII 漢字:シフトJIS
0~ 255 0-----1-----2-----3-----4-----5-----6-----7-----8-----9-----
16進High 223222989A2224454555222CBBADB22323332322322889586958C868D223333233233002232228E9A22254455222C
Low 2012C2050F2C2D5235292C2F07D802C0E055C0C2E4C23D0F1A0F2CE7E22C1998E04E30DA2022C2B00F2C265E532C2B
ANK "1","水","MERCURY","マキリ","0.055,0,-2.4","M_jiVió7i",1998.04.30.."2","い","VENUS","ヒナス",0
漢字Even "1","水星" ECR "ユ、, .5, , 2.4" 告 j フ ョ ヌ , 9 8 0 . 0 " 煩 , V N S , ヒ
Odd "","ッ","MRUY","マキリ" 0 0 5 0 - . , 口神) 神の使者" 1 9 . 4 3 " 2 , 金星" EU " "
レコード 1 = ANK:JIS8/ASCII 漢字:シフトJIS
0~ 255 0-----1-----2-----3-----4-----5-----6-----7-----8-----9-----
16進High 8A8C95223333233233002222228A969A2224454544422BCCBDC2232333232288849A868C86223333233233002
Low 222C0F2C1998E04E30DA2DD2C2FC860F2C213452F942C21D3B24E2C0E000C0C0E0C221140FC02C102C1999E08E31DA2
ANK "1","1999.04.30.."-"","い","ASTEROID","アロイト","0.000,0,0.0","mDió7i",1999.08.31.."5","の
漢字Even いの神" 1 9 . 4 3 " - , 小惑星" S E O D , ア テ イ , . 0 , . . , a . 星形の~" 1 9 . 8 3 " 5
Odd "フ、, 9 9 0 . 0 - " " ャ f ッ , A T R I " " ス ロ ト " 0 0 0 0 0 0 " = D ッ , フ , 9 9 0 . 1
レコード 2 = ANK:JIS8/ASCII 漢字:シフトJIS
0~ 209 0-----1-----2-----3-----4-----5-----6-----7-----8-----9-----
16進High A222554455222BDCB223323332332322849586958F8C9522333323323300223222848A9A2224455554422CCDCABD
Low F2C2521E532C2377D2C14E540C15C5E3C23D0F1A36B32C0F2C2000E08E31DA2082C2A3940F2C2E50545E52C28CF1D0D
ANK "","URANUS","ウヌ",14.540,15.5.3,"M_jiVió7i",2000.08.31.."8","い","NEPUTUNE","ネプチューン",17.
漢字Even "","U A U " " う 入 , 4 5 0 1 , . . , 半神) 天空の神" 2 0 . 8 3 " 8 , 海王星" E U U E , ネ ユ ン
Odd "","R N S , ウヌ" 1 . 4 , 5 5 3 " M _ j y . フ _ , 0 0 0 . 1 " " C , ッ , N P T N " " フチー"

```

----- ファイルの最後まで表示しました。 -----

表示タイプ(D): ランダム表示 コード(C): シフトJIS フォント(F)... ファイル出力(P)... OK

↑ <ランダム表示>

<テキストのみ表示> ↓

ファイル内容表示

ファイル名: C:\Program Files\FujitsuBSC\FTRANP\Samples\PLANET\PLANET.CSV

```

レコード 0: "1","水星","MERCURY","マキリ","0.055,0,-2.4,"口神) 神の使者",1998.04.30
レコード 1: "2","金星","VENUS","ヒナス","0.815,0,-4.7,"口神) 愛と美の女神",1998.08.31
レコード 2: "3","地球","EARTH","アース","1.000,1,0.0,"n. 大地",1998.12.31
レコード 3: "4","火星","MARS","マーズ","0.107,2,-3.0,"口神) 戦いの神",1999.04.30
レコード 4: "--","小惑星","ASTEROID","アロイト","0.000,0,0.0,"a. 星形の~",1999.08.31
レコード 5: "5","木星","JUPITER","ジュピター","317.832,16,-2.8,"口神) 神々の主神",1999.12.31
レコード 6: "6","土星","SATURN","サターン","95.160,17,-0.5,"口神) 農耕の神",2000.04.30
レコード 7: "7","天王星","URANUS","ウラヌ",14.540,15.5.3,"半神) 天空の神",2000.08.31
レコード 8: "8","海王星","NEPUTUNE","ネプチューン",17.150,8,7.8,"口神) 海の神",2000.12.31
レコード 9: "9","冥王星","PLUTO","プルト",0.002,1,13.6,"半神) 冥界の神",2001.04.30

```

----- ファイルの最後まで表示しました。 -----

表示タイプ(D): テキストのみ表示 コード(C): シフトJIS フォント(F)... ファイル出力(P)... OK

ファイル内容表示は、表示できる容量に制限があります。全ての内容が表示されている場合はデータの最後に、

----- ファイルの最後まで表示しました. -----

全ての内容が表示できなかった場合には、

----- ファイルの途中で表示を切りました. -----

と表示されます。

4.12 環境設定

F*TRAN+の動作に必要な設定ファイルには以下のものがあります。

動作環境情報ファイル (FP. SYS、ファイル名固定)
 主設定ファイル (FTRAN. INI、ファイル名固定)
 コード変換表ファイル (拡張子が. CCT)
 漢字対応表ファイル (拡張子が. KKT)

この中で動作環境情報ファイルは、以下の場所にあるファイルを使用しています。

`%USERPROFILE%\AppData\Roaming\FTRANP¥`

それ以外のファイルは初期状態として以下の場所にあるファイルを使用しています。

`<ユーザデータフォルダ>%EnvGroup¥Env¥`

これらの動作環境ファイル以外の設定ファイルが集まったフォルダを「**環境**」と呼び、F*TRAN+ではこの環境を任意に切り替えて実行することが可能となっています。複数の環境が集まったフォルダを「**環境群フォルダ**」と呼びます。環境設定では、環境の作成・切り替え、環境群フォルダの設定などが可能です。まず、環境群フォルダを選び、その中から使用する環境を選ぶという順番で環境を指定します。環境群フォルダ、および環境の省略値はインストールモード別に以下のようになっています。

環境群フォルダ : `<ユーザデータフォルダ>%EnvGroup¥`
 環境 : `Env`

◆注意 ---- 環境の指定は必須

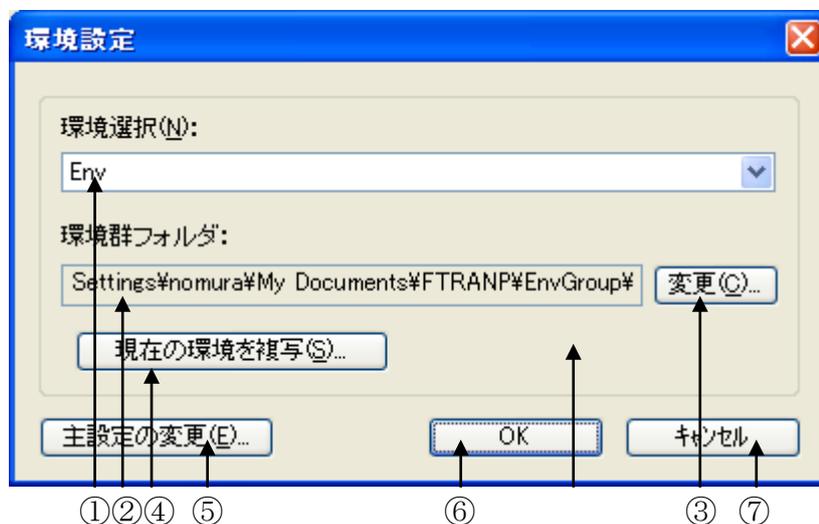
環境に必ず任意のフォルダを指定する必要があります。
 環境群フォルダ直下を環境としては使用できません。

◆注意 ---- 「INIT.SYS」について

INIT.SYS を利用すると、F*TRAN+を実行する際に%USERPROFILE%の情報を取得する為に環境変数を参照しなくなり、INIT.SYS に定義された情報を参照するようになります。詳細はインストールフォルダに格納される README.txt 内「2.0 INIT.SYS 利用方法 (環境変数を参照せずにF*TRAN+を動作させる方法について)」をご参照下さい。

■環境設定

メニューバーの設定 (S) →環境設定 (K) を実行すると、つぎのウインドウが開きます。



- ①使用する環境を選択します。
- ②現在指定されている環境群フォルダが表示されます。
- ③環境群フォルダを変更するボタンです。フォルダの選択画面が表示されます。
- ④現在選択されている環境を複写します。環境複写画面が表示されます。
- ⑤主設定変更画面が表示されます。各種エンコーディングや半角／全角設定の変更などできます。
- ⑥OKボタンをクリックすると、修正した内容が有効になります。
- ⑧キャンセルボタンをクリックすると、修正した環境設定を無効にし、環境設定のウインドウを閉じます。なお、主設定の変更画面でOKボタンを押してもここでキャンセルを選ぶと全ての修正が無効になります。複写した環境はそのまま残ります。

●現在の環境を複写

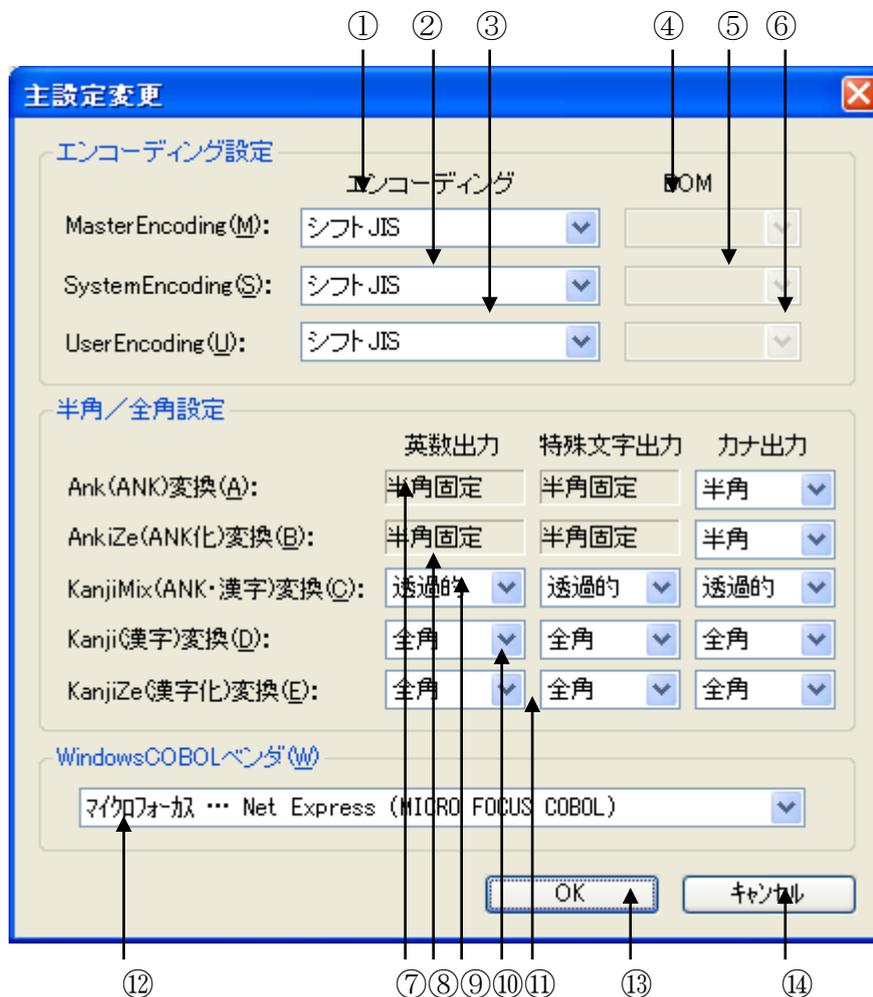
環境設定画面で『現在の環境を複写 (S)』ボタンを押すと以下の画面が表示されます。



複写先環境名を入力して複写ボタンを押すと、現在選択されている環境が複写されます。ここでは環境を複写するだけです。複写した環境は環境設定の環境選択 (N) プルダウンに追加されていますので切り替えて使用できます。

●主設定の変更

環境設定画面で『主設定の変更 (E)』ボタンを押すと以下の画面が表示されます。



- ①マスターエンコーディングを設定します。変換時、Windows側の省略時コードとなります。メイン画面の設定から一時的に変更する事もできます。
- ②システムエンコーディングを設定します。変換結果表示ファイルなど、F*TRAN+が自動で出力するファイルのエンコーディングに使われます。
- ③ユーザエンコーディングを設定します。スクリプトファイル、パラメータファイルなど、ユーザが保存するファイルのエンコーディングの省略時コードとなります。保存画面で一時的に変更する事もできます。
- ④マスターエンコーディングがUCS-2、UTF-8の時、BOMの扱いを、BOMあり、BOMなし、BOM自動の3種類から選択します。
- ⑤システムエンコーディングがUCS-2、UTF-8の時、BOMの扱いを、BOMあり、BOMなし、BOM自動の3種類から選択します。
- ⑥ユーザエンコーディングがUCS-2、UTF-8の時、BOMの扱いを、BOMあり、BOMなし、BOM自動の3種類から選択します。

- ⑦Ank変換時のカナ出力を制御します。半角、透過、全角から選択します。
- ⑧Ank化変換時のカナ出力を制御します。半角、透過、全角から選択します。
- ⑨Ank・漢字変換時の英数・特殊・カナ出力を制御します。半角、透過、全角から選択します。
- ⑩漢字変換時の英数・特殊・カナ出力を制御します。半角、透過、全角から選択します。
- ⑪漢字化変換時の英数・特殊・カナ出力を制御します。半角、透過、全角から選択します。
- ⑫Windows COBOLのベンダを選択します。
Windows COBOLのゾーン形式データを変換する時に重要になります。
- ⑬OKボタンをクリックすると、修正した主設定設定が有効になります。
環境設定画面でOKボタンをクリックすると主設定ファイルに変更が反映されます。
キャンセルボタンをクリックすると修正が無効になります。
- ⑭キャンセルボタンをクリックすると、修正した主設定を無効にし、主設定変更のウインドウを閉じます。

●エンコーディングの設定

Windowsファイルについて、そのコード系の指定は重要です。F*TRAN+では次の3種類のファイルに対するエンコーディングが設定できます。

・マスターエンコーディング (MasterEncoding)

変換元、変換先Windowsファイルのコード系を指定します。

・システムエンコーディング (SystemEncoding)

F*TRAN+が自動で出力する以下のファイルのエンコーディングに使われます。
変換結果表示ファイル

・ユーザエンコーディング (UserEncoding)

ユーザが保存する以下のファイルのエンコーディングの省略時コードとなります。
スクリプトファイル
パラメータファイル
ファイル内容表示時のファイル出力
Winファイル編集でのファイル出力

コード系は以下の4種類から選択できます。

・シフトJIS

・Unicode (UCS-2 LE)

・Unicode (UCS-2 BE)

・Unicode (UTF-8)

更に、コード系がUnicodeの場合、BOMの扱いを以下の3種類から選択できます。

・ **BOMあり**

入力の場合、BOMを検査します。UCS-2では、BOMがあればその値に従ってファイルを読み込みます。BOMがなければ指定されたエンディアンで読み込みます。UTF-8では、BOMがあれば読み捨てます。
出力の場合、BOMを付けます。

・ **BOMなし**

入力の場合、ファイルにはBOMがないものとみなして読み込みます。
出力の場合、BOMを付けません。

・ **BOM自動**

入力の場合、BOMを検査します。UCS-2では、BOMがあればその値に従ってファイルを読み込みます。BOMがなければ指定されたエンディアンで読み込みます。UTF-8では、BOMがないものとみなして読み込みます。
出力の場合、UCS-2では指定されたエンディアンのBOMを付けます。UTF-8では、BOMを付けません。

BOMの値は以下の通りです。

```
UCS-2 LE → FFFE H
UCS-2 BE → FEFF H
UTF-8   → EFBBBF H
```

● **半角／全角設定**

F*TRAN+では半角文字、全角文字の変換について**半角／全角／透過**の3通りの設定が可能になっています。以下の5つの変換型に対して設定が可能です。

- ・ Ank変換
- ・ Ank化変換
- ・ Ank・漢字変換
- ・ 漢字変換
- ・ 漢字化変換

Ank変換、Ank化変換では半角カナに対してのみ設定が可能で、英数・特殊文字は半角固定です。

4.13 変換設定

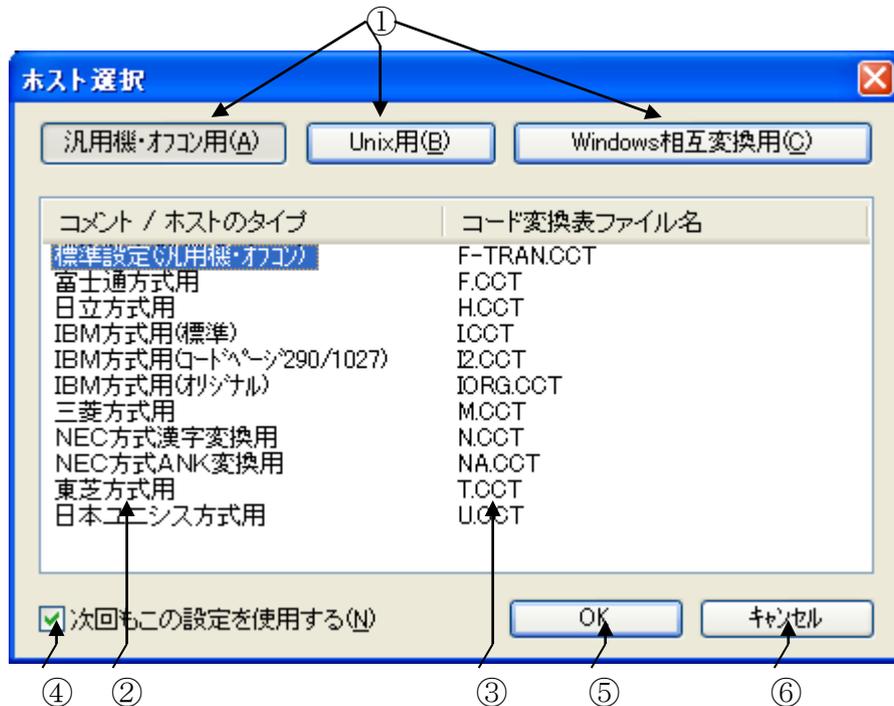
変換設定には、コード変換等に関するさまざまな設定項目があります。この設定項目は、メモリ上のコード変換表の中身です。変更した設定は即座に変換処理の動作に反映されます。

通常、ホスト選択で指定したコード変換表に対して、その中身を設定し、自分の運用に適した設定に変更していきます。その作業は、ふつうインストールの直後に1回だけ行います。しかし、運用に入ってからでも随時、設定の変更ができます。

データ交換の相手システムが複数あるときは、コード変換表の名前と設定を変えて、相手システムごとに切り替えて使うと便利です。

■ホスト選択

メニューバーの設定 (S) → ホスト選択 (H) を実行すると、つぎのウインドウが開きます。ここでは、基準となるホストを選択します。



①ホスト系を汎用機・オフコン用 (A)、Unix用 (B)、Windows相互変換用 (C) のボタンで選択します。

②選択できるホストのタイプが表示されるエリアです。

使用しているホストのタイプをクリックして選択します。

③実際に設定されているコード変換表ファイル名が表示されるエリアです。

④選択したホストの情報を保存するためのチェックボックスです。

次回にF*TRAN+を起動した時に、選択したホストを基準にする場合は、ONにします。

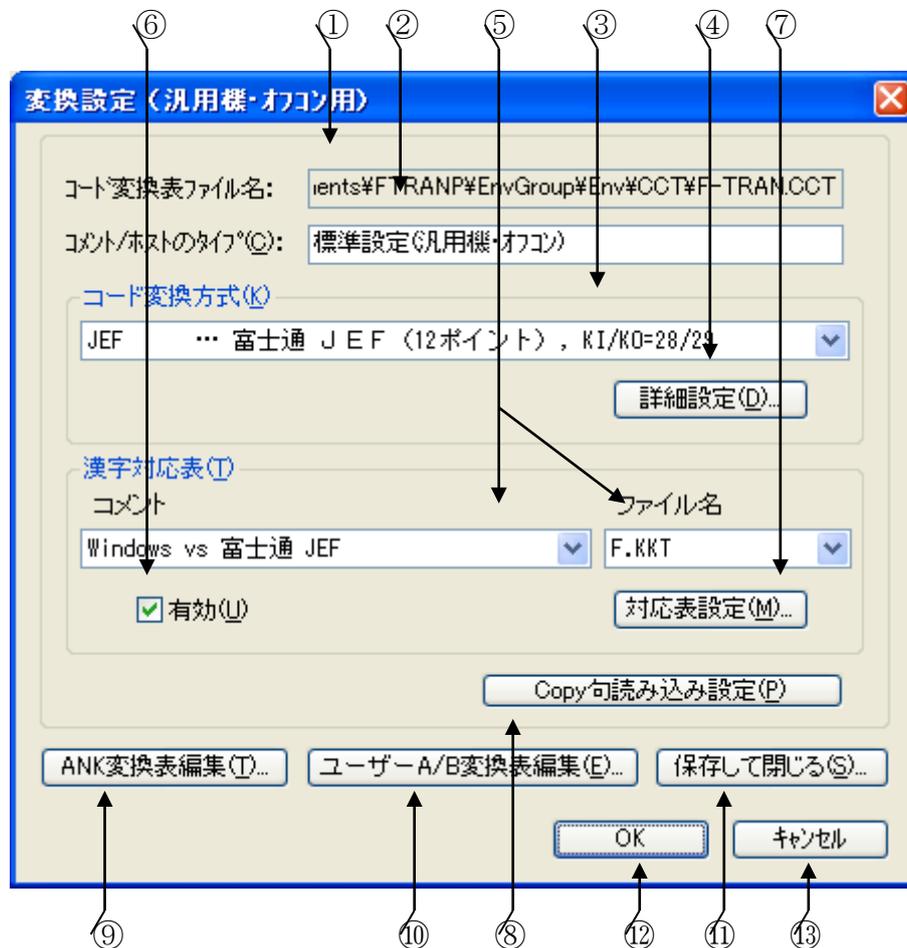
一時的にホストを変更する場合は、OFFにします。デフォルトは、ONです。

⑤②④の選択が完了した状態でOKボタンをクリックすると、基準となるホストが確定します。

⑥キャンセルボタンをクリックすると、ホスト選択を中止します。

■変換設定

メニューバーの設定 (S) →変換設定 (S) を実行すると、つぎのウインドウが開きます。



- ①読み込まれているコード変換表のファイル名が表示されています。
- ②コメントの変更ができます。
コード変換表は、40文字以内のコメントをつけることができます。
- ③コード変換方式の選択項目です。どのコード変換方式を使うかを割り当てます。
- ④コード変換方式の設定変更ボタンです。コード変換方式の設定内容を細かく変更できます。
- ⑤漢字変換時に使われる漢字対応表を選択することができます。
- ⑥選択されている漢字対応表を有効にするチェックボックスです。
- ⑦漢字対応表エディタを起動するボタンです。漢字対応表の修正ができます。
- ⑧COBOLのCopy句を読み込むときの設定を行うウインドウが開きます。
詳細は、マルチレコード編のマニュアルを参照してください。
- ⑨ANK変換表の編集ボタンです。

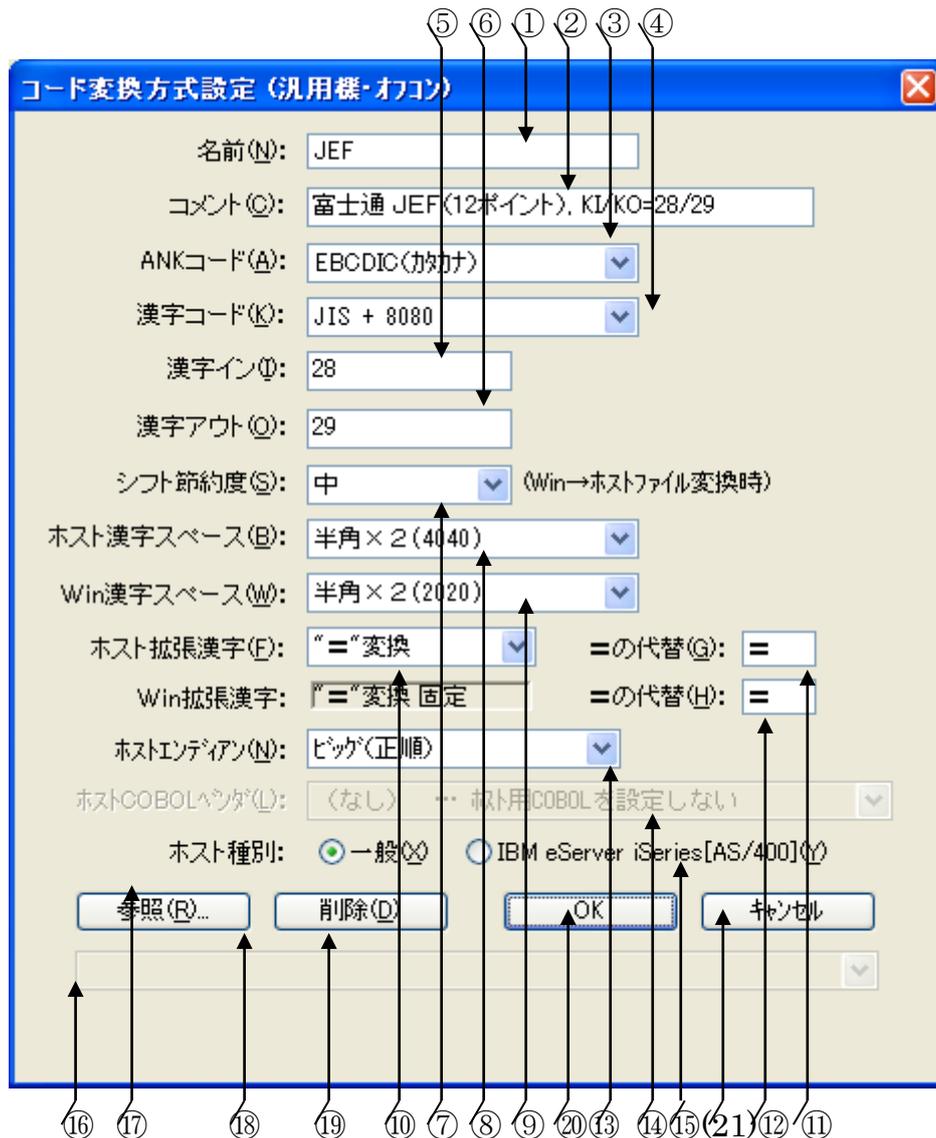
JIS8/ASCII、Unicode (半角・全角)、EBCDIC (カナ・英小文字) それぞれが対応付けられている変換表の表示・修正ができます。

- ⑩ユーザーA/B変換表の編集ボタンです。
2つの、ユーザが任意に定義できるANK変換表（1バイト単位のコード変換表）の表示・修正ができます。
- ⑪コード変換表の書き込み（保存）を行い、変換設定のウインドウを閉じます。
修正したコード変換表を元のファイルに書き戻すだけでなく、別の名前のコード変換表としてディスクに書き込むこともできます。
- ⑫OKボタンをクリックすると、修正した変換設定が有効になります。
ただし、あくまでメモリ上の変更であり、コード変換表ファイルに書き込まれるわけではありません。
- ⑬キャンセルボタンをクリックすると、修正した変換設定を無効にし、変換設定のウインドウを閉じます。なお、ユーザーA/B変換表編集画面、ANK変換表編集画面、コード変換方式の詳細設定画面などでOKボタンを押してもここでキャンセルを選ぶと全ての修正が無効になります。

●コード変換方式の設定

変換処理時に使う漢字変換方式を選択します。選択後に、詳細設定 (S) のボタンをクリックすると、コード変換方式設定のウィンドウが開き、細かな設定内容を変更することができます。新規に作成する場合は、「新規作成」を選んでから詳細設定のボタンをクリックします。ただし、登録できるのは合計で15種類までです。

<ホストが汎用機・オフコンの場合>



- ①このコード変換方式につける名前です。英数字8文字以内で設定できます。
- ②設定内容をわかりやすくするためのコメントです。40文字以内で設定できます。
- ③ANKコードを **JIS8/ASCII**、**EBCDIC (英小文字)**、**EBCDIC (カタカナ)** から設定します。

この設定でホストのANKコードが何かを設定します。全てのANK部の変換や、漢字スペースのコードを決めるためなどに使用されます。

④漢字コードのタイプを設定します。以下の中から選択します。

J I S、J I S+8080、J I S+8080 (ユニシス)、東芝 (漢字コード自体は J I S漢字)、NEC-J I P S、NEC-内部コード、I B M

J I S漢字の定義域は2121H~7E7EHです。J I S+8080は、J I Sに8080Hを加えて定義域をA1A1H~FEFEHにずらしています。多くのホストシステムがJ I S+8080を採用しています。その理由の1つは、J I S漢字との相互変換が極めて簡単であることです。2つ目の理由は、ANKコードとしてEBCDICコードを採用していることが多いため、J I S漢字をそのまま採用すると、EBCDICコードの制御文字のうち、21H~3FHがJ I S漢字の一部とぶつかってしまうからです。以上の2タイプと比較すると、NEC漢字はかなり特殊です。NEC漢字には、

J I P S (E)、J I P S (J)、内部コード (E)、内部コード (J)

の4通りのコード系があります。J I S8/ASCII系かEBCDIC系かは、③のANKコードの設定で間接的に決められます。

()内のEは、EBCDIC系、Jは、J I S8系を意味しています。

I B M漢字は、I B M漢字←→シフトJ I S漢字用の内部テーブルを持って対応しています。

⑤漢字インのシーケンスを16進で6桁 (3バイト) 以内で指定します。

⑥漢字アウトのシーケンスを16進で6桁 (3バイト) 以内で指定します。

⑦Win→ホストファイル変換時に、Ank・漢字 (KI/KO付) 混在変換するときの、シフト節約度を設定します。弱、中、強の3通りの設定ができます。

[弱] 半角スペースはあくまでANK、全角スペースはあくまで漢字、として完全に区別して扱います。この設定にすると、頻繁にKI/KOが入る傾向があり、レコード長や項目長のオーバーフローを起こすことが多くなります。ただし、Windowsファイル側で半角スペースと全角スペースを完全に使い分けしているときは、この設定が一番よいこともあります。

[中] 半角スペース2個と全角スペースを同一視し、なるべくANKモード、漢字モードが長く続くようにします。いい替えれば、ANK文字に続く半角スペースはANKとして、漢字に続く半角スペースは2個1組で漢字スペースとして扱うということです。ただし、漢字モード中で奇数個の半角スペースが続くと、最後の1個だけはANKとして扱います。この設定が、いろいろな意味で無難です。

[強] 半角スペース2個と全角スペースを同一視し、なるべくANKモード、漢字モードが長く続くようにします。また、ホストファイル側の漢字モード中に奇数個の半角スペースが現れてもよしとします。この設定のデータを受け付けることのできるシステムやソフトは少ないので、注意してください。なお、ANKがEBCDIC、漢字がJ I S漢字の組み合わせのときは、この設定にははいけません。スペースが40××Hというコードの漢字に化けてしまいます。

例をあげます。△はANKスペース、△は漢字スペースです。弱→中→強の順にK I / K Oの節約になり、オーバーフローしにくくなるのがわかると思います。

データ ---- E A R T H△△△水の惑星△△△地球△△
 (W i n→ホスト方向、A n k・漢字 (K I / K O付) 混在変換)

↓

弱 ---- E A R T H△△△<K I>△水の惑星<K O>△△△<K I>地球<K O>△△
 中 ---- E A R T H△△△△<K I>水の惑星△<K O>△<K I>地球△<K O>
 強 ---- E A R T H△△△△<K I>水の惑星△△地球△<K O>

- ⑧W i n→ホストファイル変換時に、漢字スペースを全角のコードにするか、半角×2個のコードにするかを設定します。具体的なコードは、ANKコードと漢字コードの組み合わせによって、つぎの表のように決まります。

漢字スペースのコード (W i n→ホスト変換時)

漢字スペース		全 角	半角 × 2
ANKコード&漢字コード			
J I S 8 / A S C I I	J I S	2 1 2 1 H	2 0 2 0 H
	J I S + 8 0 8 0	A 1 A 1 H *1	2 0 2 0 H
	NEC J I P S (J)	2 1 2 1 H	2 0 2 0 H
	NEC内部コード (J)	6 0 2 1 H	2 0 2 0 H
	I B M漢字	4 0 4 0 H	2 0 2 0 H
E B C D I C	J I S	2 1 2 1 H	禁止 *2
	J I S + 8 0 8 0	A 1 A 1 H *1	4 0 4 0 H
	NEC J I P S (E)	4 F 4 F H	4 0 4 0 H
	NEC内部コード (E)	7 9 4 F H	4 0 4 0 H
	I B M漢字	4 0 4 0 H	4 0 4 0 H

*1) J I S + 8 0 8 0 (ユニシス) の場合は、2 0 2 0 Hにも設定可能

*2) もし半角×2の設定にすると、漢字スペースが「誓」の字に化ける

- ⑨ホスト→W i nファイル変換時に、W i n d o w s ファイル側の漢字スペースを、全角のコードにするか、半角×2のコードにするかを設定します。W i n d o w s 側のコードがU n i c o d eの場合は半角×1も選べます
 ふつうは、半角×2の設定のほうがよいのですが、パソコンのソフトによっては、全角の漢字スペースしか受けつけないものがあります。
- ⑩ホストファイル側の漢字対応表に定義されないホスト拡張漢字 (ユーザー定義文字を含む) の扱い方を設定します。
 ゲタ変換、クエスチョン変換と呼ぶ2通りの設定ができます。

【“=” 変換 (ゲタ変換)】

ホスト漢字を“=” (ゲタ) に置き換える。

“=” 以外の全角文字にも変換可能。

ホスト→W i n方向の変換時に有効。元のコードには戻せない。

桁ずれは起こさない。

【“?(xxxx)” 変換 (クエスチョン変換)】

?は全角。xxxxは半角4桁の16進数でホストコードを表す。

ホスト←→W i n双方向で有効。

エディタで処理するのに都合がよく、扱いやすい。

ただし、2バイト←→8バイト (UCS-2の場合は14バイト) の変換なので、桁ずれを起こすことに注意。

変換の対象になるのは、句点コードでいえば1区1点～94区94点以外の漢字コードです。

J I S漢字なら2121H～7E7EH以外の漢字コード、

J I S+8080タイプならA1A1H～FEFEH以外の漢字コードです。

⑪ゲタ変換のとき、“=” に代わる別の全角文字を設定できます。

また、半角スペース2個の設定にもできます。

代わりの文字の設定を解除して“=”に戻りたいときは、半角の“=”を入力します。

⑫W i n d o w sの拡張漢字 (95区以上の漢字) に対しては、“=” 変換固定です。

“=” に代わる別の全角文字を設定できます。また、半角スペース2個の設定にもできます。

代わりの文字の設定を解除して“=”に戻りたいときは、半角の“=”を入力します。

⑬ホストエンディアン (ホストの2進項目のデータ格納順) を選択します。

⑭ホストがU n i x、W i n d o w sの場合、ホストC O B O Lのベンダを選択します。

ホストが汎用機・オフコンの場合は選択できません。

⑮ホスト種別を選択します。I B Mのオフコン「eServer i5」「eServer iSeries」「AS/400」のどれかであるか、その他の一般的なホストであるかを設定します。

⑯入力ガイダンス・メッセージが表示されるフィールドです。

⑰参照 (R) ボタンは、既存のコード変換方式設定の中から1つを選択して、

①～⑮の項目を基準値にする機能です。

おもに、既存値をベースに「新規作成」を行う場合に重宝します。

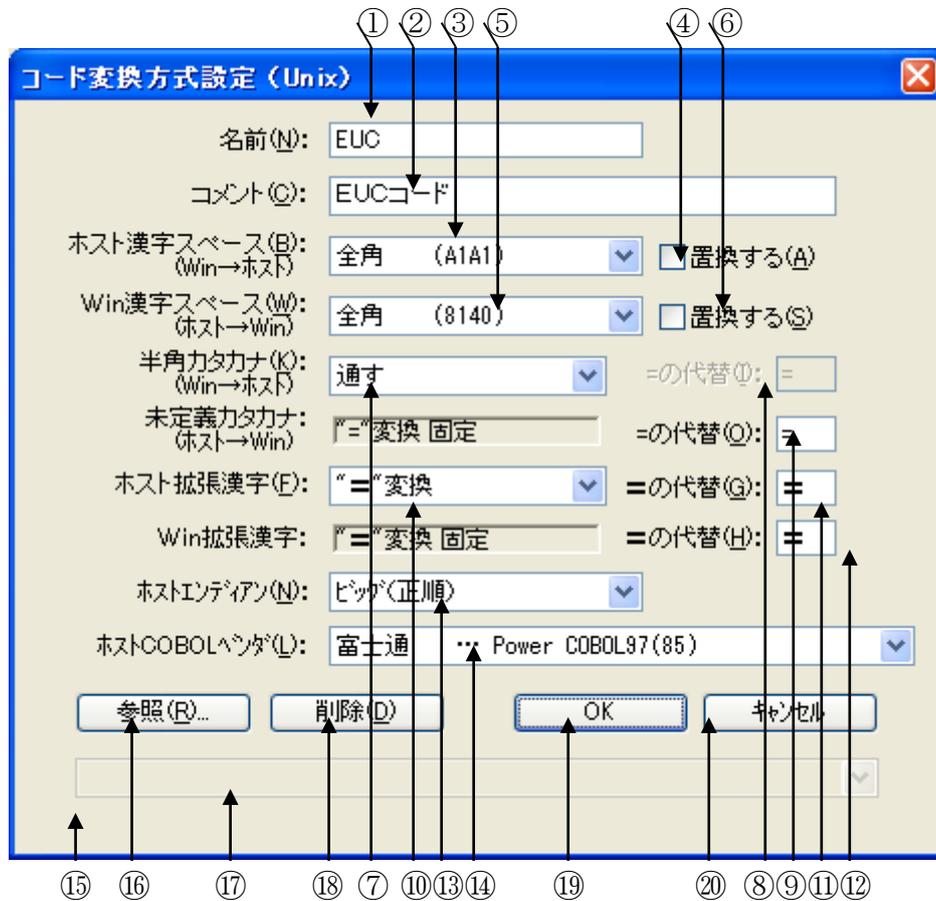
⑱⑰の参照ボタンをクリックすると、既存のコード変換方式設定の参照ダイアログが開きます。

⑲削除 (D) ボタンをクリックすると、表示されているコード変換方式設定を削除し、設定ウインドウを閉じます

⑳OKボタンをクリックすると、ウインドウ上のコード変換方式設定が有効になります。

(21)キャンセルボタンをクリックすると、ウインドウ上のコード変換方式設定を無効にし、設定のウインドウを閉じます。

<ホストがUnixの場合>



- ①このコード変換方式につける名前です。英数字8文字以内で設定できます。
- ②設定内容をわかりやすくするためのコメントです。40文字以内で設定できます。
- ③Win→ホストファイル変換時に、Unixファイル側の漢字スペースを、
全角=A1A1Hのコードにするか、半角×2=2020Hのコードにするかを設定します。
- ④Win→ホストファイル変換時に、漢字スペースの置換をする／しないを選択します。
- ⑤ホスト→Winファイル変換時に、Windowsファイル側の漢字スペースを、全角のコードにするか、半角×2のコードにするかを設定します。Windows側のコードがUnicodeの場合は半角×1も選べます
ふつうは、半角×2の設定のほうがよいのですが、パソコンのソフトによっては、全角の漢字スペースしか受けつけないものがあります。
- ⑥ホスト→Winファイル変換時に、漢字スペースの置換をする／しないを選択します。
- ⑦Win→ホストファイル変換時の半角カナ文字を、通す、“=”変換の3つの変換方法から
選択できます。
- ⑧⑦で“=”変換を選択すると、“=”に代わる別の半角文字を設定できます。
- ⑨ホスト→Winファイル変換時の未定義半角カナ文字に対しては、“=”変換固定です。
“=”に代わる別の半角文字を設定できます。

- ⑩Unixファイル側の漢字対応表に定義されないホスト拡張漢字（ユーザー定義文字を含む）の扱い方を設定します。ゲタ変換、クエスチョン変換と呼ぶ2通りの設定ができます。

["=" 変換 (ゲタ変換)]

ホスト漢字を“=” (ゲタ) に置き換える。

“=” 以外の全角文字にも変換可能。

ホスト→Win方向の変換時に有効。元のコードには戻せない。

桁ずれは起こさない。

["? (xxxx)" 変換 (クエスチョン変換)]

?は全角。xxxxは半角4桁の16進数でホストコードを表す。

ホスト↔Win双方向で有効。

エディタで処理するのに都合がよく、扱いやすい。

ただし、2バイト↔8バイト (UCS-2の場合は14バイト) の変換なので、桁ずれを起こすことに注意。

変換の対象になるのは、句点コードでいえば1区1点～94区94点以外の漢字コードです。EUCならA1A1H～FEFEH以外の漢字コードです。

- ⑪ゲタ変換のとき、“=” に代わる別の全角文字を設定できます。

また、半角スペース2個の設定にもできます。

代わりの文字の設定を解除して“=”に戻りたいときは、半角の“=”を入力します。

- ⑫Windowsの拡張漢字（95区以上の漢字）に対しては、“=” 変換固定です。

“=” に代わる別の全角文字を設定できます。また、半角スペース2個の設定にもできます。

代わりの文字の設定を解除して“=”に戻りたいときは、半角の“=”を入力します。

- ⑬ホストエンディアン（ホストの2進項目のデータ格納順）を選択します。

- ⑭ホストCOBOLのベンダを選択します。

ホストCOBOLのゾーン形式データを変換する時に重要になります。

- ⑮入力ガイダンス・メッセージが表示されるフィールドです。

- ⑯参照 (R) ボタンは、既存のコード変換方式設定の中から1つを選択して、

①～⑭の項目を基準値にする機能です。

おもに、既存値をベースに「新規作成」を行う場合に重宝します。

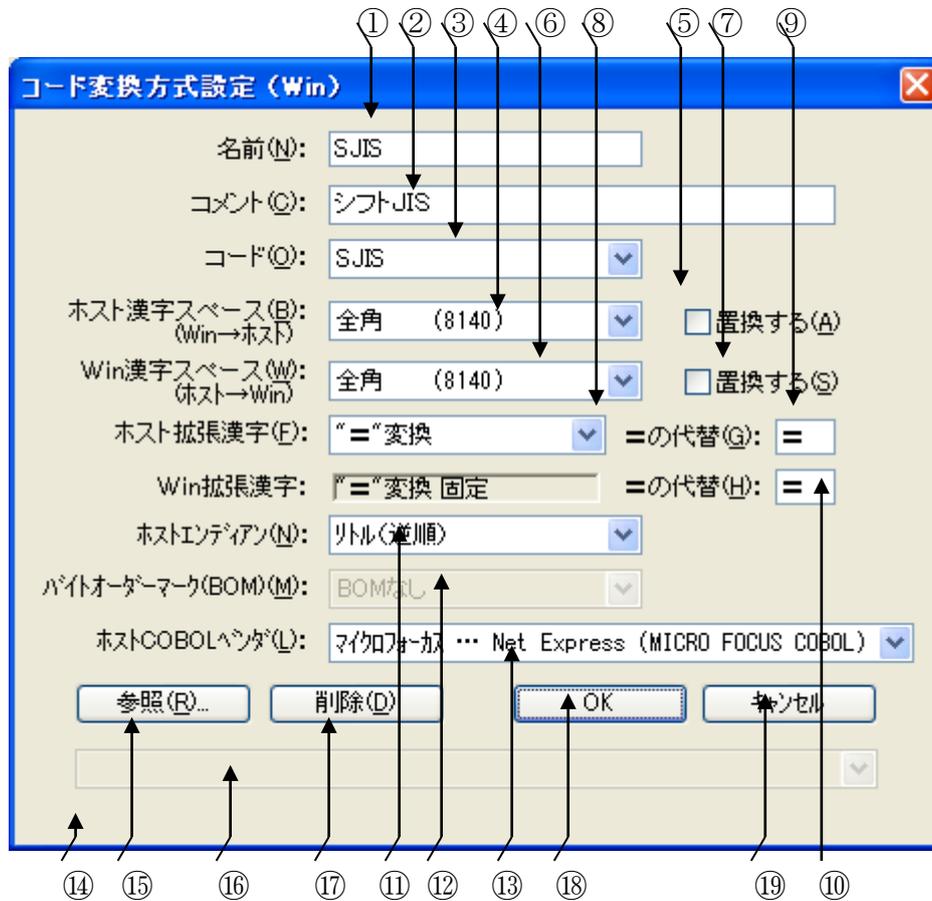
- ⑰⑯の参照ボタンをクリックすると、既存のコード変換方式設定の参照ダイアログが開きます。

- ⑱削除 (D) ボタンをクリックすると、表示されているコード変換方式設定を削除し、設定ウインドウを閉じます

- ⑲OKボタンをクリックすると、ウインドウ上のコード変換方式設定が有効になります。

- ⑳キャンセルボタンをクリックすると、ウインドウ上のコード変換方式設定を無効にし、設定のウインドウを閉じます。

<ホストがWindowsの場合>



- ①このコード変換方式につける名前です。英数字8文字以内で設定できます。
- ②設定内容をわかりやすくするためのコメントです。40文字以内で設定できます。
- ③ホストのコードを指定します。シフトJIS、Unicode (UCS-2)、Unicode (UTF-8)の中から選択します。
- ④Win→ホストファイル変換時に、ホスト（出力）側の漢字スペースを、全角のコードにするか、半角×2のコードにするかを設定します。ホスト側のコードがUnicodeの場合は半角×1も選べます
- ⑤Win→ホストファイル変換時に、漢字スペースの置換をする／しないを選択します。
- ⑥ホスト→Winファイル変換時に、Win（出力）側の漢字スペースを、全角のコードにするか、半角×2のコードにするかを設定します。Windows側のコードがUnicodeの場合は半角×1も選べます
- ⑦ホスト→Winファイル変換時に、漢字スペースの置換をする／しないを選択します。
- ⑧ホストファイル側の漢字対応表に定義されないホスト拡張漢字（ユーザー定義文字を含む）の扱い方を設定します。ゲタ変換、クエスチョン変換と呼ぶ2通りの設定ができます。

["=" 変換 (ゲタ変換)]

ホスト漢字を“=” (ゲタ) に置き換える。

“=” 以外の全角文字にも変換可能。

ホスト→W i n方向の変換時に有効。元のコードには戻せない。

桁ずれは起こさない。

["? (xxxx)" 変換 (クエスチョン変換)]

?は全角。xxxxは半角4桁の16進数でホストコードを表す。

ホスト←→W i n双方向で有効。

エディタで処理するのに都合がよく、扱いやすい。

ただし、2バイト←→8バイト (UCS-2の場合は14バイト) の変換なので、桁ずれを起こすことに注意。

変換の対象になるのは、句点コードでいえば1区1点～94区94点以外の漢字コードです。

⑨ゲタ変換のとき、“=” に代わる別の全角文字を設定できます。

また、半角スペース2個の設定にもできます。

代わりの文字の設定を解除して“=”に戻りたいときは、半角の“=”を入力します。

⑩W i n d o w sの拡張漢字 (95区以上の漢字) に対しては、“=” 変換固定です。

“=” に代わる別の全角文字を設定できます。また、半角スペース2個の設定にもできます。

代わりの文字の設定を解除して“=”に戻りたいときは、半角の“=”を入力します。

⑪ホストエンディアン (ホストの2進項目のデータ格納順) を選択します。ホストコードがUCS-2の場合にはコードのエンディアンもこの情報が使われます。

⑫ホストのコードがU n i c o d eの時、バイトオーダーマーク (B O M) の扱いを指定します。BOMあり、BOMなし、BOM自動の中から選択します。

⑬ホストCOBOLのベンダを選択します。

ホストCOBOLのゾーン形式データを変換する時に重要になります。

⑭入力ガイダンス・メッセージが表示されるフィールドです。

⑮参照 (R) ボタンは、既存のコード変換方式設定の中から1つを選択して、

①～⑬の項目を基準値にする機能です。

おもに、既存値をベースに「新規作成」を行う場合に重宝します。

⑯⑮の参照ボタンをクリックすると、既存のコード変換方式設定の参照ダイアログが開きます。

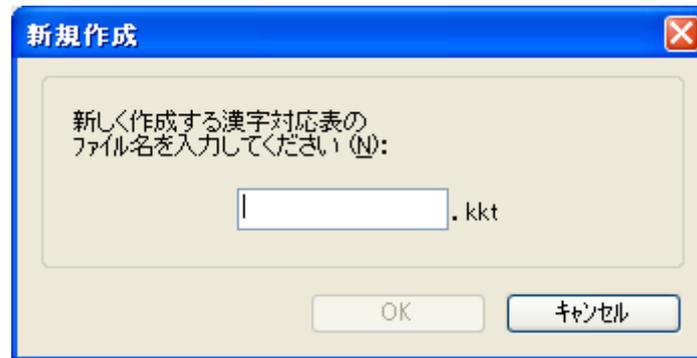
⑰削除 (D) ボタンをクリックすると、表示されているコード変換方式設定を削除し、設定ウインドウを閉じます

⑱OKボタンをクリックすると、ウインドウ上のコード変換方式設定が有効になります。

⑲キャンセルボタンをクリックすると、ウインドウ上のコード変換方式設定を無効にし、設定のウインドウを閉じます。

●漢字対応表の設定

漢字変換時に使われる漢字対応表を設定します。漢字体系に合った漢字対応表を選択し、有効のチェックボックスをONにします。対応表設定 (M) のボタンをクリックすると、漢字対応表エディタが起動し、設定内容の変更ができます。漢字対応表 (T) が“(新規作成)”の場合は、ファイル名の入力をした後に、漢字対応表エディタが起動します。



詳細は、4.14 漢字対応表エディタをご覧ください。

●HOST COBOL ベンダの設定

HOST COBOL のゾーン形式データを変換する場合は、この設定が重要になります。つぎの7種類から選択できます。選択しないこともできます。

富士通	Power COBOL 97 (85)
日立	COBOL 85
NEC	COBOL 85
マイクロフォーカス	MICRO FOCUS COBOL *
Acucorp	AcuCOBOL (コンパイルオプション : -Dca)
Acucorp	AcuCOBOL (コンパイルオプション : -Dci)
Acucorp	AcuCOBOL (コンパイルオプション : -Dcm)

*)ホストがWindowsの場合は、

マイクロフォーカス Net Express (MICRO FOCUS COBOL) です。

●Windows COBOLベンダの設定

Windows COBOLのゾーン形式データを変換する場合は、この設定が重要になります。つぎの7種類から選択できます。選択しないこともできます。

富士通	Power COBOL 97 (85)
日立	COBOL 85
NEC	COBOL 85
マイクロフォーカス	Net Express (MICRO FOCUS COBOL)
Acucorp	AcuCOBOL (コンパイルオプション : -Dca)
Acucorp	AcuCOBOL (コンパイルオプション : -Dci)
Acucorp	AcuCOBOL (コンパイルオプション : -Dcm)

●ホストエンディアン設定

2進形式データを変換する場合は、この設定が重要になります。ホスト2進形式のエンディアン(格納順)をつぎのどちらかに設定します。

- ビッグ(正順)
- リトル(逆順)

ホストのコードがUnicode(UCS-2)の場合はバイトオーダー順の指定としても使われます。

●バイトオーダーマーク(BOM)の設定

ホスト側のコードがUnicodeの場合、バイトオーダーマークの扱いを指定します。つぎの3種類から選択します。

- BOMあり
- BOMなし
- BOM自動

BOM自動の指定は、コードがUCS-2の場合はBOMありとして動作し、UTF-8の場合はBOMなしとして動作します。

Win→ホスト変換/ホスト→Win変換、テキストファイル/データファイル/ランダムファイルそれぞれの場合で動作が異なるので注意が必要です。基本的にランダムファイル変換の場合は入力側にBOMがあってもデータとして扱い、出力にはBOMを付けません。それ以外は入力側ではBOMありの場合はBOMを検査し、出力の場合はBOMを付けます。

● ANKコードの設定

ホストシステムには、J I S 8 / A S C I I 系のシステムと、E B C D I C 系のシステムがあります。さらに、E B C D I C コードには、カタカナ版と英小文字版があります。それらのどれに合わせるかを設定します。

この設定はとても重要です。というのは、J I S 8 / A S C I I 系にするか E B C D I C 系にするかで、単純な ANK 変換にとどまらず、バッファをクリアするコードや、ゾーン形式、パック形式の変換などにも影響を及ぼすからです。

● ANK変換表

3種類のANK変換表があります。

ANK変換表

ユーザー用 (A) 変換表

ユーザー用 (B) 変換表

このうち、ユーザー用 (A) 変換表、ユーザー用 (B) 変換表は、ふつうは使わない特別なものです。

<ANK変換表、ユーザーA/B変換表について>

J I S 8 / A S C I I、U n i c o d e (半角)、U n i c o d e (全角)、E B C D I C (カナ)、E B C I D C (英小文字) それぞれの対応関係を決定している変換表です。変換時にはこのテーブルが参照されます。

ある「字」に対して J I S 8 / A S C I I、U n i c o d e (半角)、U n i c o d e (全角)、E B C D I C (カナ)、E B C D I C (英小文字) が定義されています。U n i c o d e は U C S - 2 で表します。J I S 8 / A S C I I と U n i c o d e では英数カナは固定で割り当てていますが、その他の割り当ては変更が可能です。

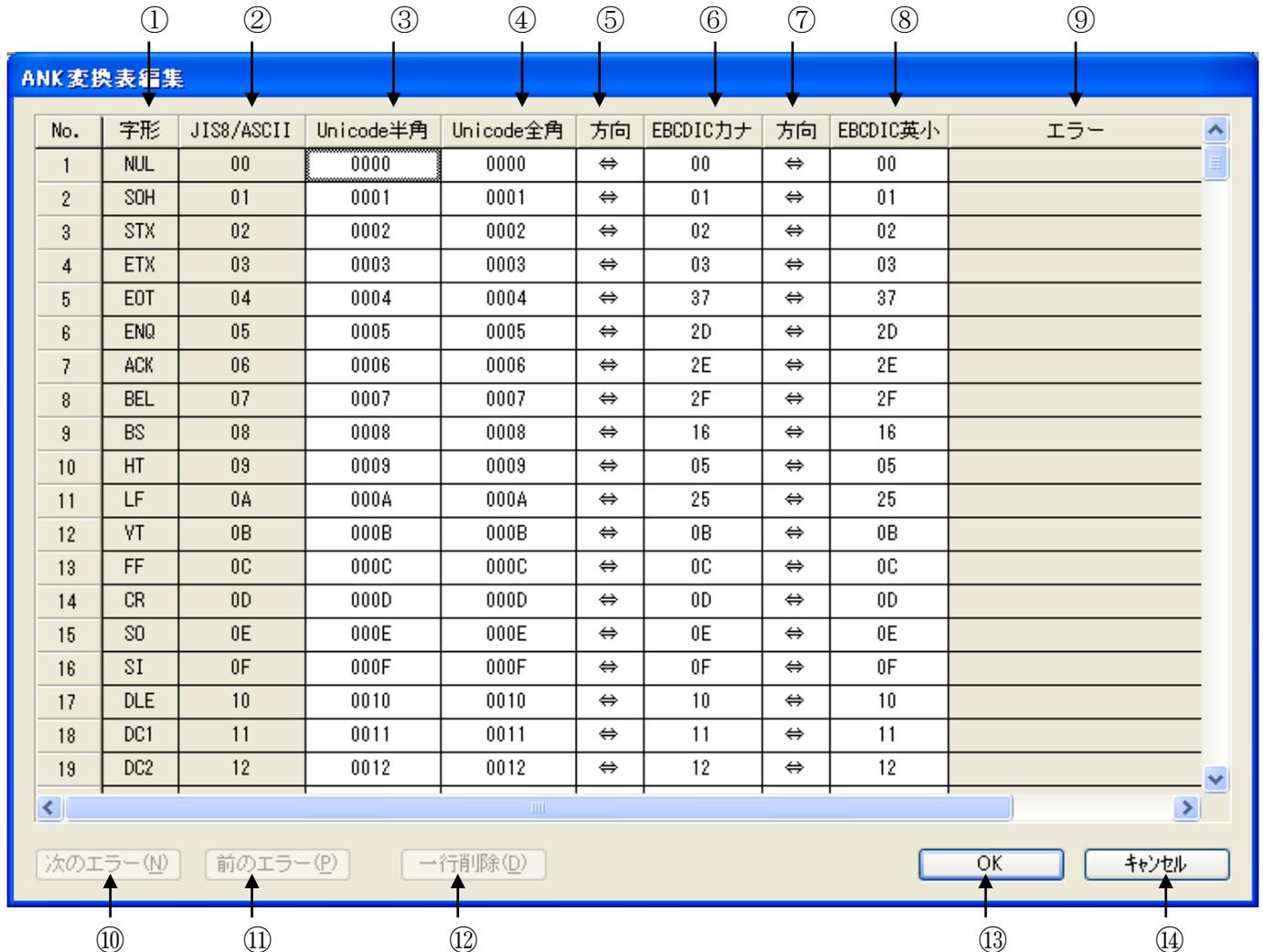
E B C D I C (カタカナ)、E B C D I C (英小文字) はそれぞれ方向の情報をもって割り当てられています。

ユーザー用 (A/B) 変換表は、利用者独自の特別な1バイト単位の変換処理が必要なときに使用します。ただし、この2枚はふつうのANK変換に使うものではありません。

なお、ユーザー用Aとユーザー用Bは、本当は互いに逆変換表になっているわけではありません。それぞれ独立した変換表です。

<ANK変換表について>

変換設定画面のANK変換表編集(T)ボタンをクリックすると、下のウィンドウが開きます。

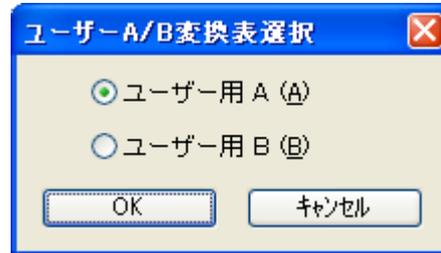


- ①字形が表示されています。制御記号は相当する意味がアルファベットで表現されています。この欄は変更できません。
- ②JIS8/ASCIIコードが表示されています。この欄は変更できません。
- ③英数カナ記号文字に対してUnicodeの半角コードが割り当てられています。英数カナ以外は変更が可能です。
- ④英数カナ記号文字に対してUnicodeの全角コードが割り当てられています。英数カナ以外は変更が可能です。
- ⑤EBCDIC (カナ) とJIS8/ASCII、Unicodeとの関係を指定します。
- ⑥JIS8/ASCII、Unicodeに対して割り当てるEBCDIC (カナ) コードを指定します。
- ⑦EBCDIC (英小文字) とJIS8/ASCII、Unicodeとの関係を指定します。

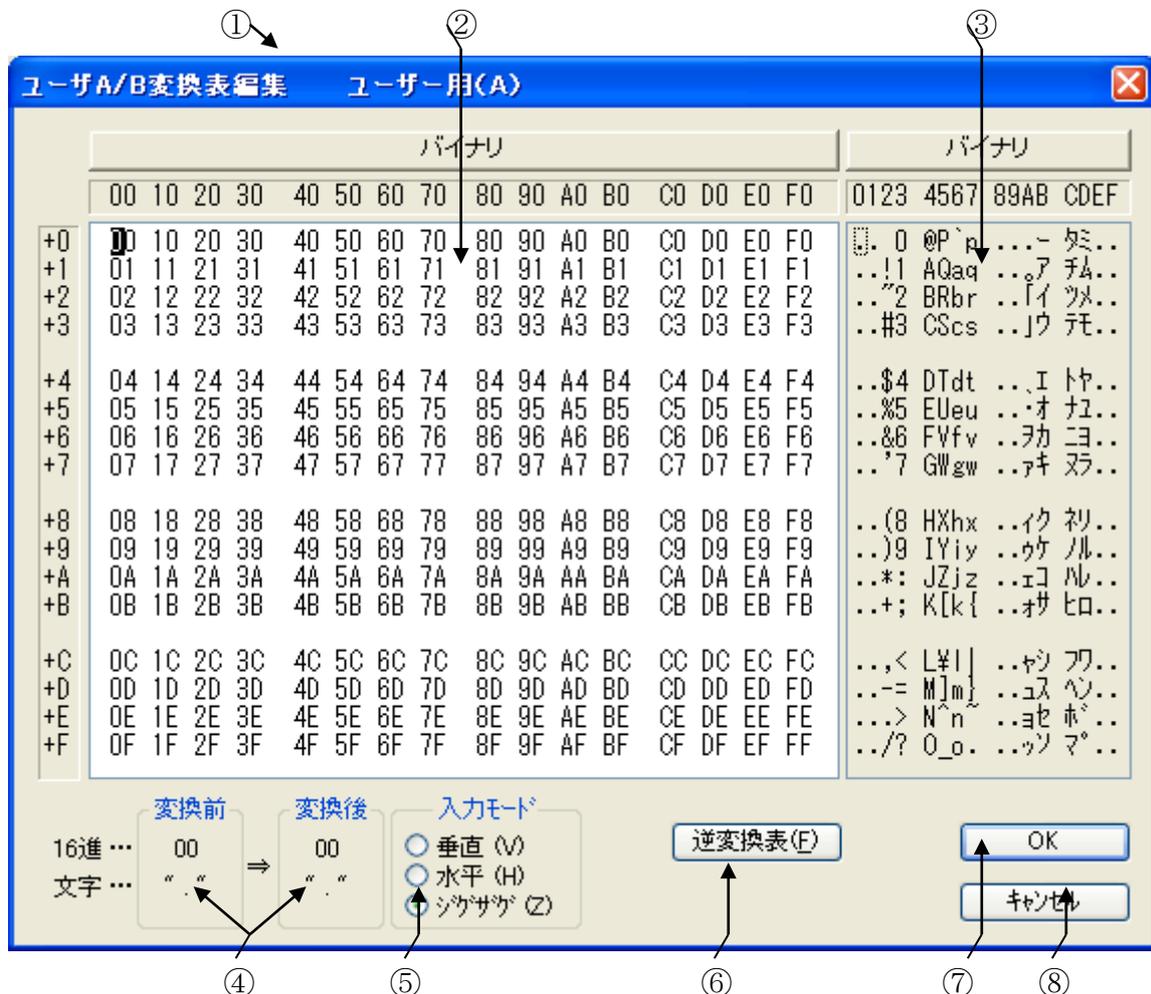
- ⑧ J I S 8 / A S C I I、U n i c o d e に対して割り当てる E B C D I C (英小文字) コードを指定します。
- ⑨ 各関連付けに矛盾があるところにエラーメッセージが表示されます。
エラーチェックは⑬のOKボタンをクリックすると行われます。
- ⑩ エラーが複数ある時にこのボタンをクリックすると、次のエラー項目にジャンプします。
- ⑪ エラーが複数ある時にこのボタンをクリックすると、前のエラー項目にジャンプします。
- ⑫ No. 257～320までのエリアで現在選択している行に登録されているデータを削除します。No. 1～256までは削除はできません。
- ⑬ OKボタンをクリックすると、修正したANK変換表が有効になります。ただし、変換設定画面でキャンセルを選ぶと修正は無効になります。ここでの修正はあくまでメモリ上の変更であり、コード変換表ファイルに書き込まれるわけではありません。
- ⑭ キャンセルボタンをクリックすると、修正したANK変換表を無効にし、ANK変換表のウィンドウを閉じます。

＜ユーザー用（A/B）変換表について＞

ユーザーA/B変換表編集（E）のボタンをクリックすると、下のウィンドウが開きます。編集するANK変換表を選択して、OKボタンをクリックします。



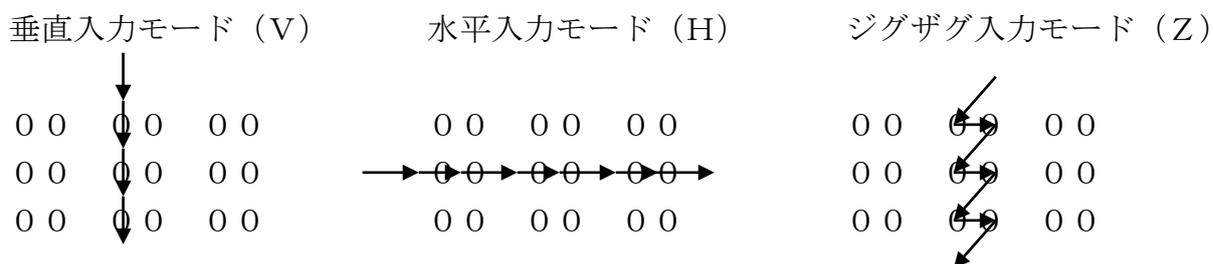
OKボタンをクリックされると、下のウィンドウが開き、編集モードに入ります。



- ①編集するANK変換表名が表示されます。
- ②変換表の内容が16進表示されるエリアです。変更入力することができます。
- ③変換表の内容が文字表示されるエリアです。
- ④変換元の16進コードと文字、変換後の16進コードと文字が表示されます。
カーソル移動とともに表示内容が変わります。
- ⑤変換表を変更入力する場合の入力モード（カーソルの動き）を指定するラジオボタンです。

入力モードとカーソルの動き

入力モードによるカーソルの動きの違いを示します。



- ⑥逆変換表に移るためのボタンです。
ユーザーA表⇄ユーザーB表相互の切り替えができます。
カーソル位置の内容が逆変換表のカーソルの位置になります。
- ⑨OKボタンをクリックすると、修正したANK変換表が有効になります。ただし、変換設定画面でキャンセルを選ぶと修正は無効になります。ここでの修正はあくまでメモリ上の変更であり、コード変換表ファイルに書き込まれるわけではありません。
- ⑩キャンセルボタンをクリックすると、修正したANK変換表を無効にし、ANK変換表のウィンドウを閉じます。

ユーザー用 (A/B) は、特別な変換処理を行う変換表です。バイト単位の変換ができます。利用者独自の処理が必要になったときに書き替えて使います。

使用目的 通常のEBCDIC↔JIS8/ASCII変換を使うべきでない、特殊な1バイト単位の変換が必要になる場合があります。たとえば、**特殊漢字↔JIS漢字変換**などが挙げられます。

利用できる所 ユーザー用 (A/B) 変換表は、**マップ設定オプションのユーザーA変換、ユーザーB変換**で利用できます。

出荷時設定 ユーザー用 (A/B) 変換表の出荷時の設定は、何もしない変換表、つまりバイナリ変換と同様になっています。

●コード変換表の書き込み（保存）

コード変換表の書き込み（保存）は、現在のメモリ上のコード変換表を元のコード変換表ファイルに保存、または別のコード変換表ファイルとしてディスクに書き込みます。

保存して閉じる (S) ボタンをクリックすると、“ファイル名を付けて保存” ウィンドウが開きます。保存するファイルを選択して、保存 (S) ボタンをクリックします。

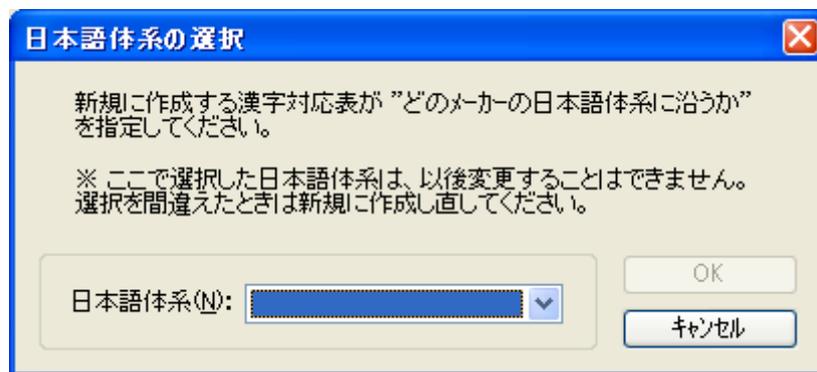
4.14 漢字対応表エディタ

漢字対応表エディタは、F * TRAN+の漢字コード変換で使用する漢字変換テーブルのメンテナンスを行う機能を持っています。

通常は、変換設定の対応表設定 (M) のボタンをクリックして起動します。

■新規作成の場合

漢字対応表を新規に作成する場合は、つぎのウィンドウが開き、どのメーカーの日本語体系で作成するのかが選択します。



選択肢は、つぎのとおりです。

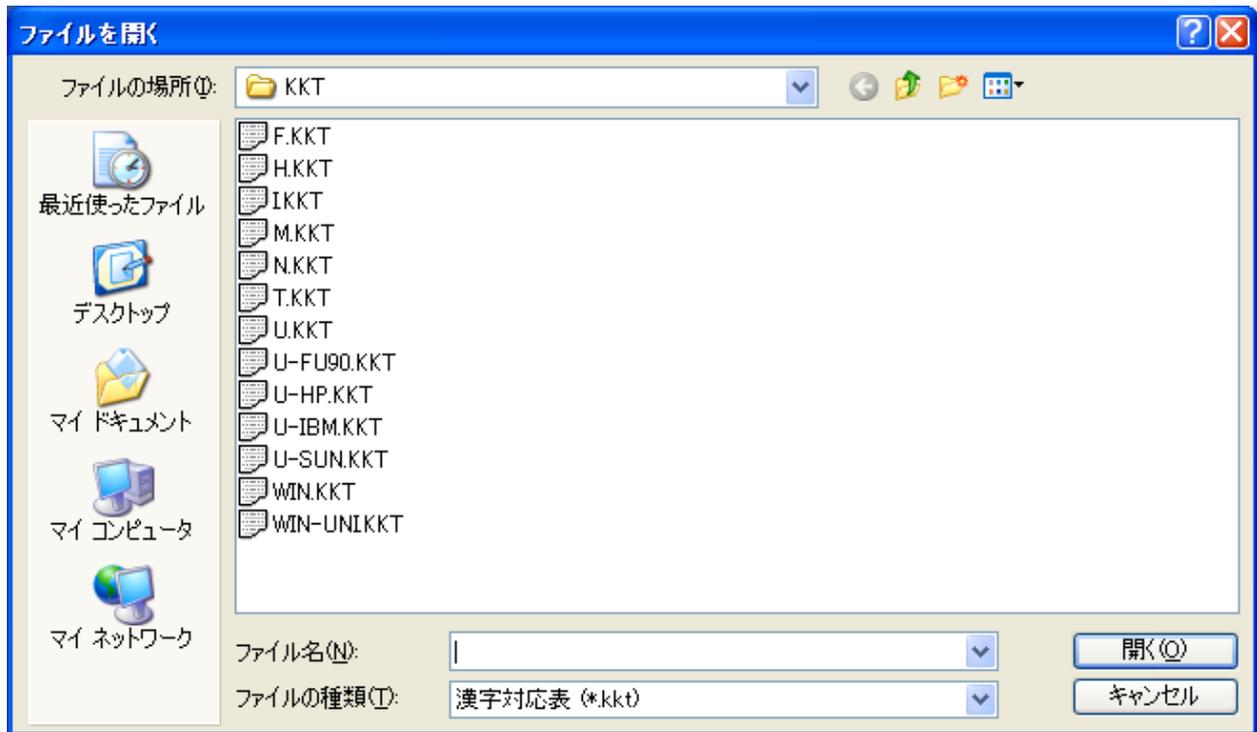
- 各社 (J I S)
- 富士通 (J E F)
- 日立 (K E I S)
- 三菱 (M E L C O M)
- 日本ユニシス (L E T S - J)
- NEC (J I P S)
- 日本 I B M (I B M)
- 東芝 (J I S)
- シフト J I S
- U n i c o d e
- E U C、G 3 なし
- E U C、G 3 あり (補助漢字)
- E U C、G 3 あり (J E F 拡張)
- D E C 漢字
- S u p e r D E C 漢字

日本語体系 (N) を選択し、OK ボタンをクリックすると、編集ウィンドウが開きます。

■既存の漢字対応表を修正する場合

F*TRAN+には、各メーカーの日本語体系に沿った漢字対応表が標準提供されています。通常は、提供されている漢字対応表をベースに変更して使用します。

変換設定で漢字対応表を選択して対応表設定 (S) のボタンをクリックすると、編集ウィンドウが開き、選択された漢字対応表の内容が読み込まれます。



F*TRAN+には、つぎの13種類の漢字対応表が標準添付されています。

F. KKT	-----	富士通・JEF用
H. KKT	-----	日立・KEIS用
I. KKT	-----	日本IBM・IBM漢字用
M. KKT	-----	三菱・MELCOM漢字用
N. KKT	-----	NEC・JIPS用
T. KKT	-----	東芝・東芝漢字用
U. KKT	-----	日本ユニシス・LETS-J用
U-FU90. KKT	----	Unix・富士通U90系用
U-HP. KKT	-----	Unix・HP系用
U-IBM. KKT	-----	Unix・IBM系用
U-SUN. KKT	-----	Unix・SUN系用
WIN. KKT	-----	Windows・シフトJIS用
WIN-UNI. KKT	--	Windows・Unicode用

■編集ウィンドウ

漢字対応表の編集画面です。登録できるタイプは以下の4種類が可能です。

- ・ **文字単位登録**

Windows側1文字とホスト側1文字を1対1で関係付けます

- ・ **連続一括登録**

Windows側のある文字からN個をホスト側のある文字からN個に連続的に1対1で対応させます。文字単位で1文字ずつ登録するより手間が省けます。

Win側のコード系がUnicode系の場合のみ指定が可能です。

- ・ **Win外字域登録**

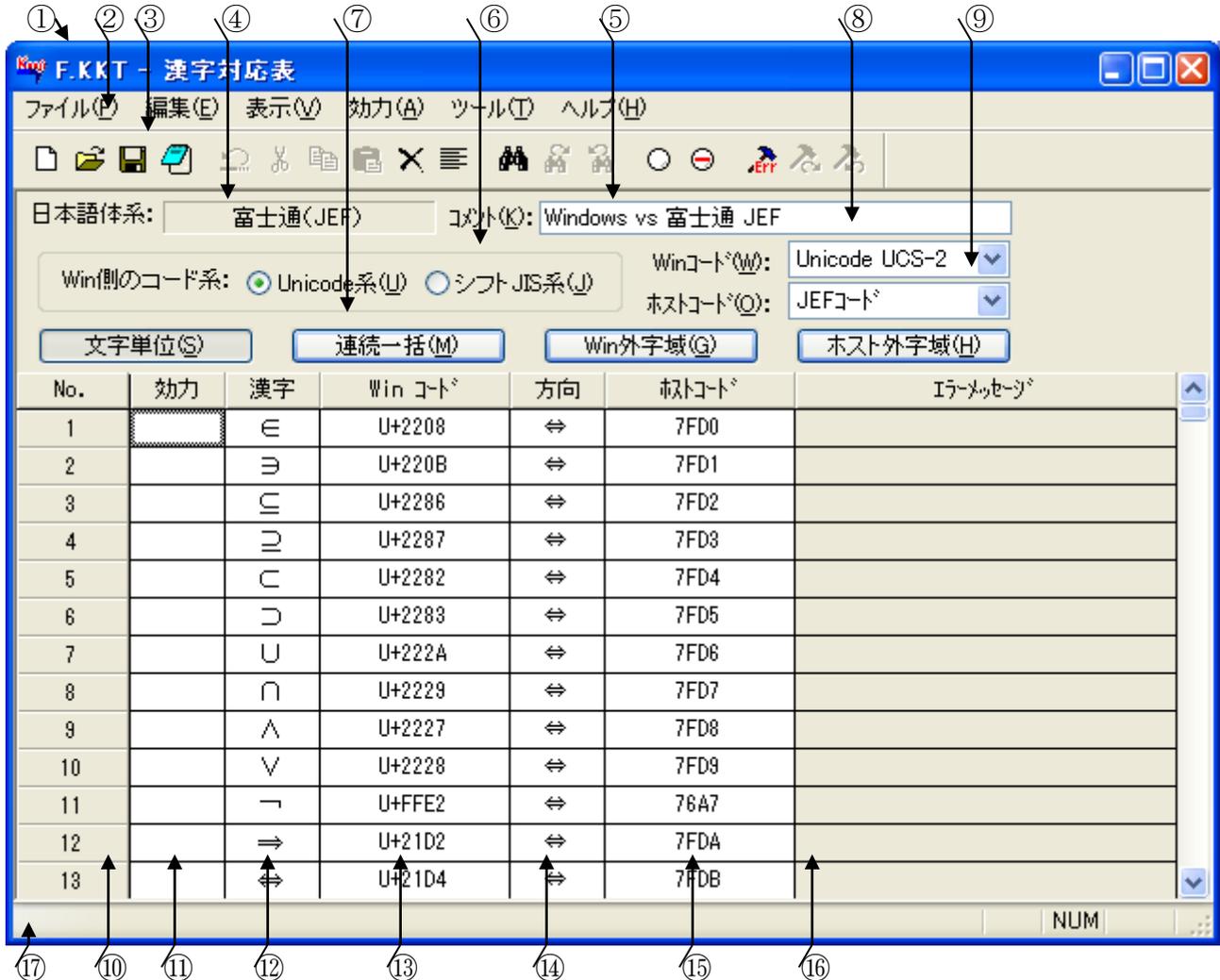
指定されたWindows側の領域を外字域として扱います。文字単位や連続一括でこの領域内文字が登録されていない場合はゲタ変換やクエスチョン変換の対象となります。

Win側のコード系がUnicode系の場合のみ指定が可能です。

- ・ **ホスト外字域登録**

指定されたホスト側の領域を外字域として扱います。文字単位や連続一括でこの領域内文字が登録されていない場合はゲタ変換やクエスチョン変換の対象となります。

<文字単位登録の場合>



①編集する漢字対応表のファイル名が表示されています。

②目的のメニューを選択して、機能を実行します。

③②の中で、よく使われる機能がボタン化されています。

④編集する漢字対応表の日本語体系が表示されています。

⑤編集する漢字対応表のコメントが表示されています。

新規に作成する場合は、漢字20文字（半角40文字）以内で設定します。変更もできます。

⑥Windows側のコード系をUnicode系かシフトJIS系か選択します。⑧で選択できるコードが変わり、⑬の表示が変更されます。

⑦登録したいタイプを指定します。連続一括、Win外字域は⑥の選択がUnicode系の場合のみ選択できます。

⑧Winコードのコード表記法を選択肢の中から選択します。

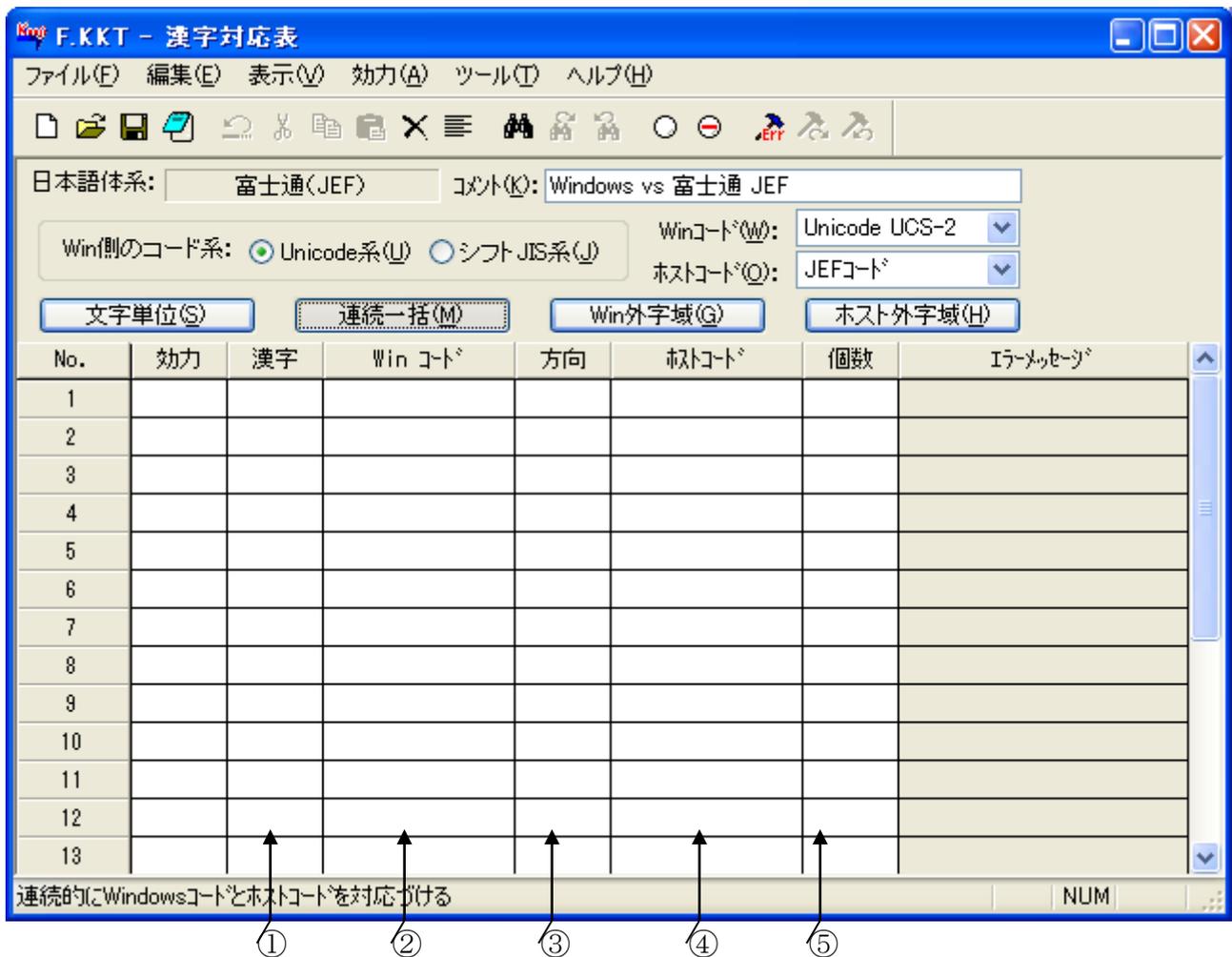
⑨Hostコードのコード表記法を選択肢の中から選択します。

⑩No. 1からNo. 12000まであります。ここをクリックして、行選択を行います。

⑪行の内容を消去せずに無効にする場合に、無効“—”の印を付ける欄です。

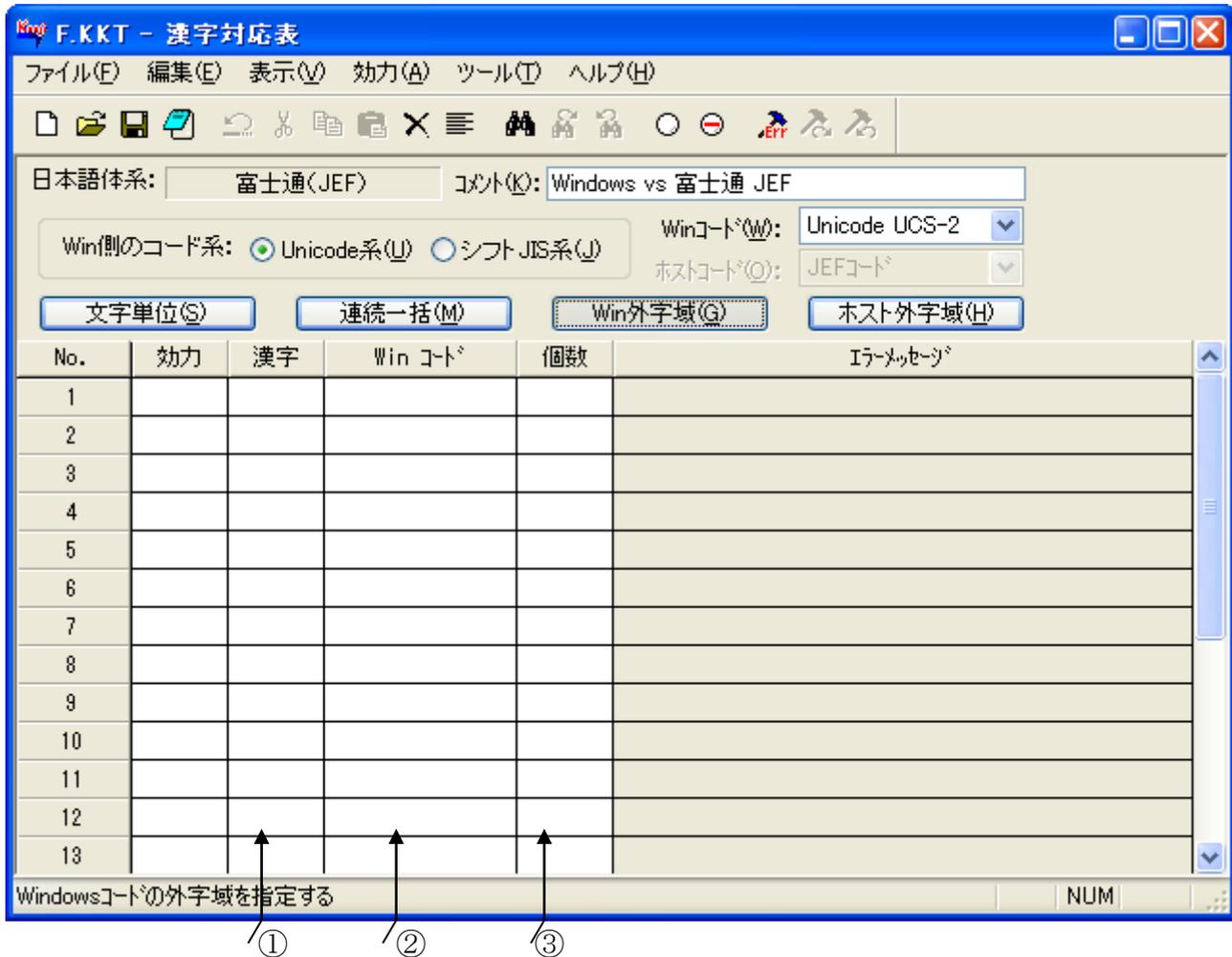
- ⑫Win側の漢字を入力します。ダブルクリックで、漢字入力サブウィンドウが開きます。
- ⑬Winコードを入力します。⑫と⑬は連動し、どちらかを入力すれば表示が変わります。
- ⑭変換する方向を3種類(⇔、→、←)の中から選択し、入力します。
- ⑮ホストコードを入力します。
- ⑯ファイルの読み／書き、“検査”を実行した時のエラーメッセージが表示されます。
- ⑰操作のガイダンスやメッセージが表示されます。

<連続一括登録の場合>



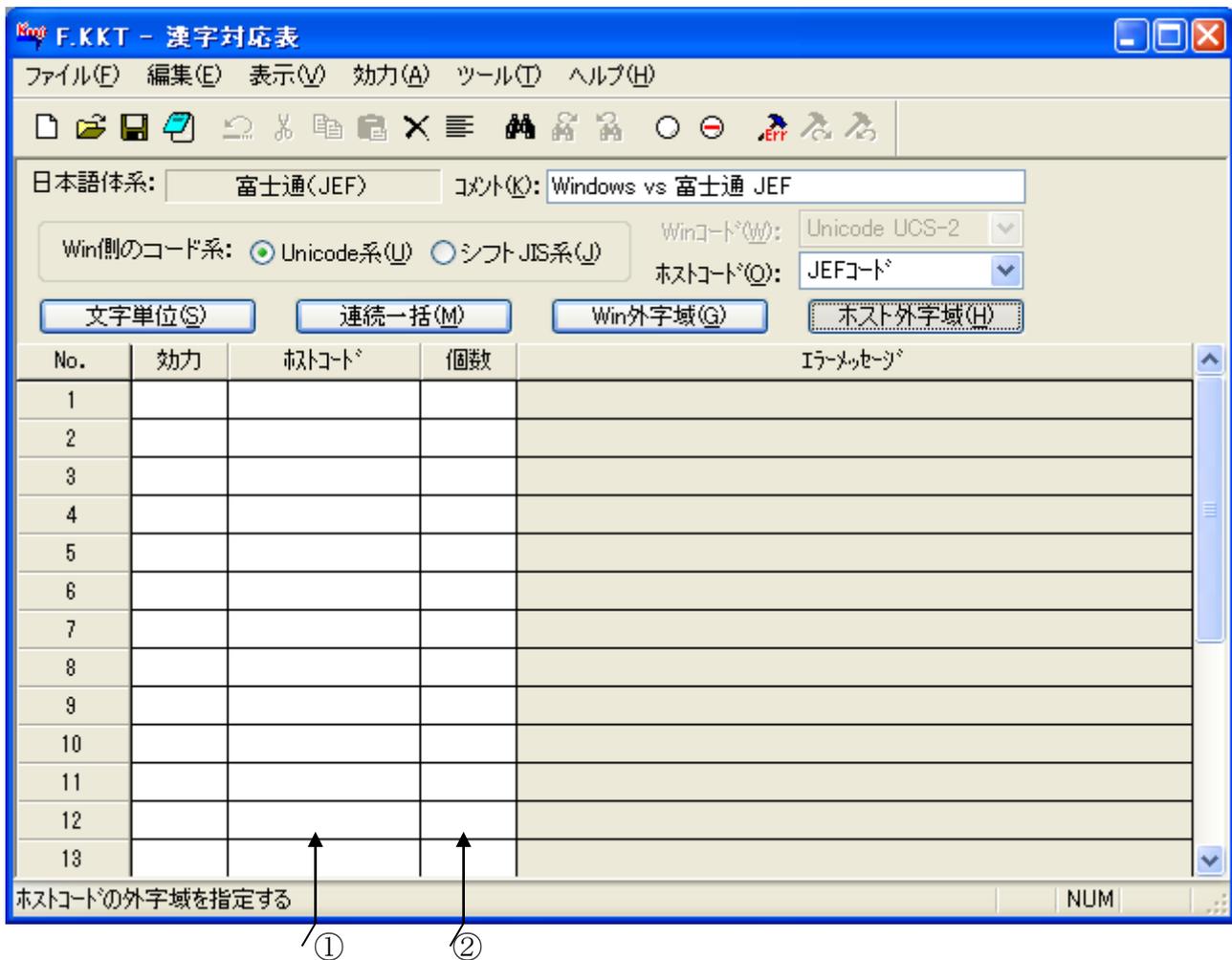
- ①Win側の開始点となる漢字を入力します。ダブルクリックで、漢字入力サブウィンドウが開きます。
- ②Win側の開始点となるWinコードを入力します。①と②は連動し、どちらかを入力すれば表示が変わります。
- ③変換する方向を3種類(⇔、→、←)の中から選択し、入力します。
- ④ホスト側の開始点となるホストコードを入力します。
- ⑤関連付ける個数を1～65535の範囲で指定します。

<Win外字域登録の場合>



- ①Win側の外字扱いする領域の開始点となる漢字を入力します。ダブルクリックで、漢字入力サブウィンドウが開きます。
- ②Win側の外字扱いする領域の開始点となるWinコードを入力します。①と②は連動し、どちらかを入力すれば表示が変わります。
- ③関連付ける個数を1～65535の範囲で指定します。

<ホスト外字域登録の場合>



- ①ホスト側の外字扱いする領域の開始点となるホストコードを入力します。
 ③関連付ける個数を1～65535の範囲で指定します。

●コメント (K)

漢字対応表のコメント欄です。通常は、漢字対応表の内容を意味するタイトルを入力します。漢字20文字（半角40文字）以内で設定します。

●Win側のコード系

Win側のコード系を以下の2種類から選択します。

Unicode系

シフトJIS系

連続一括登録、Win外字域登録を行う場合はUnicode系を選択して下さい。

●Winコード (W)

Winコード (Windowsの漢字コード) の表示・入力をどのコード表記法に従って行うかを指定します。Win側のコード系によりつぎの中から選択します。

・Win側のコード系がUnicode系の場合

拡張JISコード (基準値)

拡張句点コード

シフトJISコード

・Win側のコード系がシフトJIS系の場合

Unicode (UCS-2) (基準値)

Unicode (UTF-8)

●ホストコード (H)

選択されている日本語体系ごとにホストコードの表記法が異なり、つぎの中から選択します。

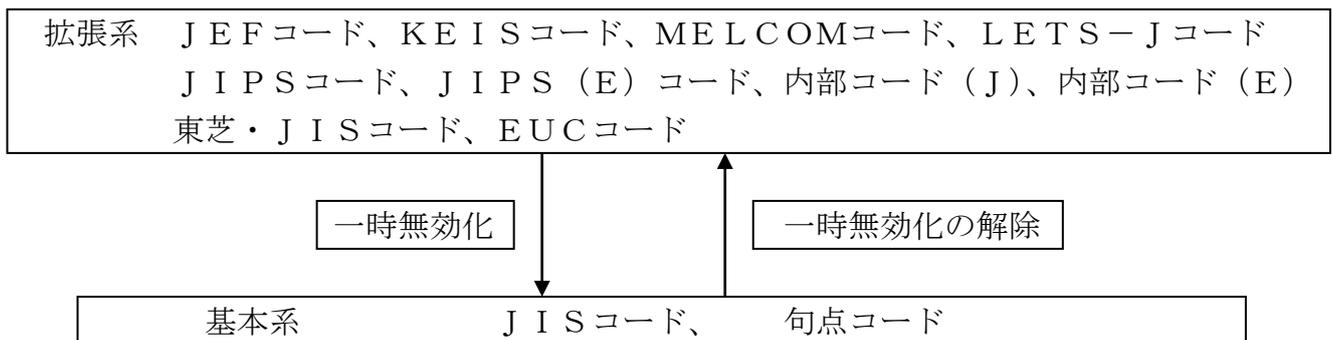
日本語体系	ホストコード	基準値	系列
各社 (J I S)	J I Sコード 句点コード	○	基本 基本
富士通 (J E F)	J I Sコード 句点コード J E Fコード	○	基本 基本 拡張
日立 (K E I S)	J I Sコード 句点コード K E I Sコード	○	基本 基本 拡張
三菱 (M E L C O M)	J I Sコード 句点コード M E L C O Mコード	○	基本 基本 拡張
日本ユニシス (L E T S - J)	J I Sコード 句点コード L E T S - Jコード	○	基本 基本 拡張
NEC (J I P S)	J I Sコード 句点コード J I P Sコード J I P S (E)コード 内部コード (J) 内部コード (E)	○	基本 基本 拡張 拡張 拡張 拡張
日本 I B M (I B M)	I B M漢字コード	○	基本
東芝 (J I S)	J I Sコード 句点コード	○	拡張 基本
シフト J I S	拡張 J I Sコード 拡張句点コード シフト J I Sコード	○	拡張 拡張 拡張
U n i c o d e	Unicode(UCS-2) Unicode(UTF-8)	○	拡張 拡張
EUC、G3なし EUC、G3あり (補助漢字) EUC、G3あり (J E F拡張) DEC漢字 S u p e r DEC漢字	J I Sコード 句点コード EUCコード	○	基本 基本 拡張

立ち上げ時／ファイル読み込み時には、常に基準値になります。

系列が“拡張”の場合のみ、拡張漢字領域のコード入力ができます。

Win側のコード系をUnicode系からシフトJIS系に切り替えると、UnicodeにはあるがシフトJISには無い文字が表現できなくなり、そのような入力データの一時無効化が起こります。効力欄が“(－)”になり、Winコード欄が“****”になります。逆に、シフトJIS系からUnicode系に変更すると、一時無効になっていた入力データが解除されて通常の表示に戻ります。

ホストコードは基本系と拡張系に大別できます。ホストコードの選択を拡張系から基本系に変更すると、拡張コードが表現できなくなり、拡張コードを持つ入力データの一时无効化が起こります。効力欄が“(－)”になり、ホストコード欄が“****”になります。逆に、基本系から拡張系に変更すると、一時無効になっていた入力データが解除されて通常の表示に戻ります。実際には、つぎの図のようになります。



例) JEFコード

効力	漢字	Winコード	方向	ホストコード
	€	223A	↔	7FD0

JISコード

効力	漢字	Winコード	方向	ホストコード
(－)	€	223A	↔	****

●入力基本パターン

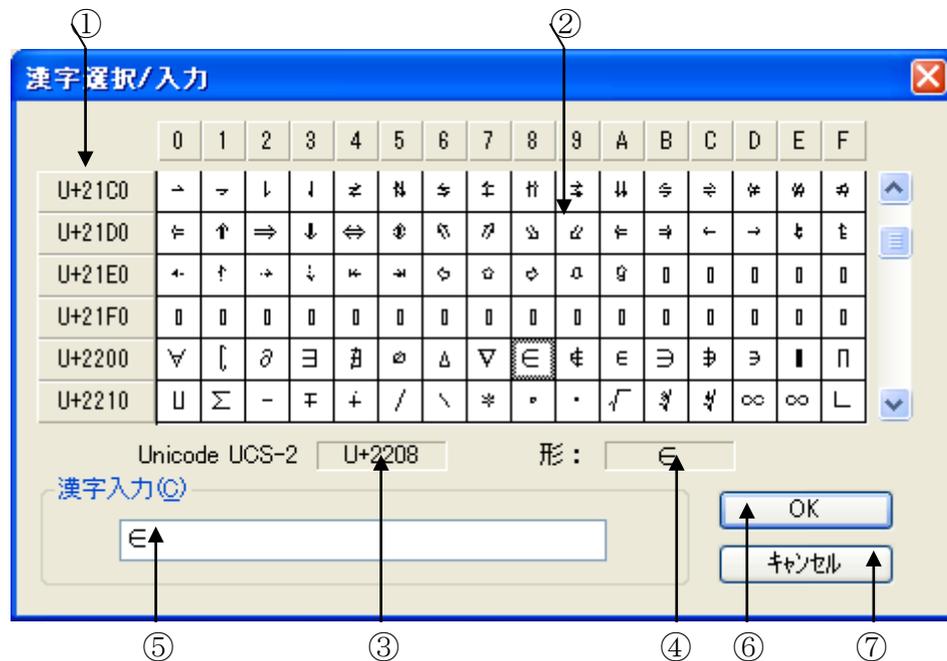
漢字対応表に定義される基本パターンはつぎのとおりです。

1. Windows 漢字 (またはコード) \longleftrightarrow ホストの漢字コード
(双方向)
2. Windows 漢字 (またはコード) \longrightarrow ホストの漢字コード
(片方向)
3. Windows 漢字 (またはコード) \longleftarrow ホストの漢字コード
(片方向)

Windows 漢字またはWindowsの漢字コード、変換する方向、ホストの漢字コードの3要素を入力します。連続一括登録、Win外字域登録、ホスト外字域登録では個数の指定が可能です。Windowsの漢字コードおよびホストの漢字コードは、正しいコードの入力のみ受け付けます。句点コードを入力する場合は、12-34のように“-”で区切って入力します。EUCのG3コードを入力する場合は、G3:A1A1のように入力します。Unicode (UCS-2)を入力する場合は、U+2208のように入力します。Unicode (UTF-8)を入力する場合は、C2-A8、またはE2-97-AFのように入力します。

●漢字入力サブウインドウ

漢字欄をダブルクリックすると、つぎのウインドウが開きます。



- ①選択されているWinコードの表記法（拡張JISコード、拡張句点コード、シフトJISコード、UCS-2、UTF-8）に従って、コードが表示されます。
- ②漢字の一覧が表示されます。入力する場合は、漢字をダブルクリックします。
- ③②で選択されている漢字のコードが表示されます。
- ④②で選択されている漢字が表示されます。
- ⑤このエディットボックスの中で文字列を入力し、目的の漢字を左端に置くことで、選択することもできます。
- ⑥漢字選択/入力後に、OKボタンをクリックして確定させます。
- ⑦キャンセルボタンをクリックすると、漢字入力サブウインドウが閉じます。

●ファイル (F) メニュー

ファイルメニューの各機能について説明します。

- 新規作成 (N)** ----- “新規に作成しますか?” と確認ウインドウが開きます。
はい (Y) ボタンをクリックし、日本語体系を選択して、
OKボタンをクリックすると、
新規に漢字対応表を作成する状態になります。
- 開く (O)** ----- “ファイルを開く” のウインドウが開きます。
ファイル名を選択し、開く (O) ボタンをクリックすると、
選択された漢字対応表が読み込まれます。
- 上書き保存 (S)** ----- タイトルバーに表示されている漢字対応表ファイルに、
現在の内容を保存します。
保存する前に内容を検査し、エラーがあれば、
“エラーデータがありますが、このまま保存を続行しますか?”
と確認ウインドウで聞いてきます。
- 名前を付けて保存 (A)** -- 実行すると、内容を検査します。エラーがあれば、
“エラーデータがありますが、このまま保存を続行しますか?”
と確認ウインドウで聞いてきます。
“名前を付けて保存” のウインドウが開いたら、
ファイル名を入力し、
保存 (S) ボタンをクリックして保存します。
- テキストファイル出力 (T)** ----- “テキスト出力” 画面が表示されます。エンコーディングを選択
して保存ボタンをクリックすると“テキストファイル出力” 画面
が開きます。ファイル名を入力し、保存 (S) ボタンをクリック
すると、設定内容をテキスト形式ファイルへ出力します。

<テキストファイル出力例>

コメント	:	Windows vs 富士通	JEF			
Winコード	:	拡張	JIS	コード		
日本語体系	:	富士通	(JEF)			
ホストコード	:	JEF	コード			
No.	効力	漢字	Winコード	方向	ホストコード	エラーメッセージ
1		㊦	223A	↔	7FD0	

- 漢字対応表の終了 (X)** -- 漢字対応表エディタを終了します。

●編集 (E) メニュー

編集メニューの各機能について説明します。

元に戻す (U) ---- 直前の編集作業を無効にし、元に戻します。

切り取り (T) ---- 選択されている行の内容が切り取られ、カットバッファに入ります。

コピー (C) ----- 選択されている行の内容がカットバッファに入ります。

貼り付け (P) ---- カットバッファの内容を、セルポインタがある行へ貼り付けます。
貼り付けた行以降に既存の設定項目があれば、挿入になります。

削除 (D) ----- 選択されている項目の内容が削除されます。

すべて選択 (L) -- すべての設定項目欄が選択された状態になります。

一行下げる (I) -- セルポインタがある行以降が1行ずつ下がります。
通常は、行挿入をする場合に使用します。

整列 (S) ----- Winコード、ホストコードを基準として、
設定値の並べ替えができます。
行選択をしてから実行すると、その範囲が対象になります。
つぎの4つのパターンがあります。

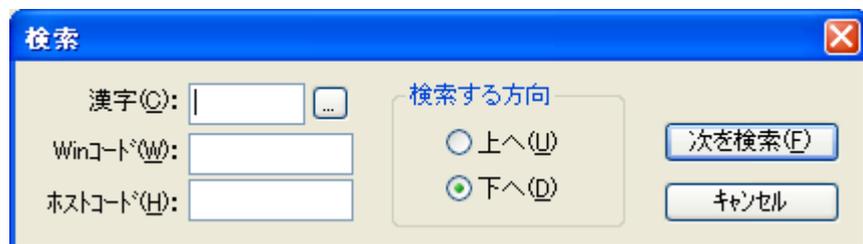
→Winコード順 (W) →昇順 (U)

→Winコード順 (W) →降順 (D)

→ホストコード順 (H) →昇順 (U)

→ホストコード順 (H) →降順 (D)

検索 (F) ----- 漢字やコードで設定項目を検索することができます。
漢字 (または、コード) と、検索する方向を指定して実行します。
漢字 (Winコード)、ホストコードの両方が指定された場合は、
どちらかが一致する項目を検索します。
漢字 (C) の入力欄の横にあるボタンをクリックすると、
漢字入力サブウィンドウが開きます。



ツールバーには、上方向検索ボタン/下方向検索ボタンがあり、
指定値で再検索することができます。

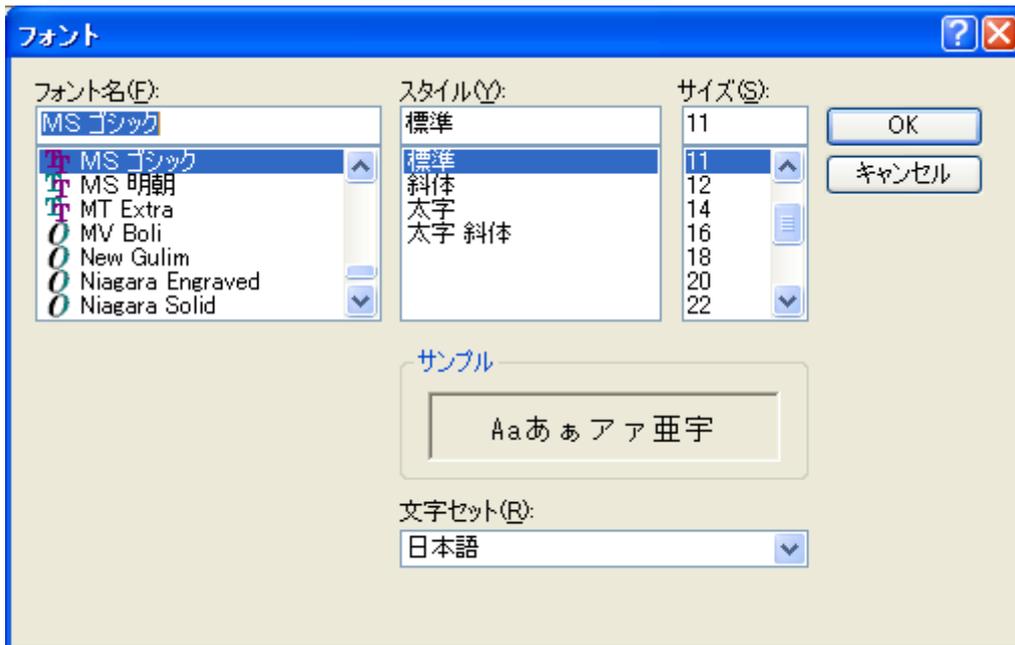
●表示 (V) メニュー

表示メニューの各機能について説明します。

ツールバー (T) ----- ツールバー表示のON/OFFができます。

ステータスバー (S) -- ステータスバー表示のON/OFFができます。

漢字部のフォント (F) ---- 漢字欄の表示フォントを、つぎのウインドウで設定できます。



●効力 (A) メニュー

効力メニューの各機能について説明します。

有効化 (E) -- 選択されている行の内容を有効にし、効力欄の印“—”を消します。
ただし、一時無効の印“(—)”は消えません。

無効化 (D) -- 選択されている行の内容を消去せずに無効にし、
効力欄に印“—”を表示します。

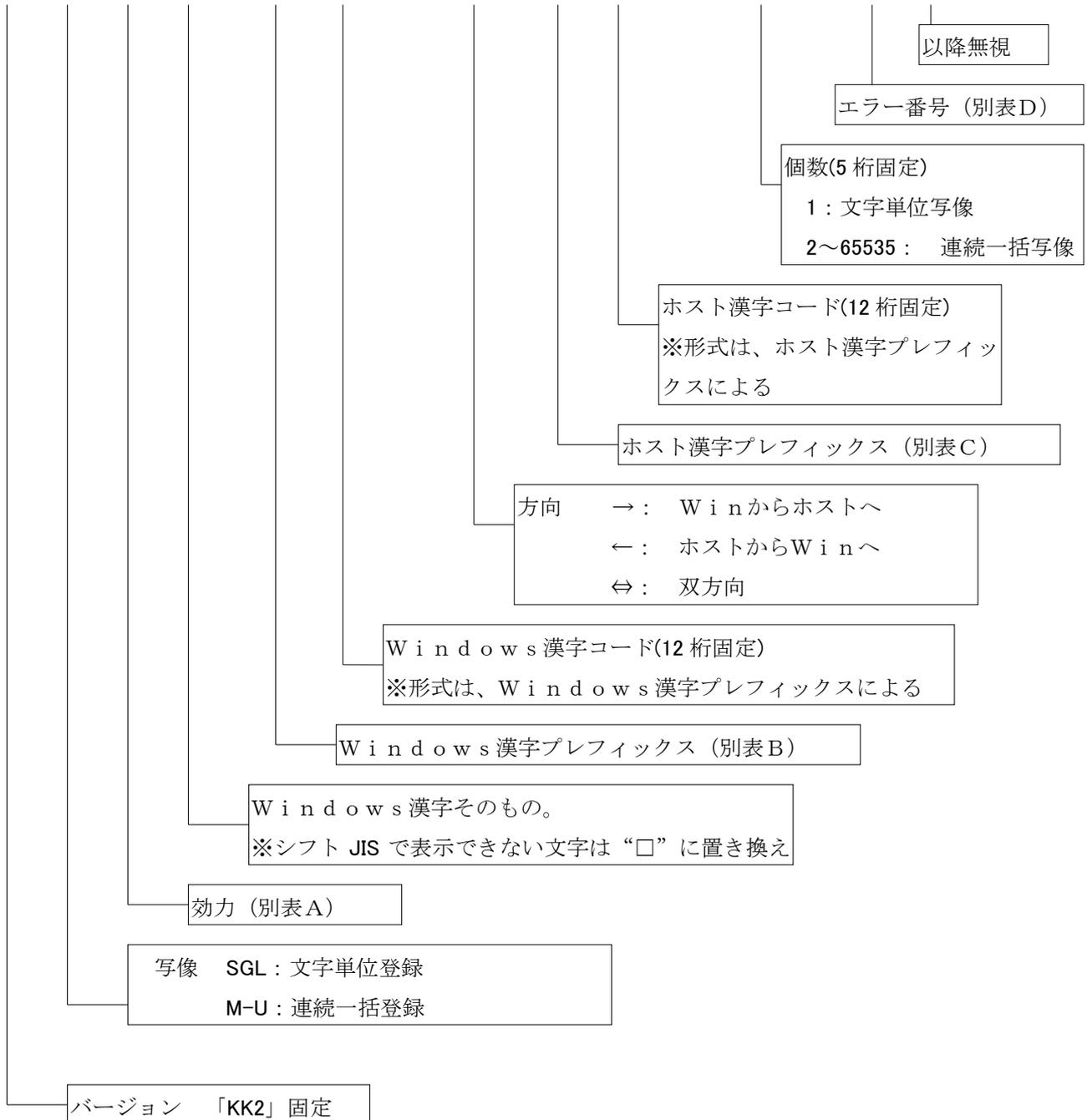
● ツール (T) メニュー

ツールメニューの各機能について説明します。

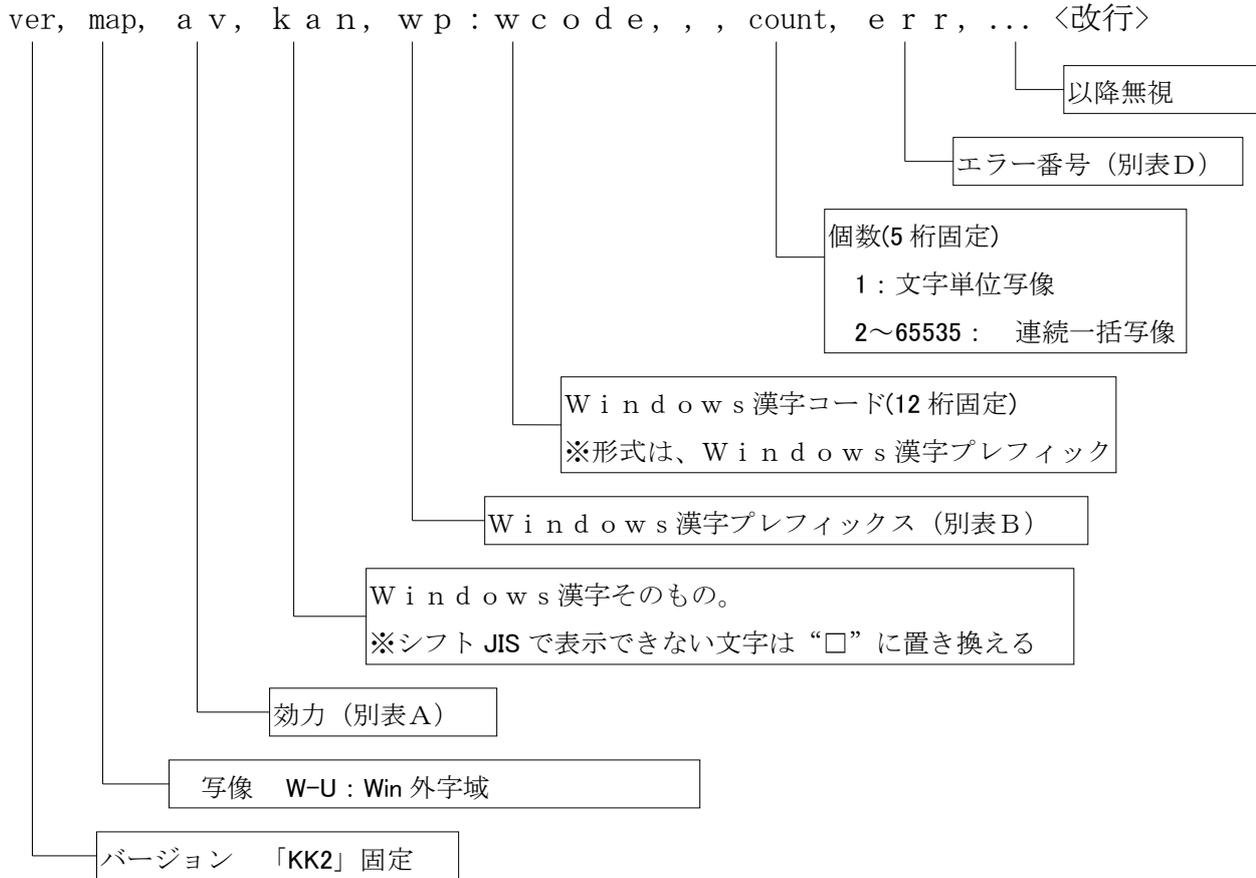
- 検査 (R)** ----- 設定されている内容に誤りがあるかどうかを検査します。
 行選択をしてから実行すると、その範囲が対象になります。
 エラーの種類はつぎのとおりです。
- 不完全** ---- W i nコード、方向、ホストコードの3要素が
 そろわない
 - 迷子** ----- W i nコード→ホストコード→別のW i nコード
 のように、コード変換が収束しない
 - 1対N** ---- W i nコード→ホストコード、
 W i nコード→別のホストコードのように、
 1つのコードが2つ以上のコードに設定されている
 - 2重定義** -- 同一の内容が重複して設定されている
 - 領域重複** -- 連続一括、外字域登録の場合で、登録されている領域が
 重なって指定されている
- 前のエラー (P)** -- 現在のセルポインタから1つ前のエラーがある行へ
 セルポインタを移動します。
- 次のエラー (N)** -- 現在のセルポインタから次のエラーがある行へ
 セルポインタを移動します。
- CSV形式 (C)** -- 漢字対応表をCSV形式で活用するために使用します。
 CSV形式ファイルの読み込み (L) / 書き込み (S) を行います。
 CSV形式ファイルのフォーマットはつぎのとおりです。

<文字単位登録と連続一括登録のテキストフォーマット>

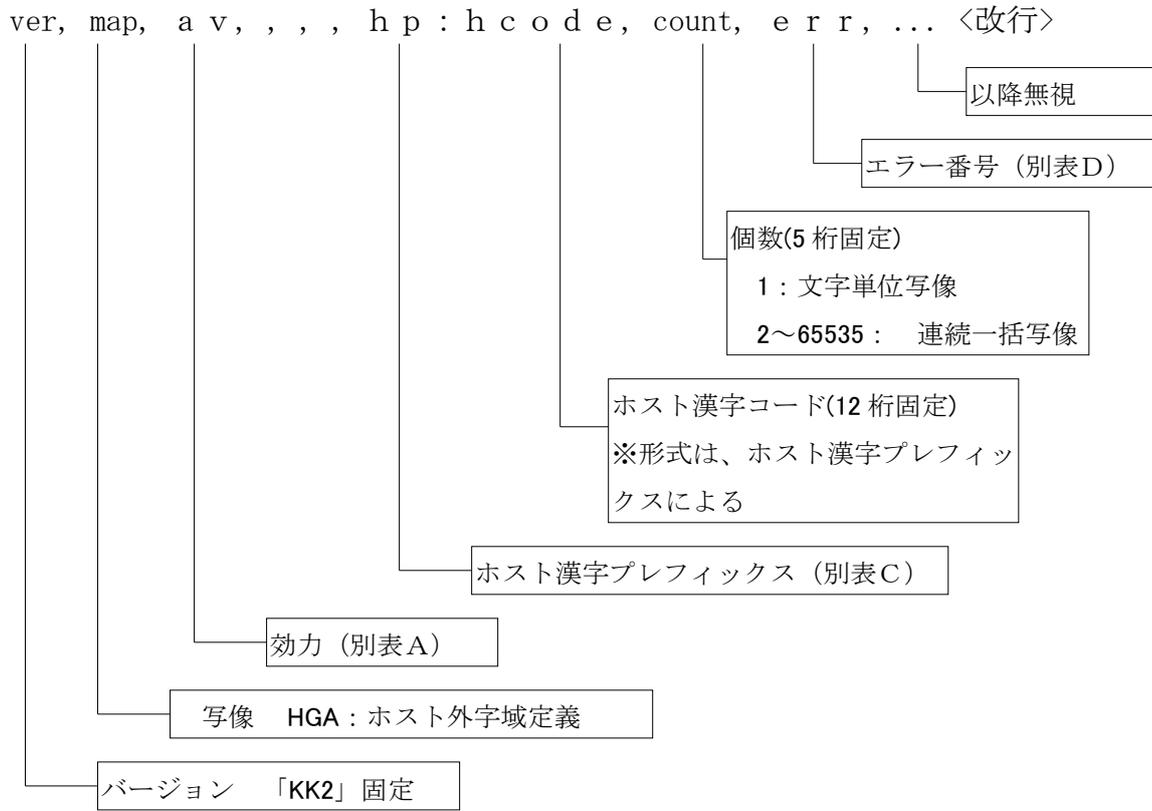
ver, map, a v, k a n, w p : w c o d e, d i r, h p : h c o d e, c o u n t, e r r, ... <改行>



<Win外字域登録のテキストフォーマット>



<ホスト外字域登録のテキストフォーマット>



[別表A：効力表記]

- A：有効
- BH：一時無効（ホストコード）
- C：無効
- BW：一時無効（Winコード）
- B2：一時無効（Winコード、ホストコード）

[別表B：Windows漢字コード表記]

- JIS：（拡張）JISコード
- KTC：（拡張）句点コード
- SJC：シフトJISコード
- UC2：Unicode（UCS-2）
- UT8：Unicode（UTF-8）

[別表C：ホスト漢字コード表記]

- JIS：JISコード
- KTC：句点コード
- SJC：シフトJISコード
- UC2：Unicode（UCS-2）
- UT8：Unicode（UTF-8）
- EUC：EUCコード
- EG3：EUC（G3あり）コード
- DEC：DEC漢字コード
- SDC：Super DEC漢字コード
- FJF：富士通JEFコード
- HKS：日立KEISコード
- MJS：三菱MELCOMコード
- ULJ：日本ユニシスLETS-Jコード
- NJJ：NEC JIPSコード
- NJE：NEC JIPS（E）コード
- NIJ：NEC内部コード（J）
- NIE：NEC内部コード（E）
- TOS：東芝漢字コード
- IBM：IBM漢字コード

[別表D：エラーコード番号]

- 0：正常
- 1：不完全
- 2：迷子
- 4：1対N
- 8：2重定義
- 10：領域重複

2つ以上のエラーがある場合は、組み合わせになります。

たとえば、エラー番号=12は、1対N／2重定義エラーになります。

●ヘルプ（H）メニュー

ヘルプメニューの各機能について説明します。

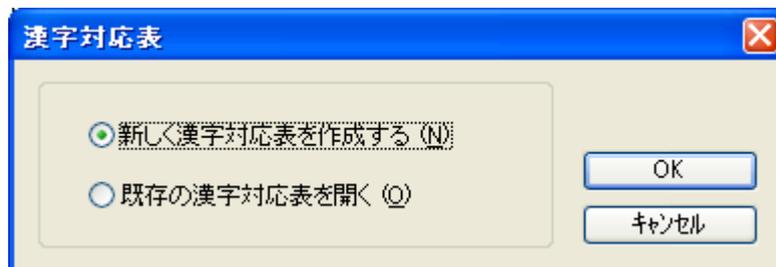
バージョン情報（A） ← バージョン情報のウインドウが開きます。

漢字対応表エディタのバージョン情報を知ることができます。



■漢字対応表エディタの単独起動

漢字対応表エディタをF*TRAN+からではなく、エクスプローラ等から単独で起動することもできます。起動するとつぎのウインドウが開きますので、新規または既存の漢字対応表の選択から始めます。既存の漢字対応表を選択した場合は、ファイル選択を行います。以下の手順は、いままで説明したとおりです。



また、エクスプローラ等から漢字対応表ファイル（拡張子が“. KKT”のファイル）をダブルクリックし、直接、漢字対応表エディタを立ち上げ、メンテナンスすることもできます。

4.15 ログの設定と表示

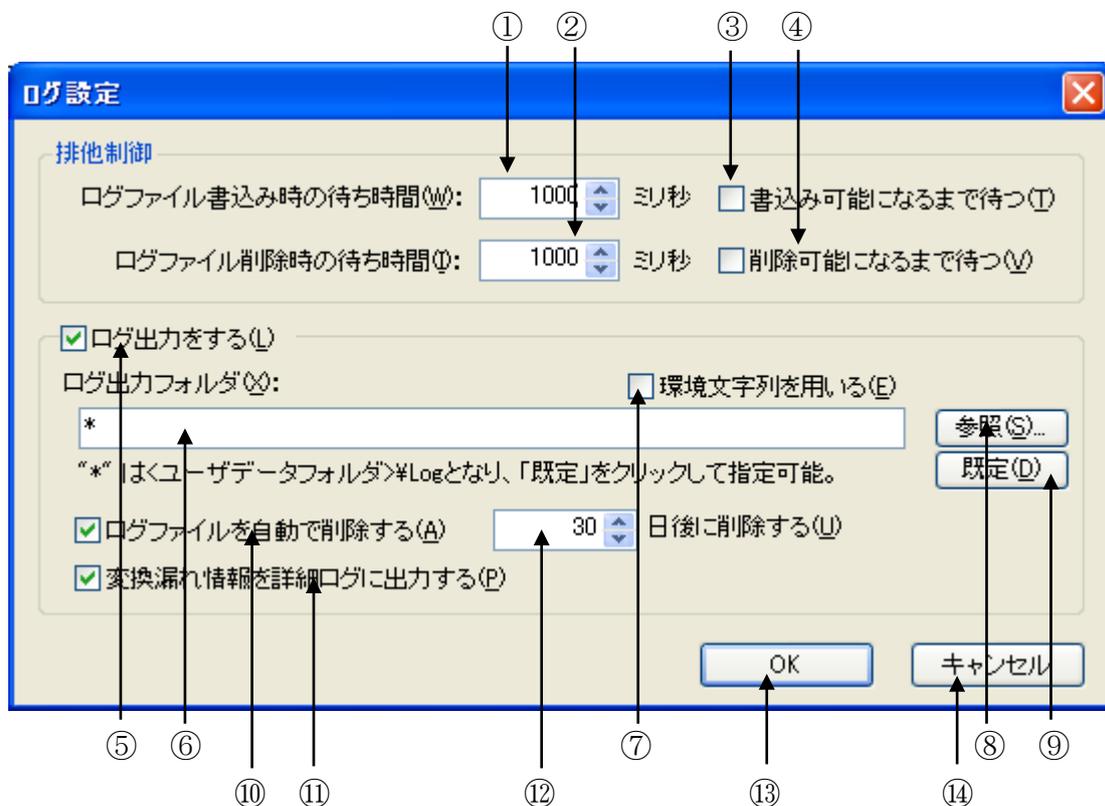
F*TRAN+は、実行や変換のログファイルを出力する機能を持っています。ここではログ出力の設定と、変換時ログ出力について説明します。

◆参考 --- ログに関しては

詳細は「第5章 ログ」を参照してください。

■ログ設定

メニューバーの設定 (S) → ログ設定 (L) を実行すると、「ログ設定」のウインドウが開きます。



- ①ログファイル書き込みの排他制御をおこなう待ち時間を0～10000（ミリ秒）で指定します。
- ②ログファイル自動削除の排他制御をおこなう待ち時間を0～10000（ミリ秒）で指定します。
- ③ログファイル書き込みの排他制御時、書き込み可能になるまで待ちます。①の指定は無効になります。
- ④ログファイル自動削除の排他制御時、削除可能になるまで待ちます。②の指定は無効になります。
- ⑤ログ出力をおこなうかどうかの指定をします。ログ出力をしないと変換速度が向上します。⑥～⑫の設定は無効になります。
- ⑥ログ出力フォルダの指定をします。フルパスでの指定か環境文字列を用いた指定が可能です。
- ⑦ログ出力フォルダの指定に環境文字列を使う場合、チェックをします。チェックを外すとその時点でのフルパスに変更されます。環境文字列にできないフルパスが指定されるとチェックすることができません。
- ⑧ログ出力フォルダの選択をするフォルダ選択画面が表示されます。
- ⑨ログ出力フォルダに“*”（半角アスタリスク）を設定します。ユーザデータフォルダ内のLogフォルダを設定したことになります。
- ⑩ログファイルを自動的に削除する機能を有効にします。
- ⑪変換漏れ情報を詳細ログに出力するかの指定をします。出力しない設定をした場合、変換漏れが多く発生する変換での変換速度が向上します。
- ⑫ログファイルの自動削除をおこなう場合、ログを保持する日数を1～1000（日）で指定します。
- ⑬OKボタンをクリックすると、修正した内容がすぐに有効になります。
- ⑭キャンセルボタンをクリックすると、修正したログ設定を無効にし、ログ設定のウインドウを閉じます。

●排他制御

ログファイルの出力や削除する際、F*TRAN+が多重起動されると複数のプログラムから同時にログファイルに追記または削除などが発生することがあります。この場合、排他制御をおこないログファイルに正しく記録したり、削除をおこないます。

排他制御はあるプログラムがファイルアクセスしている時、次に同じファイルに書き込みなどの処理をしようとするプログラムが、前のアクセスが完了するのを待って処理することで実現しています。

ログ設定ではログファイル書き込み時の排他制御待ち時間と自動削除時の待ち時間をミリ秒単位で指定可能です。また、それぞれ時間指定なしで処理を実行機能になるまで待ち続ける指定もできます。

ログファイル書き込み時の排他制御でタイムアウトになった場合、次の通番の名前を持ったログファイルが作成され、以後そのファイルに追記されていきます。また、ログファイルの自動削除時の排他制御でタイムアウトになった場合はそのままそのファイルは残され、翌日削除を試みます。

●ログ出力フォルダ

ログファイルを出力する既存のフォルダを指定します。フォルダの指定はフルパス名で指定しますが、環境文字列を利用することができます。また、“*”（半角アスタリスク）をするとユーザーデータフォルダ内の Log フォルダを指定したことになります。「既定」ボタンをクリックしても入力が可能です。

フォルダの指定に環境文字列を利用すると実行環境に依存しない指定が可能です。フルパスの指定をすることで他の環境で実行すると正常に動作しなくなる場合がありますが、それを防ぐことができます。

環境文字列には以下のような文字列が利用可能です。

環境文字列	フォルダ
@USERPROFILE@	カレントユーザのプロファイルフォルダ 例：C:\Users\ユーザー名
@ALLUSERSPROFILE@	All Users のプロファイルフォルダ 例：C:\ProgramData
@FT_Personal@	カレントユーザのドキュメント（以下と同じ） 例：@USERPROFILE%\Documents
@FT_Userdata@	F*TRAN+のユーザーデータフォルダ
@SystemDrive@	システムドライブレター
@環境変数名@	環境変数名を指定することが可能。OS の環境変数が利用可能です。

例：

@USERPROFILE%\FT_LOG

カレントユーザのプロファイルフォルダ配下の FT_LOG フォルダを指定

上記ウインドウの「環境文字列を用いる」のチェックボックスは入力されたフルパスを環境文字列を使ったパスに変更します。また、チェックされている場合、チェックを外すことでフルパス表記に変更することができます。

◆参考 ---- <ユーザーデータフォルダ>とは

<ユーザーデータフォルダ>は、ログオンユーザの「ドキュメント」にある“FTRANP”フォルダがデフォルトで指定されています。

●ログファイルの自動削除

ログファイルは一日単位に2種類のファイルがログ出力フォルダに作成されていきます。多量の変換をおこなっていくとログファイルのサイズは次第に大きくなり、そのまま放っておくとディスクの空き容量が減っていきOSの動作に支障をきたすことがあります。しかしF*TRAN+では、デフォルトでログファイルの削除をおこなうことが可能で、標準で30日間は保持されます。

ログ設定ではこの保持期間を最大1000日まで伸ばすことが可能です。また、利用者が独自にログファイルを管理し削除を行う場合は、ログの削除をしない設定が可能です。

ログファイルは一日に一度、その日最初に起動するタイミングでおこなわれます。

●変換漏れ情報を詳細ログに出力する

拡張漢字で変換できなかった変換漏れ漢字の情報は、詳細ログに最大512文字までが出力されます。変換漏れ情報が必要ない場合はこの設定をオフにすることができます。出力しないことで変換速度の向上がはかれます。

現在、詳細ログファイルには変換漏れ情報のみが記録されています。したがってこの設定をオフにすると詳細ログは出力されませんが、将来その他の情報が詳細ログに出力される可能性があります。

■変換漏れログ表示機能について

ホスト→Winファイル変換／Win→ホストファイル変換実行時に、変換できない漢字情報を以下のログファイルに出力しています。

詳細ログファイル

変換漏れログ表示機能ではこれらの変換漏れ情報をウインドウ上に表示させて見ることができます。

◆注意 ---- 画面から実行した場合のみ表示可能

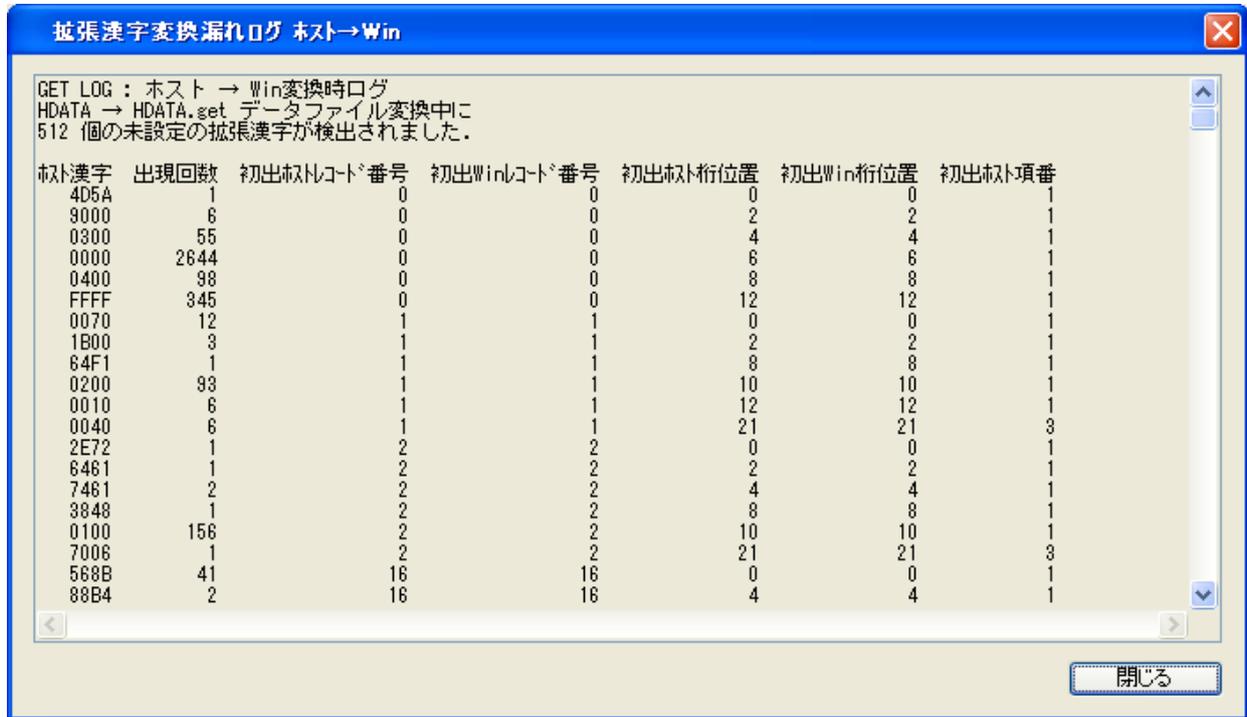
表示できる変換漏れ情報は画面上から変換を実行した場合に限り表示が可能です。バッチなど、画面以外から変換を実行した結果は表示できません。また、変換漏れ情報がない場合（変換の結果漏れが発生しなかったか、ログ設定で漏れ情報を出力しない設定の場合）は表示することはできません。

◆注意 ---- 出力される数は512種類まで

変換漏れ情報として出力される文字種は最大512種類までです。

●変換時ログ表示（ホスト→Winファイル変換）

メニューバーの変換時ログ表示（K）のホスト→Win（G）を実行すると、ホスト→Winファイル変換のログを見ることができます。通常は、ホスト→Winファイル変換を実行した直後に確認します。



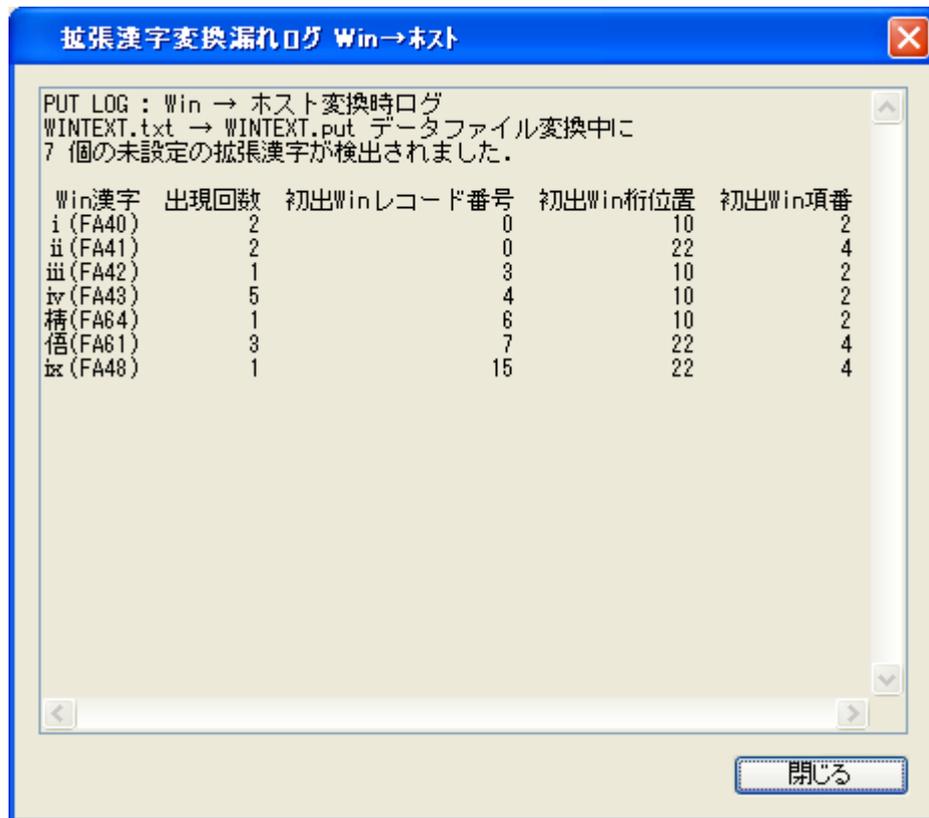
ホスト漢字	変換元ファイルの内、KKT ファイルに未設定の拡張漢字のコードを表示
出現回数	当該「ホスト漢字」が、ファイル内に何回出現したかを表示
初出ホストレコード番号	当該「ホスト漢字」が、変換元ファイル内の何番目のレコードで初めて出現したかを表示
初出Winレコード番号	当該「ホストコード」に対して「＝」変換、「?xxxx」変換」等の変換代替処理による出力が行なわれた最初のレコードを表示。大抵の場合は初出ホストレコード番号と一致するが、「フェーズ」機能を利用して変換前処理などを行なった場合は一致しない
初出ホスト桁位置	「初出ホストレコード番号」のレコードにおいて、当該「ホスト漢字」が最初に出現したバイト桁位置を表示
初出Win桁位置	「初出Winレコード番号」のレコードにおいて、「＝」変換、「?xxxx」変換」等の変換代替処理による出力がされた最初のバイト桁位置を表示
初出ホスト項番	「初出ホストレコード番号」のレコードにおいて、当該「ホスト漢字」が最初に出現した項目番号を表示

ここに出力されるホスト漢字は、変換設定／漢字変換方式の設定／ホスト拡張漢字の指定（＝変換、「?xxxx」変換）の対象になった漢字コードです。

ログの中に「＝」等の変換対象にしたくないホスト漢字があった場合は、ログに出力されたホスト漢字コードを漢字対応表に登録してから、再度、変換をしてください。

●変換時ログ表示 (Win→ホストファイル変換)

メニューバーの変換時ログ表示 (U) のWin→ホスト (P) を実行すると、Win→ホストファイル変換のログを見ることができます。通常は、Win→ホストファイル変換を実行した直後に確認します。



Win漢字	変換元ファイルの内、KKTファイルに未設定の拡張漢字のコードを表示
出現回数	当該「Win漢字」が、ファイル内に何回出現したかを表示
初出Winレコード番号	当該「Win漢字」が、変換元ファイル内の何番目のレコードで初めて出現したかを表示
初出Win桁位置	「初出Winレコード番号」のレコードにおいて、当該「Win漢字」が最初に出現したバイト桁位置を表示
初出Win項番	「初出Winレコード番号」のレコードにおいて、当該「Win漢字」が最初に出現した項目番号を表示

ここに出力されるWin漢字は、変換設定／漢字変換方式の設定／Win拡張漢字の指定(“=” 変換固定)の対象になった漢字コードです。

ログの中に“=”等の変換対象にたくないWin漢字があった場合は、ログに出力されたWin漢字コードを漢字対応表に登録してから、再度、変換をしてください。

4.16 ヘルプ

ヘルプ・メニューには、オンラインマニュアルの表示、バージョン情報の2つの機能があります。

●オンラインマニュアルの表示

メニューバーのヘルプ (H) → マニュアル (導入編/コマンド編/解説編/マルチレコード編/プログラム応用編) を選択すると、オンラインマニュアルが表示されます。

オンラインマニュアルはすべて、「Acrobat Reader」などのPDFビューワで表示してください。

●バージョン情報

メニューバーのヘルプ (H) → バージョン情報 (A) を実行すると、バージョン情報のウインドウが開きます。

現在、立ち上がっているF*TRAN+のバージョン情報を知ることができます。



ここには、サポートサービスを受ける場合に必要となる情報が入っています。

第5章



ログ

5.1 ログについて

F*TRAN+は変換実行の際、変換結果などをログファイルとして出力することができます。

■ログファイル

デフォルトで出力されるログファイルの他に起動オプションやスクリプトファイルで指定することにより変換のたびに出力される**個別ログ**があります。個別ログに関してはコマンド編の「2.1 FP.EXE 起動と終了」にある起動オプションの /Log の説明およびこの解説編の「4.10 スクリプトファイル」をご覧ください。

5.2 ログの詳細

以下2種類のログがあります。

名称	ファイル名	フォーマット	出力先	記載情報
基本ログ ファイル	ftBasic_YYYYMMDD_nnnnnn.log	UTF-8 テキスト CSV 形式	〈ユーザーデータ フォルダ〉¥Log	各処理開始・終了情報、実 行エラー情報など
詳細ログ ファイル	ftDetail_YYYYMMDD_nnnnnn.log	UTF-8 テキスト CSV 形式		拡張エリア漢字の変換漏 れ情報

●基本ログファイル

実行、変換に関する履歴が記録されるファイルです。

実行の開始・終了日時、発生した実行エラーの日時・内容などを調べる事ができます。

●詳細ログファイル

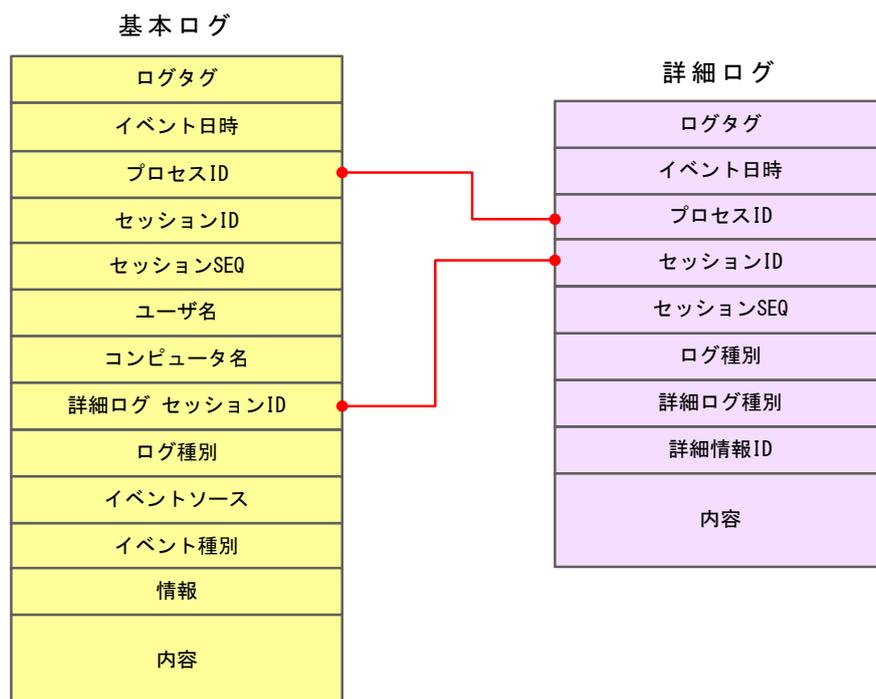
基本ログファイルに関連付けられ、基本ログ項目の詳細情報を記録しているファイルです。現在は変換実行時の変換漏れ情報が記録されます。

出力されるログファイルはテキストファイルで、**UTF-8 (固定) のCSV形式**です。UTF-8形式のファイルを表示が可能なメモ帳などのアプリケーションで表示、確認することができます。

これらのファイル内には『行』と『セッション』という2つの単位で情報が出力されています。

一連の処理をセッションとして扱い、セッション単位でログは連続して出力されています。現在、変換処理では1つのデータファイルの変換開始から終了までが1セッションとしてまとめられており、そこには複数の行が存在しています。その他変換処理以外では1セッションに1行という処理がほとんどです。

各セッションには「セッションID」というIDが降られています。基本／詳細ログファイルの個別のセッションで書き込まれています。もし基本ログに付随した詳細ログがある場合は、次ページの図のように基本ログの個別項目に詳細ログのセッションIDが記録されます。これによりこの二つのログは連結しています。



■ ログファイルの格納先

これら2種類のログはデフォルトでユーザデータフォルダ内の **Log** フォルダに格納されますが、格納先はログ設定で変更が可能です。詳細は「4. 15 ログの設定と表示」をご覧ください。なお、この設定はログオンユーザ毎の設定となります。

◆ 参考 ----- <ユーザデータフォルダ>とは

<ユーザデータフォルダ>は、ログオンユーザの「ドキュメント」にある“**FTRANP**”フォルダがデフォルトで指定されています。

■ ログファイルの自動削除

これら2種類のログはF*TRAN+が実行される度に記録されるために徐々にディスクの使用量が増え、ディスクの空き容量が減少してしまいます。この問題に対処するためログファイルを自動的に削除することができます。ログファイルの削除は最後にF*TRAN+を起動してから日付が変わった最初の起動時にのみおこなわれます。

自動削除はログ設定で変更が可能です。詳細は「4. 15 ログの設定と表示」をご覧ください。

■ ログ出力の抑制

これらログはGUIでの起動、コマンドラインの起動、およびOLEオートメーションの起動のいずれでも出力されますが、ログ出力に伴いプログラムの実行速度が低下します。速度の低下を防ぎたい場合はログ出力を抑止することができます。詳細は「4. 15 ログの設定と表示」をご覧ください。

■基本ログファイルについて

基本ログファイルは以下のようなファイル名で出力されます。

```
ftBasic_YYYYMMDD_nnnnnn.log
```

ファイル名の各部分は以下のような意味となっています。

ファイル名の各部分	種別	内容
ftBasic	ログ種別	“ftBasic” は、基本ログファイルであることを表す。
YYYYMMDD	日付	セッション開始日 例：2006年2月1日なら“20060201”となる
nnnnnn	通番	10進数で“000000”～“999999”の形式をとる。 最初の通番は“000000”となる

基本ログファイルのレコードは、1ファイルに対して追記される形で行なわれます。ログファイルは容量が1MBを超えると新たに作成されます。また、日付が変わると容量に関係なく新たに作成されます。追記しようとしたファイルが、何らかの理由で書き込み禁止であった場合も、新たに作成されます。新たに作成されたファイルは以下のような連番ファイルとなります。

```
ftBasic_20080120_000000.log
```

```
ftBasic_20080120_000001.log
```

```
ftBasic_20080120_000002.log
```

増加した通番のファイルが作成されると、以後の書き込みはそのファイルに対して行われ通番の小さいファイルには追記されません。なお、..._999999.log以降は新たに作成されず、ファイルサイズにかかわらずこのファイルに追記されます。

出力されるログファイルはCSV形式のテキストファイル（UTF-8）で、各項目は次のページのような構成になっています。

◆注意 ---- メイン画面上での操作エラーは基本的に記録されない

基本ログファイルはスクリプトファイルの実行エラーなどについては、網羅的に出力しますが、メイン画面上の操作（例えば、変換マップ設定）におけるポップアップ警告等については出力しません。

基本ログのフォーマット

項目名	内容
1 ログタグ	#FTB2 ログファイルの個別の形式、バージョンを意味します。ログの形式が異なると変更されます。
2 イベント日時	イベント発生日時 YYYY-MM-DD HH:mm:ss.MSC (MSCはm秒単位の3桁)
3 プロセスID	プロセスID文字列 13~22文字の文字列(※2を参照)
4 セッションID	基本ログのセッション単位につきプロセス内で一意なセッションID文字列(※3を参照)
5 セッションSEQ	セッション内の連番。 セッション毎に付けられる1からの連番。
6 ユーザ名	利用ユーザ名
7 コンピュータ名	利用コンピュータ名
8 詳細ログのセッションID	関連する詳細ログのセッションID。 関連する詳細ログの項目がない場合は空になります。
9 ログ種別	D: デバッグ情報ログ I: 情報ログ W: 警告ログ E: エラーログ C: 致命的エラーログ
10 イベントソース	FTRANP80: F*TRAN+V8. 0で出力されたログを表す
11 イベント種別	イベント毎の種別ID。(※1を参照)
12 メッセージ番号	メッセージ番号(10進数)
13 メッセージ	最大512文字、なお、ログファイル中では"¥"エスケープされるとこの最大数を超える場合があります。

◆参考 ---- メッセージ一覧

ログのメッセージ一覧は本書の付録Bに掲載しています。

(※1) イベント種別

イベント種別ID	イベント内容	イベント詳細
APP	アプリケーション	アプリケーションおよび動作環境の情報
TRANS	ファイル変換	ファイル変換情報
ATLAS	ATLAS	ATLAS98情報
CMD	コマンドオプション、パラメータ	コマンドオプションおよびパラメータのエラー情報
ACTION	動作	動作情報

【出力例】(以下の例は長さの関係で複数行になっていますが、実際には一行で出力されています)

```
#FTB2, 2017-01-01 09:39:54.123, ABCabc12@xz-123,
Abad-1, 1, USERNAME, FMV, , I, FTRANP80, APP, 2000, "F*TRAN+が起動されました (GUI モード)"
```

(※2) プロセスID

プロセスIDはログ中では文字列と表されます。これはWindowsが管理しているプロセスIDとそのプロセスが起動した時間の情報から以下のような13～22桁の文字列となります。

●プロセスID文字列

バイト列																					
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22
プロセス起動日時											-	プロセスID									

プロセス起動日時:
プロセスが起動した日時を64ビット
(FILETIME形式)を6ビットで区切って文字
列化したもので表す。

プロセスID:
Windowsが管理しているFP.EXEのプロセスID
を10進の数値文字列で表す。0～4294967295
を取るため1～10桁の文字列となる

(※3) セッションID

セッションIDはセッション開始日とセッション作成から順に付けられた1からの連番をー(ハイフン)で連結した下記のような6～25桁の文字列となります。

●セッションID文字列

バイト列																								
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
年	月	日	-	シーケンシャル番号																				

セッション開始日

セッション開始日:
セッションが開始した年月日を格納する。
年は実際の西暦から2000を減算したものにな
る。(2000から6095年までを表現可能)
6ビットで区切って文字列化したもので表す。

シーケンシャル番号:
プロセス内でセッションが作成された順に振られる1
からの連番を10進数の文字列で表す。
64ビットの数値をとり1～18446744073709551616に
なるため最大20桁となる。

プロセスID文字列のプロセス起動日時とセッションID文字列のセッション開始日の文字列化は数値上位ビットから6ビットずつ区切り以下の表により文字列変換(64進数化)をおこなったものです。

数値	文字	数値	文字	数値	文字	数値	文字
0	0	16	G	32	W	48	m
1	1	17	H	33	X	49	n
2	2	18	I	34	Y	50	o
3	3	19	J	35	Z	51	p
4	4	20	K	36	a	52	q
5	5	21	L	37	b	53	r
6	6	22	M	38	c	54	s
7	7	23	N	39	d	55	t
8	8	24	O	40	e	56	u
9	9	25	P	41	f	57	v
10	A	26	Q	42	g	58	w
11	B	27	R	43	h	59	x
12	C	28	S	44	i	60	y
13	D	29	T	45	j	61	z
14	E	30	U	46	k	62	@
15	F	31	V	47	l	63	#

■ 詳細ログファイルについて

詳細ログファイルは、基本ログファイルに関連付けられ、基本ログ項目の詳細情報を記録しているファイルです。現在は変換実行時の変換漏れ情報が記録されます。

詳細ログファイルは以下のようなファイル名で出力されます。

```
ftDetail_YYYYMMDD_nnnnnn.log
```

ファイル名の各部分は以下のような意味となっています。

ファイル名の各部分	種別	内容
ftDetail	ログ種別	“ftDetail” は、詳細ログファイルであることを表す。
YYYYMMDD	日付	セッション開始日 例：2006年2月1日なら“20060201”となる
nnnnnn	通番	10進数で“000000”～“999999”の形式をとる。 最初の通番は“000000”となる

詳細ログファイルも基本ログファイルと同様にレコードは、1ファイルに対して追記される形で行なわれます。ログファイルは**容量が1MBを超えると新たに作成**されます。また、**日付が変わると容量に関係なく新たに作成**されます。追記しようとしたファイルが、何らかの理由で書込み禁止であった場合も、新たに作成されます。新たに作成されたファイルは以下のような連番ファイルとなります。

```
ftDetail_20080120_000000.log
```

```
ftDetail_20080120_000001.log
```

```
ftDetail_20080120_000002.log
```

増加した通番のファイルが作成されると、以後の書き込みはそのファイルに対して行われ通番の小さいファイルには追記されません。なお、..._999999.log以降は新たに作成されず、ファイルサイズにかかわらずこのファイルに追記されます。

メインウィンドウからおこなった変換漏れ情報は、メインウィンドウから表示することが可能です。詳細は「4. 15 ログの設定と表示」をご覧ください。

出力されるログファイルはCSV形式のテキストファイル（UTF-8）で、各項目は次のページのような構成になっています。

詳細ログのフォーマット

項目名	内容
1 ログタグ	#FTD2 ログファイルの個別の形式、バージョンを意味します。ログの形式が異なると変更されます。
2 イベント日時	イベント発生日時 YYYY-MM-DD HH:mm:ss.MSC (MSC は m 秒単位の 3 桁)
3 プロセス ID	プロセス ID 文字列 13~22 文字の文字列
4 セッション ID	詳細ログのセッション単位につくプロセス内で一意なセッション ID 文字列
5 セッション SEQ	セッション内の連番。セッション毎に付けられる 1 からの連番。
6 ログ種別	D: デバッグ情報ログ (未使用) I: 情報ログ W: 警告ログ (未使用) E: エラーログ (未使用) C: 致命的エラーログ (未使用)
7 詳細ログ種別	FTRANP80: F*TRAN+V8. 0 で出力されたログを表す
8 詳細情報 ID	ログごとの詳細情報の種別 ID。(※1 を参照)
9 メッセージ	最大 1024 文字 (※2 を参照)

(※1) 詳細情報の種別 (ID)

詳細情報の種別 ID	内容	詳細
GET_KANJI	GET 変換ログ	ホスト→Win 変換時の未定義の拡張漢字情報
PUT_KANJI	PUT 変換ログ	Win→ホスト変換時の未定義の拡張漢字情報

(※2) メッセージ内容

- ◆ 詳細情報 ID が GET_KANJI (ホスト→Win 変換ログ) の場合
以下のように出力されます。

```
未定義拡張漢字 HOSTCODE=XXXX COUNT=nn HOSTREC=nn WINREC=nn HOSTPOS=nn WINPOS=nn MAPROW=nn
```

HOSTCODE=	変換元ファイルの内、KKT ファイルに未設定の拡張漢字のコード (16 進数) を表示
COUNT=	当該「ホスト漢字」が、ファイル内に何回出現したかを表示
HOSTREC=	当該「ホスト漢字」が、変換元ファイル内の何番目のレコードで初めて出現したかを表示
WINREC=	当該「ホストコード」に対して「=” 変換」、「?”xxxx” 変換」等の変換代替処理による出力が行なわれた最初のレコードを表示。大抵の場合は初出ホストレコード番号と一致するが、「フェーズ」機能を利用して変換前処理などを行なった場合は一致しない
HOSTPOS=	「初出ホストレコード番号」のレコードにおいて、当該「ホスト漢字」が最初に出現したバイト桁位置を表示
WINPOS=	「初出Winレコード番号」のレコードにおいて、「=” 変換」、「?”xxxx” 変換」等の変換代替処理による出力がされた最初のバイト桁位置を表示
MAPROW=	「初出ホストレコード番号」のレコードにおいて、当該「ホスト漢字」が最初に出現した項目番号を表示

- ◆詳細情報IDが PUT_KANJI (Win→ホスト変換ログ) の場合
以下のように出力されます。

未定義拡張漢字 WINCODE=□(XXXX) COUNT=nn WINREC=nn WINPOS=nn MAPROW=nn

WINCODE=	変換元ファイルの内、KKT ファイルに未設定の拡張漢字の文字とコード(16進数)を表示
COUNT=	当該「Win漢字」が、ファイル内に何回出現したかを表示
WINREC=	当該「Win漢字」が、変換元ファイル内の何番目のレコードで初めて出現したかを表示
WINPOS=	「初出Winレコード番号」のレコードにおいて、当該「Win漢字」が最初に出現したバイト桁位置を表示
MAPROW=	「初出Winレコード番号」のレコードにおいて、当該「Win漢字」が最初に出現した項目番号を表示

【出力例】(以下の例は長さの関係で複数行になっていますが、実際には一行で出力されています)

```
#FTD2, 2017-01-01 09:39:54. 123, ABCabc12@xz-123, ABEa-40, 1, I, FTRANP80, GET_KANJI,
  "未定義拡張漢字 HOSTCODE=FFD8 COUNT=2 HOSTREC=0 WINREC=0 HOSTPOS=0 WINPOS=0 MAPROW=1"
#FTD2, 2017-01-31 11:41:22. 931, 0780#jnpUnf-2588, 081V-4, 1, I, FTRANP80, PUT_KANJI,
  "未定義拡張漢字 WINCODE= i (FA40) COUNT=2 WINREC=0 WINPOS=10 MAPROW=2"
```

◆注意 ---- ログのフォーマットはF*TRAN+ V5.5から変更されました

V5.3のログファイルとV5.5以降のログファイルはフォーマットが異なります。ログデータの先頭にある「ログタグ」項目で見分けることが可能です。この項目が #FTB1（基本ログ）、#FTD1（詳細ログ）となっている場合、V5.3形式のログファイルです。

変更内容は以下のようになっています。

- これまでのセッションのみの管理でなく、プロセスを意識して管理できるフォーマットに改めました。「セッションID」項目を「プロセスID」と「セッションID」に分離しました。
- ログの項目の「イベント種別」のカテゴリを整理し、ログにつけられたイベント種別を見直しました。
- メッセージ自体を見直しました。
- 基本ログと詳細ログのリンク項目を変更しました。
- 基本ログのログ種別とイベント種別／詳細情報の種別内容を見直しました。

ログファイルの項目は以下のように変更になりました。

基本ログファイル項目の変更

V5.3		V5.5 以降		変更 有無
	項目名		項目名	
1	ログタグ	1	ログタグ	有り
2	イベント日時	2	イベント日時	有り
	—	3	プロセスID	追加
3	セッションID	4	セッションID	有り
4	セッションインデックス	5	セッションSEQ	無し
5	ユーザ名	6	ユーザ名	無し
6	コンピュータ名	7	コンピュータ名	無し
	—	8	詳細ログのセッションID	追加
7	ログ種別	9	ログ種別	無し
8	イベントソース	10	イベントソース	有り
9	イベント種別	11	イベント種別	有り
10	情報	12	メッセージ番号	有り
11	メッセージ	13	メッセージ	有り

詳細ログファイル項目の変更

V5.3		V5.5 以降		変更 有無
	項目名		項目名	
1	ログタグ	1	ログタグ	有り
2	イベント日時	2	イベント日時	有り
	—	3	プロセス ID	追加
3	セッション ID	4	セッション ID	有り
4	セッションインデックス		—	削除
5	詳細ログインデックス	5	セッション SEQ	無し
6	ログ種別	6	ログ種別	無し
7	詳細ログ種別	7	詳細ログ種別	有り
8	詳細情報 ID	8	詳細情報 ID	無し
9	メッセージ	9	メッセージ	無し

付 録 A

エラーメッセージ
(共通)

●起動時（F P . S Y S、F T R A N . I N I読み込み）のエラーメッセージ

システムファイルが開けません.

動作環境設定ファイル“F P . S Y S”ファイルがオープンできませんでした。大抵はF P . S Y Sが「%USERPROFILE%\AppData\Roaming\FTRANP¥」に見つからないのが原因です。再度インストーラを起動して「ようこそ」画面から「修正」を選択し、再インストールを実行してみてください。

システムファイル中の[XXXX]セクションの記述に誤りがあります.

F P . S Y Sファイル中の[XXXX]項目の記述に誤りがあります。F P . S Y Sファイルをメモ帳などで開いて記述が正しいか確認してください。良く分からない場合は、購入元、または販売元にお問合せください。（“[XXXX]”には任意の名前が入ります）

ローカル環境ファイルが開けません.

主設定ファイル“F T R A N . I N I”ファイルがオープンできませんでした。F P . S Y Sファイルで指定している環境フォルダ配下にF T R A N . I N Iファイルがあるか確認してください。または、F P . S Y Sファイルの環境指定を正しく修正してください。

ローカル環境ファイル中のXXXXに誤りがあります.

F T R A N . I N Iファイル中のXXXX項目に誤りがあります。F T R A N . I N Iファイルをメモ帳などで開いて記述が正しいか確認してください。良く分からない場合は、購入元、または販売元にお問合せください。（“XXXX”には任意の名前が入ります）

●起動・終了時のエラーメッセージ

ファイルのオープンに失敗しました.

F*TRAN+の起動時に、コード変換表ファイルがオープンできませんでした。大抵はコード変換表のファイル名のつづりミスです。

必要なら/H o s tオプションを指定してください。/H o s tオプションを指定したのにこのメッセージが出るときは、/H o s tオプションの指定の誤りです。修正して再起動してください。

ファイルの読み込みに失敗しました.

何らかの入出力エラーが発生し、コード変換表ファイルを読み込めませんでした。このエラーは、めったに発生しません。

コード変換表(CCT) ファイルの形式が正しくありません.

F*TRAN+の起動時に、/H o s tオプションで指定したコード変換表ファイルは実際にはコード変換表ではありませんでした。正しいコード変換表ファイルを指定してください。

ファイルへの書き込みに失敗しました.

何らかの入出力エラーが発生し、コード変換表ファイルの書き込み（保存）ができませんでした。

●共通のエラーメッセージ

メモリが足りません.

作業に必要なメモリの空き容量が足りません。他のアプリケーションを終了させるなどメモリ不足を解消してください。

コマンドがありません.

指定されたコマンドは、ありません。正しいコマンドを指定してください。

未定義のコマンドオプションが指定されました.

該当するオプションがありません。たぶん他のオプションのつづり誤りでしょう。間違って別のコマンドを呼び出していたということもあります。

コマンドオプションに誤りがあります.

オプションの指定に何らかの誤りがあります。修正して再実行してください。

余分なパラメータが指定されています.

パラメータの指定に誤りがあるため、うしろの部分が余分だとみなされました。修正して再実行してください。

エラーが発生しました. 内部エラーの可能性がります.

F * T R A N +自身のエラーが発生しました。購入元または発売元に通報してください。

●パラメータファイルに関するエラーメッセージ

++の後にパラメータファイルを指定してください.

パラメータファイル参照の開始マーク ++ があるのに、そのあとにファイルが指定されていませんでした。++のあとに空白を置いた場合も同様です。空白を置いてはいけません。

ファイルのオープンに失敗しました.

指定されたパラメータファイルがオープンできませんでした。大抵はファイル名のつづり誤りです。

ファイルの読み込みに失敗しました.

指定されたパラメータファイルの読み込み時に入出力エラーが発生しました。このエラーは、めったに発生しません。

パラメータバッファがいっぱいです.

パラメータファイルを展開した結果、約64Kバイトあるパラメータバッファがいっぱいになってしまいました。パラメータファイル内の記述を単純化する必要があります。

●ホストファイルに関するエラーメッセージ

ホストファイルのパス名が長すぎます.

ホストファイルの指定のパス名が長すぎます。ファイルを位置や、パス名の指定の変更を検討してください。

ホストファイル名が指定されていません.

ホスト側のファイル名が必要なところです。指定してください。

ホストファイルが存在しません.

指定されたホストファイルはありません。

ホストファイルが作成できません.

ホストファイルを作成できません。つぎのような原因が考えられます。

ファイル名や拡張子に使えない文字を含んでいた
ホストファイルが書き込み禁止になっていた
目的のディレクトリがなかった

ホストファイルがオープンできません.

指定されたホストファイルをオープンできません。そのホストファイルがない場合が大部分です。

ホストファイルがクローズできません.

ホストファイルをクローズできません。ディスクの容量不足などが考えられます。

ディスクへ出力時にこのメッセージが出たときは、そのディスクが壊れた可能性があります。すぐに、ユーティリティ・プログラム等で検査・修復してください。

ホストファイルの書き込みでエラーになりました.

ホストファイルの書き込みでエラーになりました。ほとんどの場合、ディスクがいっぱいになってそれ以上書き込めなくなったことを意味しています。

ホストファイルの読み込みでエラーになりました.

ホストファイルの読み込みでエラーになりました。通常は発生しないエラーです。

●Windowsファイルに関するエラーメッセージ

Windowsファイルの指定に誤りがあります.

Windowsファイルの指定に構文的な誤りがあります。指定し直してください。

Windowsファイル名が必要です.

Windows側のファイル名が必要なところですが。指定してください。

Windowsファイルがありません.

指定されたWindowsファイルはありません。

Windowsファイルを作成できません.

Windowsファイルを作成できません。つぎのような原因が考えられます。

ファイル名や拡張子に使えない文字を含んでいた
Windowsファイルが書き込み禁止になっていた
目的のディレクトリがなかった

Windowsファイルをオープンできません.

指定されたWindowsファイルをオープンできません。そのWindowsファイルがない場合が大部分です。

Windowsファイルをクローズできません.

Windowsファイルをクローズできません。ディスクの容量不足などが考えられます。

ディスクへ出力時にこのメッセージが出たときは、そのディスクが壊れた可能性があります。すぐに、ユーティリティ・プログラム等で検査・修復してください。

Windowsファイルの書き込みでエラーになりました.

Windowsファイルの書き込みでエラーになりました。ほとんどの場合、ディスクがいっぱいになってそれ以上書き込めなくなったことを意味しています。

Windowsファイルの読み込みでエラーになりました.

Windowsファイルの読み込みでエラーになりました。通常は発生しないエラーです。

● Get Data (gd) コマンドのエラーメッセージ

MAPバッファがいっぱいです.

変換する項目数が多すぎて、MAPバッファがいっぱいになりました。項目数を減らせないかどうか検討してください。

● Get Rand (gr) コマンドのエラーメッセージ

MAPバッファがいっぱいです.

変換する項目数が多すぎて、MAPバッファがいっぱいになりました。項目数を減らせないかどうか検討してください。

● Put Data (pd) コマンドのエラーメッセージ

/MAPオプションの各項目をコンマ(,)で区切ってください.

/Delimitedオプションを指定したのに、/MAPオプションでコンマ(,)を1つも指定していませんでした。

MAPバッファがいっぱいです.

変換する項目数が多すぎて、MAPバッファがいっぱいになりました。項目数を減らせないかどうか検討してください。

● Put Rand (pr) コマンドのエラーメッセージ

MAPバッファがいっぱいです.

変換する項目数が多すぎて、MAPバッファがいっぱいになりました。項目数を減らせないかどうか検討してください。

● `VirDrive (vd)` コマンドのエラーメッセージ

VirDrive コマンドのドライブ名が正しくありません.

第1パラメータとして指定したドライブ名が正しくありません。使えるのは、A : ~ Z : 、@ : 、 ? : だけです。

VirDrive コマンドの第1パラメータはドライブ名だけです.

第1パラメータとして、仮想ドライブ名でないものを指定しようとしてしまいました。誤りです。

VirDrive コマンドにより生成するパス名が長過ぎます.

第2パラメータとして指定した実ドライブ・ディレクトリを表すパス名が63文字を超えました。長過ぎます。

● `Code (co)` コマンドのエラーメッセージ

指定された漢字変換方式名はありません.

`Code (co)` コマンドのパラメータとして指定した漢字変換方式が、コード変換表に登録されていません。パラメータをつけずに `Code (co)` コマンドを呼び出して漢字変換方式の一覧を表示し、そのなかに目的の漢字変換方式があるかどうか調べてください。単純なつづりミスでないかどうか調べてください。

● `Comment (com)` コマンドのエラーメッセージ

コメントが長すぎます.

コード変換表につけるコメント文字列が40文字（漢字20文字）を超えました。もう少し短いコメントにしてください。

● `cLoad (cl)` コマンドのエラーメッセージ

コード変換表(CCT) ファイルの形式が正しくありません.

`cLoad (cl)` コマンドで指定したコード変換表ファイルは、実際にはコード変換表ファイルではありませんでした。正しいコード変換表ファイルを指定してください。

ファイルのオープンに失敗しました.

`cLoad (cl)` コマンドで読み込もうとしたコード変換表ファイルが、オープンできませんでした。コード変換表のファイル名のつづりを間違えた場合が大部分です。ドライブ名、ファイル名、拡張子をチェックしてください。

ファイルの読み込みに失敗しました.

何らかの入出力エラーが発生し、コード変換表ファイルを読み込めませんでした。このエラーは、めったに発生しません。

● `cSave (cs)` コマンドのエラーメッセージ

ファイルへの書き込みに失敗しました.

コード変換表の書き込み（保存）に失敗しました。つぎのような原因が考えられます。

- 目的のディレクトリが見つからなかった
- ファイルが書き込み禁止だった
- ディスクの残り容量不足だった

別のドライブや名前を指定して、再実行してください。

付 録 B

ログメッセージ一覧

基本ログに出力されるメッセージ一覧

メッセージ番号	ログ種別	イベント種別	メッセージ内容
4001	E	CMD	++ の後にパラメータファイルを指定してください。
4002	E	CMD	パラメータバッファがいっぱいです。
4003	E	CMD	コマンドがありません。
4004	E	CMD	未定義コマンドです。
4005	E	CMD	余分なパラメータが指定されています。
4006	E	CMD	MAPバッファがいっぱいです。
4007	E	CMD	/MAPオプションの各項目をコンマ(,) で区切ってください。
4008	E	CMD	VirDrive コマンドのドライブ名が正しくありません。
4009	E	CMD	Windows側のドライブ名が正しくありません。
4010	E	ACTION	パラメータファイルの読み込みに失敗しました。
4011	D	APP	DBGMSG:
4019	E	CMD	VirDrive コマンドの第1パラメータはドライブ名だけです。
4020	E	CMD	VirDrive コマンドにより生成するパス名が長過ぎます。
4021	E	CMD	パラメータの指定に誤りがあります。
4022	E	CMD	コメントが長過ぎます。
4023	E	CMD	指定されたコード変換方式名はありません。
4024	E	CMD	コード変換表(CCT)ファイルの形式が正しくありません。
4025	E	CMD	コマンドオプションに誤りがあります。
4026	E	CMD	未定義のコマンドオプションが指定されました。
4027	E	ACTION	ファイルが作成できません。
4028	E	ACTION	ファイルのオープンに失敗しました。
4029	E	ACTION	ファイルのクローズに失敗しました。
4030	E	ACTION	ファイルの読み込みに失敗しました。
4031	E	ACTION	ファイルへの書き込みに失敗しました。
4032	E	CMD	Windowsファイルの指定に誤りがあるか、パス名が長過ぎます。
4033	E	ACTION	Windowsファイルを作成できません。
4034	E	ACTION	Windowsファイルをクローズできません。
4035	E	ACTION	Windowsファイルの書き込みでエラーになりました。
4036	E	ACTION	Windowsファイル名が必要です。
4037	E	ACTION	Windowsファイルがありません。
4038	E	ACTION	Windowsファイルをオープンできません。
4039	E	ACTION	Windowsファイルの読み込みでエラーになりました。
4040	E	ACTION	ホストファイルがオープンできません。
4041	E	ACTION	ホストファイルがクローズできません。
4042	E	ACTION	ホストファイルが削除できません。
4043	E	ACTION	ホストファイルが作成できません。
4045	E	CMD	ホストファイルが存在しません。
4047	E	CMD	ホストファイルのパス名が長過ぎます。
4048	E	CMD	ホストドライブ名は省略できません。
4049	E	CMD	ホストファイル名が指定されていません。
4050	E	ACTION	ホストファイルが書き込み禁止です。
4051	E	ACTION	KKT ファイルがオープンできません。
4052	E	ACTION	KKT ファイル読み込み中にエラーが発生しました。
4053	E	ACTION	KKT ファイルにエラーがありました。
4055	E	ACTION	KKT ファイルが漢字体系と一致しません。
4056	E	ACTION	漢字体系基準テーブル読み込み中にエラーが発生しました。

メッセージ番号	ログ種別	イベント種別	メッセージ内容
4057	E	ACTION	漢字体系基準テーブルが異常です.
4058	E	LOG	ログファイルを作成できません.
4059	E	LOG	ログファイル書き込み中にエラーが発生しました.
4060	E	LOG	ログファイルがクローズできません.
4061	E	CMD	ホストファイルのパス名が長すぎます.
4062	E	CMD	Windowsファイル名が長すぎます.
4063	E	CMD	入出力ファイル名が同一です.
4064	E	CMD	ホスト側のドライブ名が正しくありません.
4065	W	TRANS	入力レコード長が間違っている可能性があります.
4066	W	TRANS	『0E 変換』の指定は『ゲタ変換』の指定に変更されました.
4067	W	APP	サンプル版では変換できるレコード数に制限があります. 制限数
4068	I	APP	F*TRAN+が起動されました(GUI モード)
4069	I	APP	F*TRAN+が起動されました(コマンドモード)
4070	I	APP	F*TRAN+が起動されました(OLE オートメーション)
4071	I	APP	F*TRAN+は正常終了しました.
4072	I	APP	OLE オートメーションは終了しました.
4073	I	TRANS	変換開始
4074	I	TRANS	変換を終了しました.
4075	E	TRANS	変換は異常終了しました.
4076	I	TRANS	ホスト→Win テキストファイル変換開始
4077	I	TRANS	ホスト→Win データファイル変換開始
4078	I	TRANS	ホスト→Win ランダムファイル変換開始
4079	I	TRANS	Win→ホスト テキストファイル変換開始
4080	I	TRANS	Win→ホスト データファイル変換開始
4081	I	TRANS	Win→ホスト ランダムファイル変換開始
4082	I	TRANS	変換オプションの指定
4083	E	TRANS	変換はユーザによって中断されました.
4084	E	APP	F*TRAN+はエラー終了しました.
4085	I	TRANS	変換中に未定義の拡張漢字は検出されませんでした.
4086	W	TRANS	変換中に未定義の拡張漢字が検出されました.
4087	E	TRANS	Windowsファイルのサイズが制限を超えています.
4088	E	TRANS	ホストファイルのサイズが制限を超えています.
4089	E	TRANS	Windowsファイルが既に存在する為、変換できません.
4090	E	TRANS	ホストファイルが既に存在する為、変換できません.
4091	W	CMD	/ENVironment オプションに誤りがある為、デフォルトの設定を適用しました. ¥n 環境設定を確認してください.
4092	W	CMD	/ENVironment オプションに “ * ” は指定できなくなりました. ¥n オプションの指定を確認してください.
4093	W	CMD	/LogPath オプションは使用できない為、無視しました. ¥n オプションの指定を確認してください.
4094	W	CMD	/ErrorLogName オプションは使用できない為、無視しました. ¥n オプションの指定を確認してください.
4095	W	CMD	/ConvertLogName オプションは使用できない為、無視しました. ¥n オプションの指定を確認してください.
4096	E	ACTION	出力ファイルの書き込みでエラーになりました.
4097	E	ACTION	BOM の書き込みでエラーになりました.
4098	E	ACTION	BOM の読み込みでエラーになりました.
4099	W	CMD	/Log オプションにログフォルダの指定がありません. ¥n ログは標準ログ設定の出力先に出力されます.
4100	W	CMD	/Log オプションで指定されたログフォルダの指定が正しくありません. ¥n ログは標準ログ設定の出力先に出力されます.

メッセージ番号	ログ種別	イベント種別	メッセージ内容
4101	W	CMD	/Log オプションで指定されたログファイルのベース名が正しくありません。¥n ベース名は @DATETIME@ になります。
4102	W	CMD	/Log オプションで指定されたログファイルのパス名が長すぎます。¥n ログは標準ログ設定の出力先に出力されます。
4103	E	CMD	/Log オプションのベース名が長すぎます。
4104	E	CMD	オプションで指定したパス名が長すぎます。
4300	E	ACTION	システムファイルが開けません。
4301	E	ACTION	システムファイルのエンコーディングが正しくありません。
4302	E	ACTION	システムファイル中の[Environments]セクションの記述に誤りがあります。
4303	E	ACTION	システムファイル中の[CurrentEnvironment]の記述に誤りがあります。
4304	E	ACTION	指定されたスクリプトファイルのサイズが大きすぎます。
4305	E	ACTION	指定されたスクリプトファイルのオープンに失敗しました。
4306	E	ACTION	指定されたスクリプトファイルの読み込みに失敗しました。
4307	E	ACTION	スクリプトファイルの形式が違為、読み込めません。
4308	E	ACTION	ローカル環境ファイルが開けません。
4309	E	ACTION	ローカル環境ファイルのエンコーディングが正しくありません。
4310	E	ACTION	ローカル環境ファイル中のホスト種別指定に誤りがあります。
4311	E	ACTION	ローカル環境ファイル中のCCTファイル名指定に誤りがあります。
4312	E	ACTION	ローカル環境ファイル中のエンコーディング名指定に誤りがあります。
4313	E	ACTION	ローカル環境ファイル中の MasterEncoding 設定に誤りがあります。
4314	E	ACTION	ローカル環境ファイル中の MasterEncodingBOM 設定に誤りがあります。
4315	E	ACTION	ローカル環境ファイル中の SystemEncoding 設定に誤りがあります。
4316	E	ACTION	ローカル環境ファイル中の SystemEncodingBOM 設定に誤りがあります。
4317	E	ACTION	ローカル環境ファイル中の UserEncoding 設定に誤りがあります。
4318	E	ACTION	ローカル環境ファイル中の UserEncodingBOM 設定に誤りがあります。
4319	E	ACTION	ローカル環境ファイル中の WindowsCOBOL ベンダ設定に誤りがあります。
4320	E	ACTION	ローカル環境ファイル中の半角／全角設定に誤りがあります。
4321	C	APP	ログファイルアクセスライブラリ (FtLogFile.DLL) が存在しないためログ出力できません。
4322	C	LOG	ログ出力フォルダが存在しないためログ出力できません。
4323	C	LOG	ログ出力フォルダに書き込み権限がないためログ出力できません
4324	I	APP	過去のログファイルを削除しました。
4325	C	APP	OLE ライブラリの初期化に失敗しました。
4326	C	APP	初期化に失敗しました。
4327	C	LOG	エラーログファイルへのメッセージの出力に失敗しました。
5000	E	ATLAS	プログラムエラーが発生しました！
5001	E	ATLAS	項目に誤りがあります。
5002	E	ATLAS	ピクチャを正しく指定してください。
5003	E	ATLAS	幅指定が不正です。
5004	E	ATLAS	引数を正しく指定してください。
5005	E	ATLAS	日付マスクを正しく指定してください。
5006	E	ATLAS	Atlas 宣言から始めてください。
5007	E	ATLAS	Atlas 宣言に誤りがあります。
5008	E	ATLAS	余分な文字が検出されました。
5009	E	ATLAS	手続きが一つもありません。
5010	E	ATLAS	呼び出し先の不明な Call 文があります。
5011	E	ATLAS	手続き名がないか、手続き名の記述に誤りがあります。
5012	E	ATLAS	Proc タグ以外はここには書けません。
5013	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5014	E	ATLAS	手続きを二重定義しています。
5015	E	ATLAS	名標が多すぎます。

メッセージ番号	ログ種別	イベント種別	メッセージ内容
5016	E	ATLAS	Proc に対応する EndProc がありません。
5017	E	ATLAS	Atlas 文の記述が途切れています。
5018	E	ATLAS	名標の記述に誤りがあります。
5019	E	ATLAS	Atlas 文の記述に誤りがあります。
5020	E	ATLAS	Atlas 文の記述に誤りがあります。
5021	E	ATLAS	名標を二重定義しています。
5022	E	ATLAS	予約された名標は使えません。
5023	E	ATLAS	名標を削除中にエラーが発生しました。
5024	E	ATLAS	名標はつけられません。
5025	E	ATLAS	ループ回数の指定に誤りがあります。
5026	E	ATLAS	ループ回数は 0~65535(0xFFFF)の範囲で指定してください。
5027	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5028	E	ATLAS	Loop に対応する EndLoop がありません。
5029	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5030	E	ATLAS	Select 文の型が不正です。
5031	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5032	E	ATLAS	Select 節に変換文がありません。
5033	E	ATLAS	Select 文の記述が途切れています。
5034	E	ATLAS	Case 節の記述に誤りがあります。
5035	E	ATLAS	Select 文には、一つ以上の Case 節が必要です。
5036	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5037	E	ATLAS	Select に対応する EndSelect がありません。
5038	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5039	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5040	E	ATLAS	Case 値の指定に誤りがあります。
5041	E	ATLAS	整数と Min、Max 以外は指定できません。
5042	E	ATLAS	Case 値の範囲指定に誤りがあります。
5043	E	ATLAS	Case 値の値がありません。
5044	E	ATLAS	Case 値の指定は 32 個までです。
5045	E	ATLAS	Case 値の指定に誤りがあります。
5046	E	ATLAS	小数と Min、Max 以外は指定できません。
5047	E	ATLAS	Case 値の範囲指定に誤りがあります。
5048	E	ATLAS	Case 値の値がありません。
5049	E	ATLAS	Case 値の指定は 32 個までです。
5050	E	ATLAS	Case 値の指定に誤りがあります。
5051	E	ATLAS	文字列と Space、LowValue、HighValue 以外は指定できません。
5052	E	ATLAS	Case 値の範囲指定に誤りがあります。
5053	E	ATLAS	Case 値の値がありません。
5054	E	ATLAS	Case 値の指定は 32 個までです。
5055	E	ATLAS	手続き名の指定に誤りがあります。
5056	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5057	E	ATLAS	手続き名を指定してください。
5058	E	ATLAS	名標が多すぎます。
5059	E	ATLAS	Return 文の記述に誤りがあります。
5060	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5061	E	ATLAS	返り値は-32768~32767(0xFFFF)の範囲で指定してください。
5062	E	ATLAS	システム変数以外は指定できません。
5063	E	ATLAS	Break 文は構造文の中だけに指定できます。
5064	E	ATLAS	構造名が違います。
5065	E	ATLAS	構造名が違います。

メッセージ番号	ログ種別	イベント種別	メッセージ内容
5066	E	ATLAS	Break 文の記述に誤りがあります。
5067	E	ATLAS	Break 文の記述に誤りがあります。
5068	E	ATLAS	システム変数以外は指定できません。
5069	E	ATLAS	ブレークコードは-32768～32767(0xFFFF)の範囲で指定してください。
5070	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5071	E	ATLAS	Quit 文の記述に誤りがあります。
5072	E	ATLAS	RecSkip、RecDone 以外は指定できません。
5073	E	ATLAS	Quit 文の記述に誤りがあります。
5074	E	ATLAS	Quit 文の記述に誤りがあります。
5075	E	ATLAS	Continue、EOF 以外は指定できません。
5076	E	ATLAS	余分な記述があるか、}で閉じる必要があります。
5077	E	ATLAS	2進ピクチャを正しく指定してください。
5078	E	ATLAS	列定数の指定に誤りがあります。
5079	E	ATLAS	数値定数の指定に誤りがあります。
5080	E	ATLAS	指定できない定数です。
5081	E	ATLAS	Quit 値は-32768～32767(0xFFFF)の範囲で指定してください。
5082	E	ATLAS	両方のピクチャを同時に省略できません。
5083	E	ATLAS	Atlas のネストが深すぎる為、変換を中断しました。

F*TRAN+ V8.0 操作説明書／解説編

2017年 1月 第1版発行

編集・著作 株式会社 富士通ビー・エス・シー

所在地 〒135-8300 東京都港区台場2-3-1 トレードピアお台場

- Windows、MS-DOS、Visual Basic、Access、Visual C++は米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- Unicode は Unicode コンソーシアムの商標です。
- F*TRAN は富士通ビー・エス・シーの登録商標です。
- 会社名および製品名はそれぞれ各社の商標または登録商標です。
- 本書およびシステムは、改善のため事前連絡なしに変更することがあります。
- 無断複製、および転載を禁じます。