

Hitachi Virtual Storage Platform 5000  
Cross-OS File Exchange

Code Converter for AIX  
Code Converter for HP-UX  
Code Converter for Solaris  
Code Converter for Windows  
Code Converter for Linux

ユーザーズガイド

株式会社日立製作所

All rights reserved, Copyright (C) 2007,2022.Hitachi, Ltd.

## ご注意

- ① 本書の内容の一部または全部を無断で使用、複製することはできません。
- ② 本書の内容については、将来予告なしに変更することがあります。
- ③ 本書を運用した結果の影響については、一切責任を負いかねますのでご了承ください。
- ④ 本書の内容については、万全を期していますが、ご不審な点や誤り等、お気づきのことがありましたら、弊社までご連絡ください。

- \* UNIXは、X/Open Company Limited が独占的にライセンスしている米国ならびに他の国における登録商標です。
- \* IBMは、米国における米国International Business Machines Corp. の登録商標です。
- \* MVSは、米国における米国International Business Machines Corp. の登録商標です。
- \* MVS/ESAは、米国における米国International Business Machines Corp. の商標です。
- \* MVS/XAは、米国における米国International Business Machines Corp. の商標です。
- \* MVS/SPは、米国における米国International Business Machines Corp. の商標です。
- \* RS/6000は、米国における米国International Business Machines Corp. の商標です。
- \* AIXは、米国における米国International Business Machines Corp. の登録商標です。
- \* HP9000は、米国Hewlett-Packard Companyの商品名称です。
- \* HP-UXは、米国Hewlett-Packard Companyのオペレーティングシステムの名称です。
- \* SCD適合ロゴを含むすべてのSPARC商標は、米国における米国SPARC International, Inc. の商標または登録商標です。  
SPARC商標が付いた製品は、米国Sun Micro systems, Inc. が開発したアーキテクチャに基づくものです。
- \* Solarisは、米国SunSoft Inc. の商品名称です。
- \* 富士通は、(株)富士通の登録商標です。
- \* MSP/EXは、(株)富士通の登録商標です。
- \* Microsoft (R) :Microsoftは、米国およびその他の国における米国Microsoft Corp. の登録商標です。
- \* Windows NTは、米国およびその他の国における米国Microsoft Corp. の登録商標です。
- \* Windowsは、米国およびその他の国における米国Microsoft Corp. の登録商標です。
- \* Windows Serverは、米国およびその他の国における米国Microsoft Corp. の商標です。
- \* Red Hatは、Red Hat, Incの米国およびその他の国における登録商標です。
- \* Linuxは、Linus Torvaldsの登録商標です。

なお、本文中では、® および ™ は明記しておりません。

2022年 11月 (第3版)

All rights reserved, Copyright (C) 2007. 2022 Hitachi, Ltd.

## 改訂歴

NO	改訂月	追加機能
第1版	2019年 7月	初版発行
第2版	2022年 5月	製品名称修正
第3版	2022年 11月	RHEL8.4 サポート

# 目 次

1.	概要	1
2.	システム構成	2
2.1.	製品の位置付け	2
2.2.	プログラム構造	4
3.	機能	5
3.1.	英数文字変換機能	7
3.2.	漢字変換機能	8
3.3.	無変換機能 (=単純転送)	9
3.4.	出力抑止変換機能	10
3.5.	数値変換機能	11
3.6.	エラーログ出力機能	12
3.7.	コード変換に関わる編集諸機能	15
3.8.	EUCコード変換処理	18
3.9.	Unicodeコード変換処理	20
4.	Code Converterの使い方	22
4.1.	提供媒体およびインストール方法概説	22
4.1.1.	FCUのインストール及びアンインストール方法	23
4.1.2.	Code Converterのインストール/アンインストール	23
4.1.3.	Code Converter-EUCのインストール/アンインストール	25
4.1.4.	Code Converter-CODECHKのインストール/アンインストール	27
4.1.5.	Code Converter-UNICODEのインストール/アンインストール	29
4.1.6.	提供媒体	30
4.2.	ライセンスキーコードの入力	35
4.3.	使い方概説	38
4.3.1.	変換操作対象となるDBデータファイルのレコード形式	39
4.3.2.	フィールド定義ファイルの作成	45
4.3.3.	パラメタ定義ファイルを使った「フィールド定義ファイル」との関連付け	54
4.3.4.	編集オプションファイル	55
5.	制限事項・その他	64
6.	Code Converterエラー情報	65
6.1.	Code Converterエラーコード/エラーメッセージ	65
6.2.	トラブル発生時の情報収集	72
7.	メインフレーム環境からオープン環境への文字コード変換 (m t o)	73
7.1.	IBMメインフレームコードからシフト J I Sコードへの変換	73
7.1.1.	文字コード変換の概要	73
7.1.2.	英数文字フィールド変換 (CHAR:OP14, OP15)	74
7.1.3.	漢字フィールド変換 (KANJI:OP11)	75
7.1.4.	漢字フィールド変換 (KANJI:OP12, OP13)	76
7.1.5.	コード変換テーブルのカスタマイズ	77
7.2.	日立メインフレームコードからシフト J I Sコードへの変換	78
7.2.1.	文字コード変換の概要	78
7.2.2.	英数文字フィールド変換 (CHAR:OP24, OP34, OP25, OP35)	80
7.2.3.	漢字フィールド変換 (KANJI:OP21, OP31)	80
7.2.4.	漢字フィールド変換 (KANJI:OP22, OP32, OP23, OP33)	81
7.2.5.	コード変換テーブルのカスタマイズ	82
7.3.	富士通メインフレームコードからシフト J I Sコードへの変換	83
7.3.1.	文字コード変換の概要	83
7.3.2.	英数文字フィールド変換 (CHAR:OP44, OP45)	84
7.3.3.	漢字フィールド変換 (KANJI:OP41)	85

7.3.4.	漢字フィールド変換 (KANJI:OP42, OP43)	86
7.3.5.	コード変換テーブルのカスタマイズ	87
7.4.	IBMメインフレームコードからEUCコードへの変換	88
7.4.1.	文字コード変換の概要	88
7.4.2.	英数文字フィールド変換 (CHAR:OPC4, OPC5)	89
7.4.3.	漢字フィールド変換 (KANJI:OPC1)	90
7.4.4.	漢字フィールド変換 (KANJI:OPC2, OPC3)	90
7.4.5.	コード変換テーブルのカスタマイズ	91
7.5.	IBMメインフレームコードからUnicodeへの変換	92
7.5.1.	文字コード変換の概要	92
7.5.2.	英数文字フィールド変換 (CHAR:OP64, OP65)	92
7.5.3.	漢字フィールド変換 (KANJI:OP61)	93
7.5.4.	漢字フィールド変換 (KANJI:OP62, OP63)	94
7.5.5.	コード変換テーブルのカスタマイズ	95
7.6.	日立メインフレームコードからUNICODEへの変換	96
7.6.1.	文字コード変換の概要	96
7.6.2.	英数文字フィールド変換 (CHAR:OP74, OP84, OP75, OP85)	96
7.6.3.	漢字フィールド変換 (KANJI:OP71, OP81)	97
7.6.4.	漢字フィールド変換 (KANJI:OP72, OP82, OP73, OP83)	97
7.6.5.	コード変換テーブルのカスタマイズ	98
7.7.	富士通メインフレームコードからUNICODEへの変換	99
7.7.1.	文字コード変換の概要	99
7.7.2.	英数文字フィールド変換 (CHAR:OP94, OP95)	99
7.7.3.	漢字フィールド変換 (KANJI:OP91)	100
7.7.4.	漢字フィールド変換 (KANJI:OP92, OP93)	100
7.7.5.	コード変換テーブルのカスタマイズ	101
7.8.	日立メインフレームからEUCコードへの変換	102
7.8.1.	文字コード変換の概要	102
7.8.2.	英数文字フィールド変換 (CHAR:OPD4, OPE4, OPD5, OPE5)	103
7.8.3.	漢字フィールド変換 (KANJI:OPD1, OPE1)	104
7.8.4.	漢字フィールド変換 (KANJI:OPD2, OPE2, OPD3, OPE3)	104
7.8.5.	コード変換テーブルのカスタマイズ	105
<b>8.</b>	<b>オープン環境からメインフレーム環境への文字コード変換 (otm)</b>	<b>106</b>
8.1.	シフトJISコードからIBMメインフレームコードへの変換	106
8.1.1.	文字コード変換の概要	106
8.1.2.	英数文字フィールド変換 (CHAR:OP19, OP1A)	107
8.1.3.	漢字フィールド変換 (KANJI:OP16)	108
8.1.4.	漢字フィールド変換 (KANJI:OP17, OP18)	108
8.1.5.	コード変換テーブルのカスタマイズ	109
8.2.	シフトJISコードから日立メインフレームコードへの変換	110
8.2.1.	文字コード変換の概要	110
8.2.2.	英数文字フィールド変換 (CHAR:OP29, OP39, OP2A, OP3A)	111
8.2.3.	漢字フィールド変換 (KANJI:OP26, OP36)	112
8.2.4.	漢字フィールド変換 (KANJI:OP27, OP37, OP28, OP38)	112
8.2.5.	コード変換テーブルのカスタマイズ	113
8.3.	シフトJISコードから富士通メインフレームコードへの変換	114
8.3.1.	文字コード変換の概要	114
8.3.2.	英数文字フィールド変換 (CHAR:OP49, OP4A)	115
8.3.3.	漢字フィールド変換 (KANJI:OP46)	116
8.3.4.	漢字フィールド変換 (KANJI:OP47, OP48)	116

8.3.5.	コード変換テーブルのカスタマイズ	117
8.4.	EUCコードからIBMメインフレームコードへの変換	118
8.4.1.	文字コード変換の概要	118
8.4.2.	英数文字フィールド変換 (CHAR:OPC9, OPCA)	119
8.4.3.	漢字フィールド変換 (KANJI:OPC6)	120
8.4.4.	漢字フィールド変換 (KANJI:OPC7, OPC8)	120
8.4.5.	コード変換テーブルのカスタマイズ	121
8.5.	UNICODEからIBMメインフレームコードへの変換	122
8.5.1.	文字コード変換の概要	122
8.5.2.	英数文字フィールド変換 (CHAR:OP69, OP6A)	122
8.5.3.	漢字フィールド変換 (KANJI:OP66)	122
8.5.4.	漢字フィールド変換 (KANJI:OP67, OP68)	123
8.5.5.	コード変換テーブルのカスタマイズ	123
8.6.	UNICODEから日立メインフレームコードへの変換	124
8.6.1.	文字コード変換の概要	124
8.6.2.	英数文字フィールド変換 (CHAR:OP79, OP89, OP7A, OP8A)	124
8.6.3.	漢字フィールド変換 (KANJI:OP76, OP86)	125
8.6.4.	漢字フィールド変換 (KANJI:OP77, OP87, OP78, OP88)	125
8.6.5.	コード変換テーブルのカスタマイズ	126
8.7.	UNICODEから富士通メインフレームコードへの変換	127
8.7.1.	文字コード変換の概要	127
8.7.2.	英数文字フィールド変換 (CHAR:OP99, OP9A)	127
8.7.3.	漢字フィールド変換 (KANJI:OP96)	128
8.7.4.	漢字フィールド変換 (KANJI:OP97, OP98)	128
8.7.5.	コード変換テーブルのカスタマイズ	129
8.8.	EUCコードから日立メインフレームコードへの変換	130
8.8.1.	文字コード変換の概要	130
8.8.2.	英数文字フィールド変換 (CHAR:OPD9, OPE9, OPDA, OPEA)	131
8.8.3.	漢字フィールド変換 (KANJI:OPD6, OPE6)	131
8.8.4.	漢字フィールド変換 (KANJI:OPD7, OPE7, OPD8, OPE8)	131
8.8.5.	コード変換テーブルのカスタマイズ	132
<b>9.</b>	<b>数値変換機能について</b>	<b>133</b>
9.1.	メインフレーム環境からオープン環境への数値変換 (m t o)	133
9.1.1.	数値変換の概要	133
9.1.2.	パック形式変換 (NUM:OPA1)	133
9.1.3.	ゾーン形式変換 (NUM:OPA2)	136
9.2.	オープン環境からメインフレーム環境への数値変換 (o t m)	139
9.2.1.	数値変換の概要	139
9.2.2.	パック形式変換 (NUM:OPA6)	139
9.2.3.	ゾーン形式変換 (NUM:OPA7)	142
<b>10.</b>	<b>コード変換テーブルのカスタマイズ</b>	<b>145</b>
10.1.	テーブルカスタマイズの概要	145
10.2.	ユティリティの機能	146
10.3.	実行環境の設定	146
10.4.	定義文の文法	147
10.5.	コマンドの文法	153
10.6.	実行例	155
10.7.	CCNVエラーメッセージ	158

<付録>

- 付録A . . . . . 編集オプションファイル
- 付録B . . . . . IBM漢字<->シフトJISコード変換用  
標準提供のコード変換テーブル
- 付録C . . . . . JEF漢字<->シフトJISコード変換用  
標準提供のコード変換テーブル
- 付録D . . . . . KEIS漢字<->シフトJISコード変換用  
標準提供のコード変換テーブル
- 付録E . . . . . IBM漢字<->Unicode変換用  
標準提供のコード変換テーブル
- 付録F . . . . . JEF漢字<->Unicode変換用  
標準提供のコード変換テーブル
- 付録G . . . . . KEIS漢字<->Unicode変換用  
標準提供のコード変換テーブル
- 付録H . . . . . IBM漢字<->EUCコード変換用  
標準提供のコード変換テーブル
- 付録I . . . . . KEIS漢字<->EUCコード変換用  
標準提供のコード変換テーブル

## 1. 概要

### 1. 概要

本マニュアルは、日立マルチプラットフォーム RAID とパッケージ製品である File Exchange ファンクション(FAL for AIX / FAL for HP-UX / FAL for Solaris / FAL for Windows)を使用したシステムにおいて、メインフレームから UNIX/PC サーバへ DB データ等を移行する際、同時に各種漢字のコード変換など、データの一部を変換・編集することを目的とする”メインフレーム/サーバ間データ移行支援 APP(Code Converter)”の機能について説明するものです。

本 APP は、上記 File Exchange ファンクションに属するファイル変換ユーティリティ (FCU: File Conversion Utility) が支援しているユーザオプショニング切口から呼ばれて動作する関数として、標準的な IBM 漢字コード変換などの機能を組み込み、作成したものです。

本 APP を導入することによりユーザは、IBM 漢字コード変換などの目的で当該業務専用 AP を個別開発することなくメインフレーム/サーバ間データ移行システムを構築できるようになります。

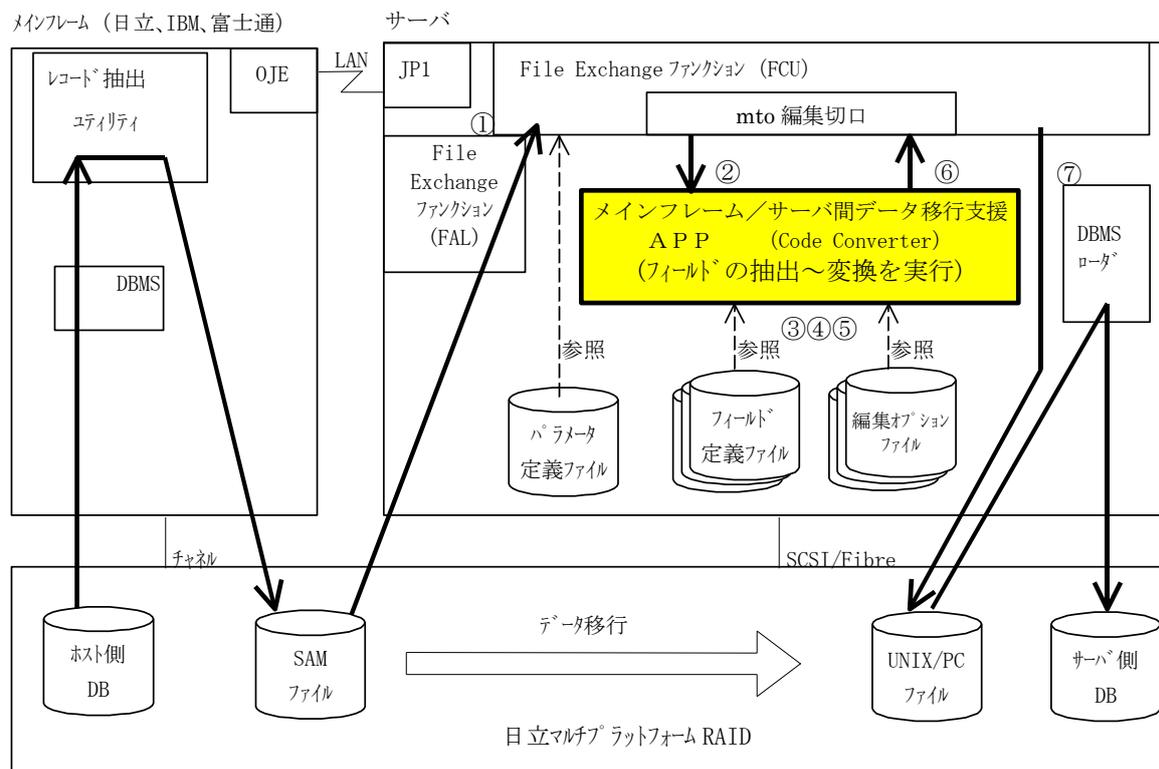
## 2. システム構成

### 2. システム構成

#### 2.1. 製品の位置付け

Code Converter は、ファイル変換ユーティリティから呼ばれて動作するUOC(ユーザ・オンコーディング)関数として機能します。

Code Converter の位置付けと動作の概要について、mto(メインフレーム→オープンシステム)方向の変換を図 2.1.1に、また、otm(オープンシステム→メインフレーム)方向の変換を図 2.1.2に示します。



- (注) ・ JP1 及び OJE (Open Job Entry) は、メインフレーム/サーバ間でのデータ移行するジョブの起動など運用支援に必要なとなる(株)日立製作所提供のパッケージ製品です。  
 ・ フィールドとは、レコードを構成するレコードの一部または全体を示す用語で、Code Converter の基本処理単位です。

<動作概要>

- ① ファイル変換ユーティリティがメインフレーム中間ファイル(SAM)のデータを読み込みます。
- ② ファイル変換ユーティリティがパラメータ定義ファイル(事前にユーザが作成)を参照し、mto 編集切口の使用が宣言されていれば、1レコード分のデータを抽出し、Code Converter へレコードデータを渡します。

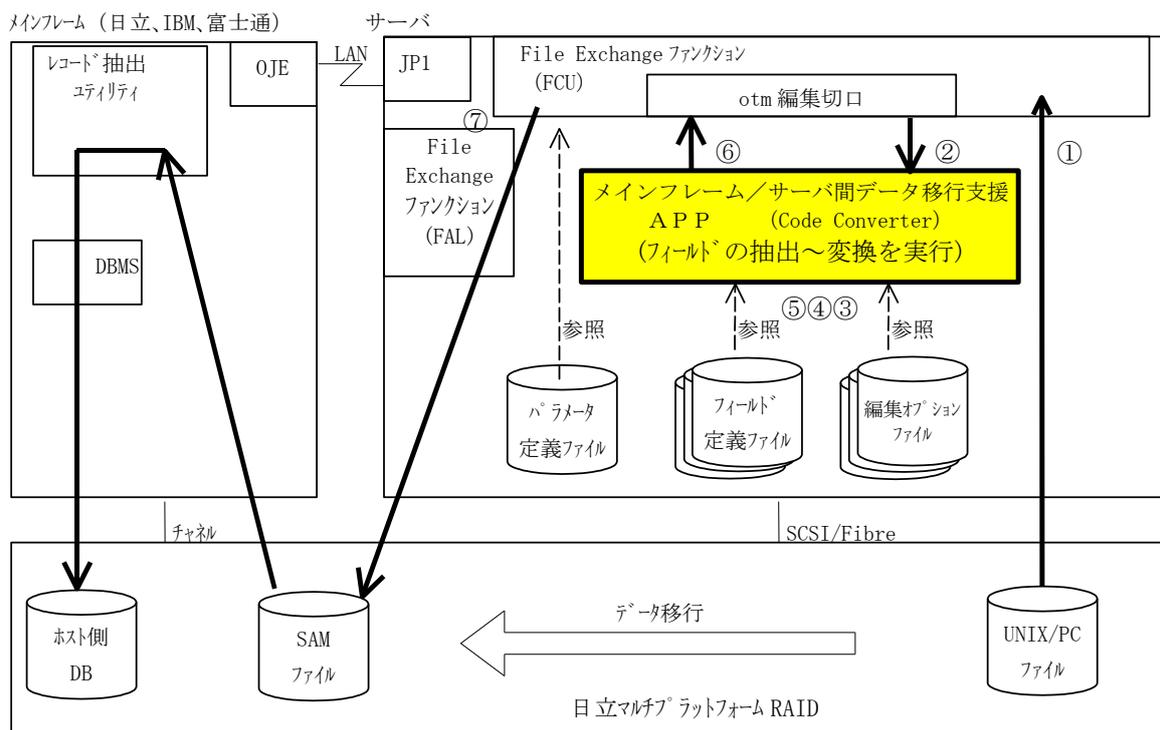
Code Converter の処理範囲 (枠内)

- ③ Code Converter は、フィールド定義ファイル(事前にユーザが作成)を参照し、1フィールド分のデータを抽出すると共に指定された変換種別を認識します。
- ④ フィールド定義ファイルと編集オプションファイル(標準提供)を参照して、漢字コードなどの変換を実行します。
- ⑤ 1レコード分の全フィールドに対して、③④の抽出処理～変換処理を繰り返します。
- ⑥ Code Converter が、1レコード分の全フィールドの変換処理を終了すると、変換後のデータを File Exchange (FCU)に返します。

- ⑦ ファイル変換ユーティリティは、1レコード分のデータをサーバ側ファイル(UNIX等)に出力します。(以下①から繰り返します。)

図 2.1.1 Code Converter の位置付けと動作概要 (m t o 方向)

## 2. システム構成



(注)・ JP1 及び OJE (Open Job Entry) は、メインフレーム/サーバ間でのデータ移行するジョブの起動など運用支援に必要となる(株)日立製作所提供のパッケージ製品です。

- ・ フィールドとは、レコードを構成するレコードの一部または全体を示す用語で Code Converter の基本処理単位です。

### <動作概要>

- ① ファイル変換ユーティリティがサーバ側ファイル (UNIX 等) のデータを読み込みます。
- ② ファイル変換ユーティリティがパラメータ定義ファイル(事前にユーザが作成)を参照し、otm 編集切口の使用が宣言されていれば、1レコード分のデータを抽出し、Code Converter へレコードデータを渡します。

#### Code Converter の処理範囲 (枠内)

- ③ Code Converter は、フィールド定義ファイル (事前にユーザが作成) を参照し、1フィールド分のデータを抽出すると共に指定された変換種別を認識します。
- ④ フィールド定義ファイルと編集オプションファイル (標準提供) を参照して、漢字コードなどの変換を実行します。
- ⑤ 1レコード分の全フィールドに対して、③④の抽出処理～変換処理を繰り返します。
- ⑥ Code Converter が、1レコード分の全フィールドの変換処理を終了すると、変換後のデータを File Exchange (FCU) に返します。

- ⑦ ファイル変換ユーティリティは、1レコード分のデータをメインフレーム中間ファイル (SAM) へ出力します。(以下①から繰り返します。)

図 2.1.2 Code Converter の位置付けと動作概要 (otm 方向)

(注意) デリミタが付加されていないサーバ側ファイルを入力とする場合、FCU はメインフレーム側ファイルのデータ長と同じ長さで当該ファイルが構成されているものと仮定して動作します。このため、otm 変換した結果、レコード長が変わるような場合には必ず、サーバ側ファイルのデータにデリミタを付加してください。

## 2. システム構成

### 2.2. プログラム構造

Code Converter のプログラム構造概要を、図 2.2.1 に示します。

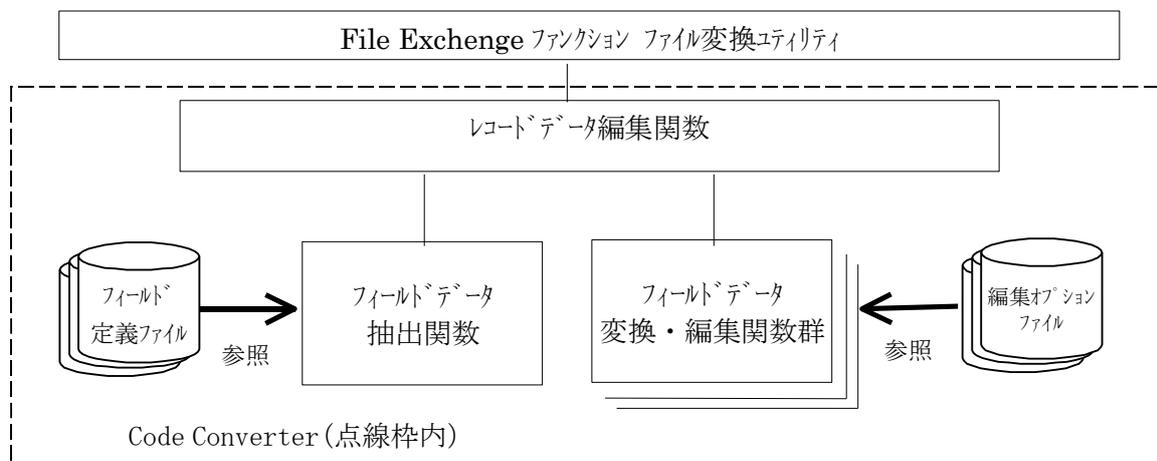


図 2.2.1 Code Converter のプログラム構造概要

フィールドデータ抽出関数が参照する「フィールド定義ファイル」は、変換操作対象となるDBデータファイル毎に異なるレコード形式（フィールド構成）に対応させて作成するもので、入力側レコードに含まれる全フィールドについて、その（位置／長さ／変換操作方法／出力するときの長さ）を、一定の書式に従って、ユーザがテキストファイル形式で記述するものです。

フィールドデータ変換・編集関数群が参照する「編集オプションファイル」は、Code Converter で支援している変換様式について詳細を定義するもので、オープン環境（UNIX/PCサーバ）のカレントディレクトリ上に格納して使います。

### 3. 機能

### 3. 機能

Code Converter の機能概要を、表 3.1 に示します。

表 3.1 Code Converter の機能概要

項番	機能	概要	備考
1	英数文字変換機能	<p>半角文字列のみから成るフィールドデータに対して、コード変換を行います。</p> <p>メインフレーム側 → EBCDIC / EBCDIK (注1)                      オープンシステム側 → JIS8 / ASCII (注2) , EUCコードセット 0/2 , Unicode                      の半角文字コードを変換します。</p> <p>(注1) 通称 EBCDIC にはカナ文字を含まないコード体系と含むコード体系の2種類があり、Code Converter では両者を区別するため、前者(カナ文字を含まないコード体系)を <b>EBCDIC</b>、後者(カナ文字を含むコード体系)を <b>EBCDIK</b> と表記し、使い分けています。</p> <p>(注2) ASCII コードは、カナ文字を含まないコード体系であるため、EBCDIC コード体系のみを変換の対象としています。                      また、EBCDIC⇔ASCII 変換テーブルに対しては、若干のカスタマイズが必要になる場合があります。(4.3.2項 表 4.3.7参照)</p>	EUCコードセット2はシフトコードを付加して2バイトになるのでご注意ください。
2	漢字変換機能	<p>全角文字列のみ、または全角文字と半角文字が混在する文字列から成るフィールドデータに対して、コード変換を行います。</p> <p>メインフレーム側 → IBM 漢字 , JEF 漢字 , KEIS78 , KEIS83/90                      オープンシステム側 → シフト JIS , EUCコードセット 0/1/2/3 , Unicode                      の全角文字コードを変換します。</p>	EUCコードセット3はシフトコードを付加して3バイトになるのでご注意ください。
3	無変換機能 (=単純転送)	入力フィールド内のデータに何も手を加えずに、そのまま出力します。	入力データは文字に限定しません。
4	出力抑止変換機能	入力フィールド内のデータを読み捨てて出力しません。	出力コード長がその分縮まります。
5	数値変換機能	<p>数値文字列のみから成るフィールドデータに対して、コード変換を行います。</p> <p>メインフレーム側 → パック形式 , ゾーン形式                      オープンシステム側 → 数値文字                      の数値文字コードを変換します。</p>	
6	エラーログ出力機能	未定義文字やシフトコードのチェックをし、ログファイルに出力を行います。	オプション製品(CODECHK)が必要です。
7	外字登録機能	<p>提供する標準変換テーブルに外字コードを設定することで変換テーブルのカスタマイズを行います。</p> <p>標準テーブルのカスタマイズは、Code Converter が提供するユーティリティ(CCNV)を使用して行います。また、弊社サービスメニューも提供しております。</p>	カスタマイズの詳細は、10章を参照してください。

### 3. 機能

表 3.1、項番 1～6 の各変換動作の概要を 3.1 節～3.6 節に図で示します。

更に、項番 1～2 の文字コード変換に関わる編集諸機能のうち、以下(1)～(6)の変換動作の概要を 3.7 節に図で示します。

[コード変換に関わる編集諸機能]

- (1) 未定義文字コード処理 (半角)
- (2) 未定義文字コード処理 (全角)
- (3) シフトコード削除処理 (全半角混在時)
- (4) シフトコード挿入処理 (全半角混在時)
- (5) シフトコード挿入抑止処理 (全半角混在時)
- (6) 出力先フィールドの方が短い場合の処理

また、オプション製品として、EUCコード変換処理を 3.8 節に、Unicode 変換処理を 3.9 節に示します。

### 3. 機能

#### 3.1. 英数文字変換機能

半角文字列のみから成るフィールドデータに対して、コード変換を行います。対象とするコード体系は次の通りです。

- ・メインフレーム側コード体系 → EBCDIC(かな文字を含まない) または EBCDIK(かな文字を含む)
- ・オープンシステム側コード体系 → JIS8 または ASCII (注) , EUCコードセット 0/2 , Unicode

(注) ASCIIコードは、かな文字を含まないコード体系であるため、EBCDICコード体系のみを変換の対象としています。また、EBCDIC⇔ASCII変換テーブルに対しては、若干のカスタマイズが必要になる場合があります。(4.3.2項 表 4.3.7参照)

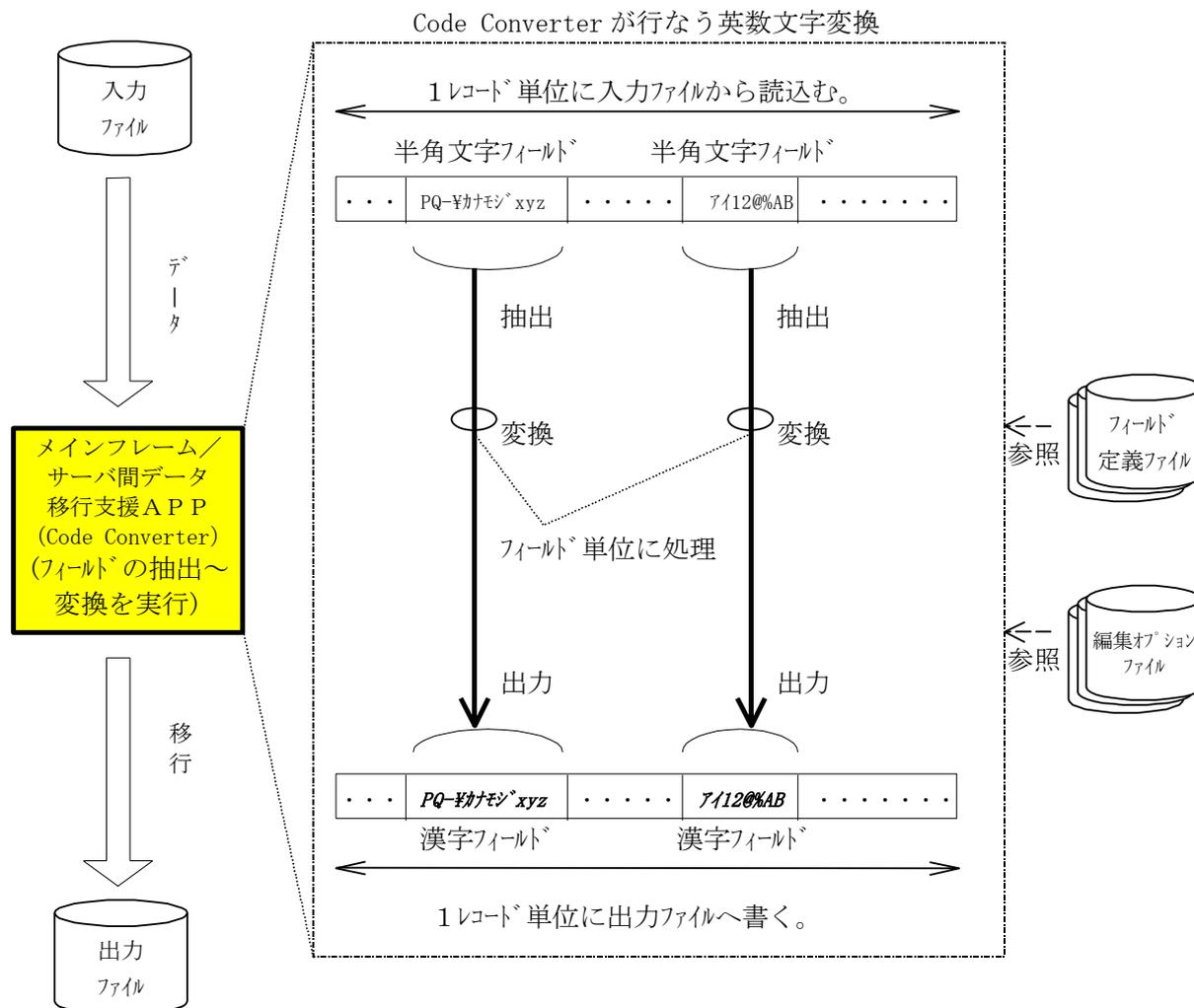


図 3.1.1 英数文字変換概要

[説明]

1レコード単位に入力ファイルからデータを読み込み、その中から変換対象となる半角文字フィールドの部分抽出し、指定の英数文字変換を行います。変換は1フィールド毎に実施、結果は1レコード単位に編集し、出力ファイルへ書き込みます。

レコード内の半角文字フィールドを抽出する位置や、出力するときのフィールド長は、ユーザ作成の「フィールド定義ファイル」中の指定値を参照することにより認識します。(詳細は、4.3.2項参照)

また、英数文字変換方法の詳細(使用する変換テーブルの種類等)の指定は、標準提供の「編集オプションファイル」中のデフォルト指定パラメータにより制御しています。(詳細は、4.3.4項参照)

### 3. 機能

#### 3.2. 漢字変換機能

全角文字列のみ、または全角文字と半角文字が混在する文字列から成るフィールドデータに対して、コード変換を行います。対象とするコード体系は、次の通りです。

- ・メインフレーム側 → IBM 漢字, JEF 漢字, KEIS78, KEIS83/90  
(半角文字が混在する場合は、EBCDIC(カナ文字を含まない) または EBCDIK(カナ文字を含む))
- ・オープンシステム側 → シフトJIS, EUCコードセット 0/1/2/3, Unicode  
(半角文字が混在する場合は、JIS8, EUCコードセット 0/1/2/3, Unicode)

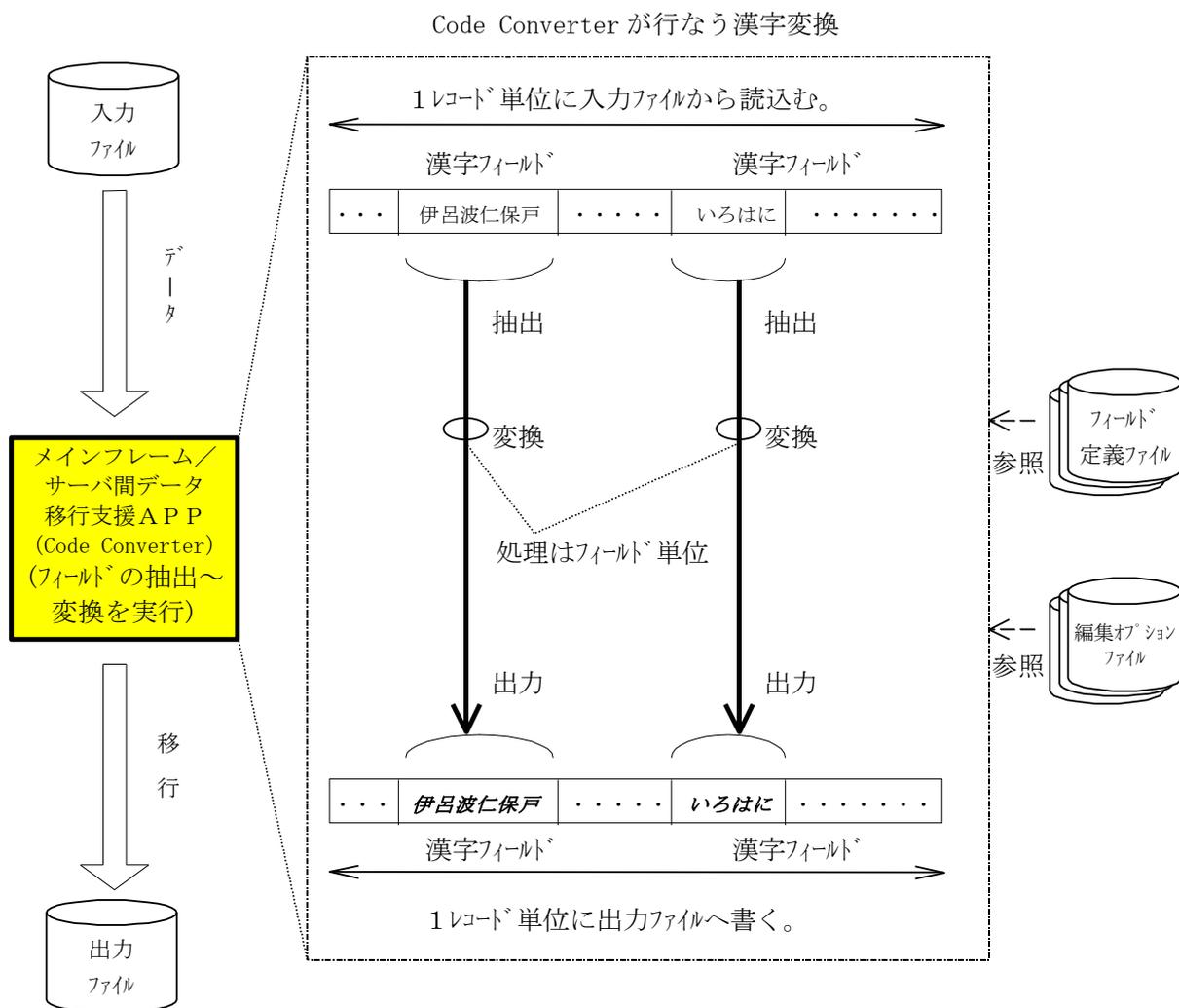


図 3.2.1 漢字変換概要

[説明]

1レコード単位に読み込み、その中から変換対象となる漢字フィールド(半角文字混在可)の部分を抽出し、指定の漢字変換を行います。

変換の要領は、3.1節の英数文字変換の場合と同じです。

### 3. 機能

#### 3.3. 無変換機能 (=単純転送)

入力フィールド内のデータに何も手を加えずに、そのまま出力します。入力データは文字コードに限定しません。

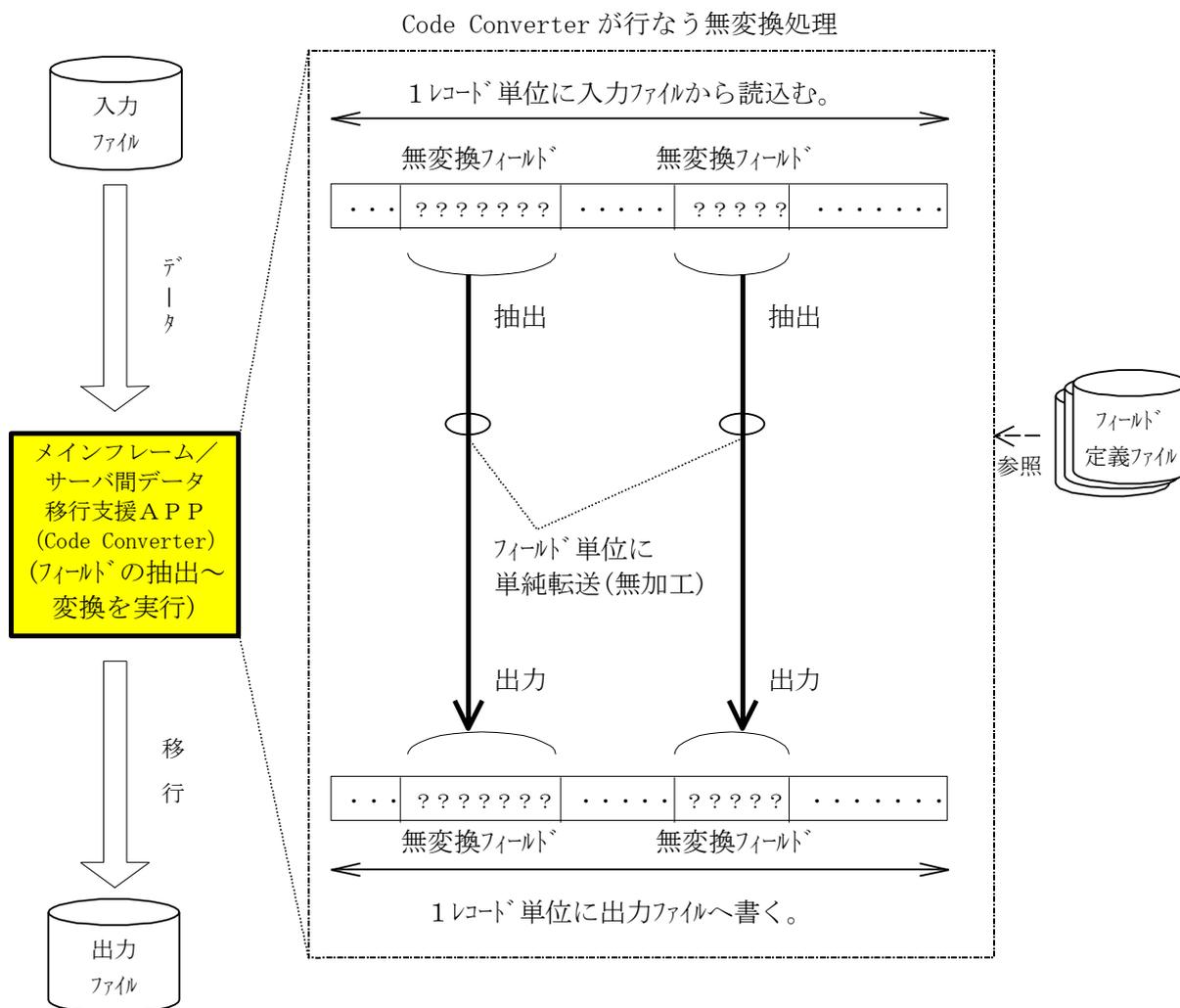


図 3.3.1 無変換概要

#### [説明]

1レコード単位に入力ファイルからデータを読み、その中から変換対象となる無変換フィールド(無加工)の部分抽出し、単純転送を行います。

レコード内の無変換フィールドを抽出する位置は、ユーザ作成の「フィールド定義ファイル」中の指定値を参照することにより認識します。(詳細は、4.3.2項参照)

- (注) 無変換の場合、出力フィールド長は、入力フィールド長と必ず同じにしてください。「フィールド定義ファイル」の中で、出力フィールド長の記述を省略すると、入力フィールドと同じ長さを指定したことになります。  
無変換の場合、単純転送すること以外に処理方法の詳細を必要としないので、標準提供の「編集プログラム」を参照しません。

### 3. 機能

#### 3.4. 出力抑止変換機能

入力フィールド内のデータを読み捨てて出力しません。出力レコード長がその分、縮まります。

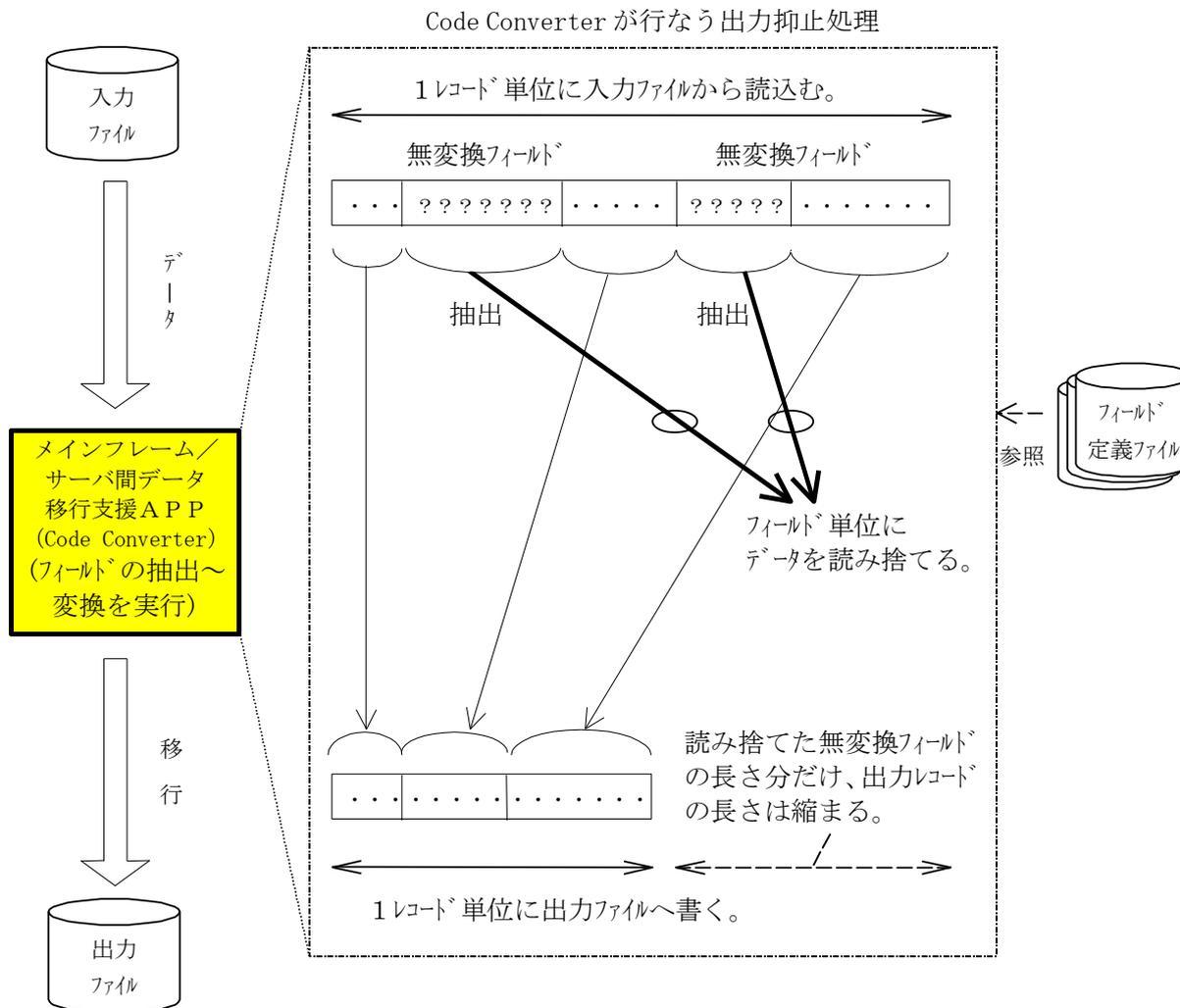


図 3.4.1 出力抑止変換概要

[説明]

1レコード単位に入力ファイルからデータを読み込み、その中から出力抑止フィールドの部分抽出し、読み捨てを行います。出力抑止フィールドの場合、出力フィールド長は、必ず 0 になります。

レコード内の無変換フィールドを抽出する位置は、ユーザ作成の「フィールド定義ファイル」中の指定値を参照することにより認識します。(詳細は、4.3.2項参照)

(注) 出力抑止の場合、出力フィールド長は指定しないでください。

出力抑止の場合、読み捨てを行なうこと以外の処理をしませんので、標準提供の「編集オプションファイル」を参照しません。

### 3. 機能

#### 3.5. 数値変換機能

数値文字列のみから成るフィールドデータに対して、コード変換を行います。対象とするコード体系は、次の通りです。

- ・メインフレーム側コード体系 → パック形式 または ゾーン形式
- ・オープンシステム側コード体系 → 数値文字

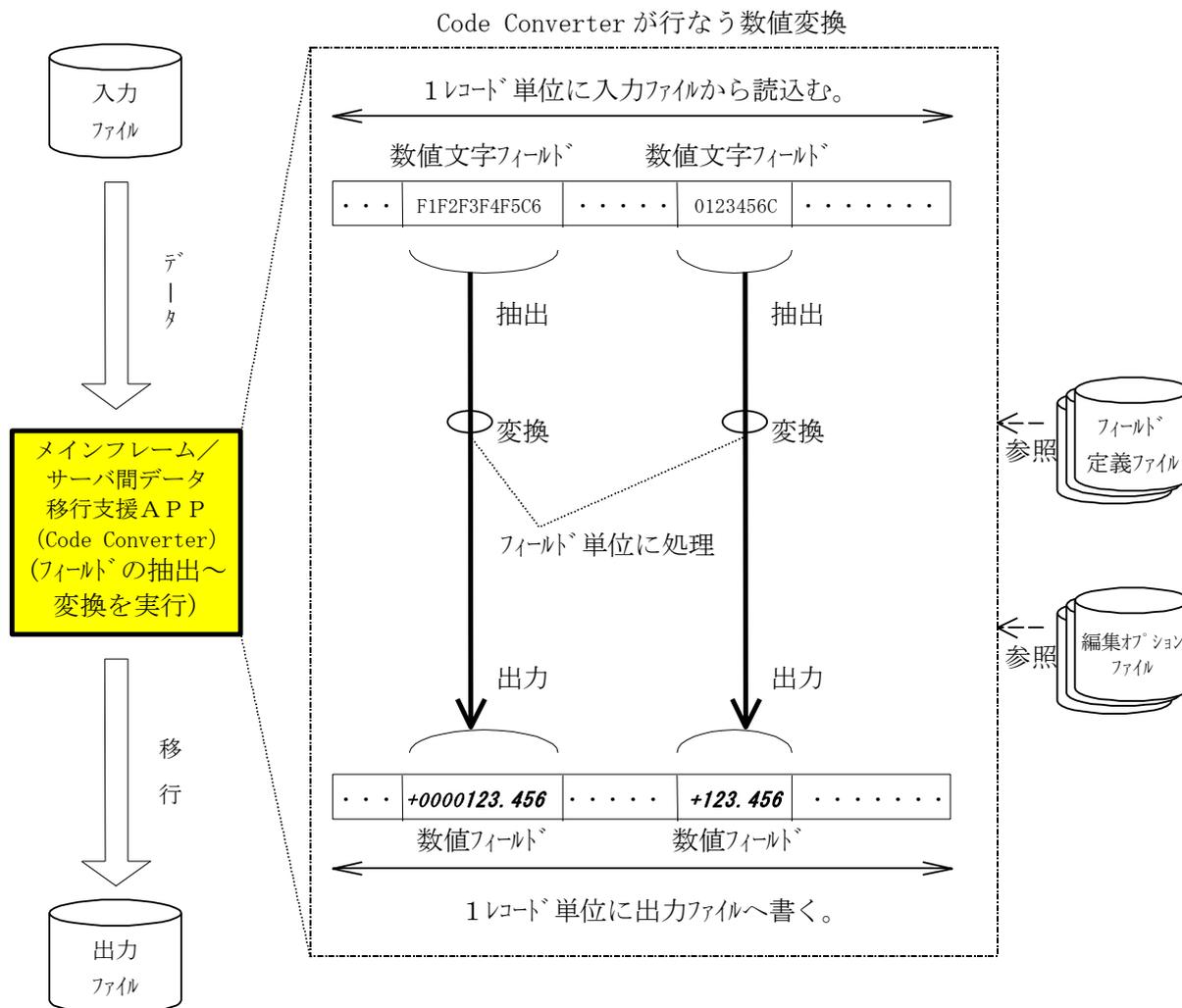


図 3.5.1 数値文字変換概要

[説明]

1レコード単位に入力ファイルからデータを読み、その中から変換対象となる数値文字フィールドの部分抽出し、指定の数値文字変換を行います。変換は1フィールド毎に実施、結果は1レコード単位に編集し、出力ファイルへ書き込みます。

レコード内の数値文字フィールドを抽出する位置や、出力するときのフィールド長は、ユーザ作成の「フィールド定義ファイル」中の指定値を参照することにより認識します。(詳細は、4.3.2項参照)

また、数値文字変換方法の詳細(符号位置や小数点桁数等)の指定は、標準提供の「編集オプションファイル」中のデフォルト指定パラメータにより制御しています。(詳細は、4.3.4項参照)

### 3. 機能

#### 3.6. エラーログ出力機能

本機能を使用するには、オプション製品の Code Converter-CODECHK が必要です。

##### (1) 未定義文字コードログ出力処理

コード変換する処理データに未定義文字コードがあった場合、編集オプションファイルに指定した未定義置換コードに変換する機能です。その時に未定義文字コードの置換とともに未定義文字コードがあるレコード番号、レコード中のオフセット及び未定義文字コードの内容をログファイルに出力します。

本機能は、編集オプションファイルのコード変換モード指定に TRAN\_CODECHK あるいは TRAN\_CODE\_UNDEFCHK が指定された時に動作します。

##### ◆ログファイル出力内容

```
*** CODE CONVERT LIST ***  DATE:日付け及び時刻
M/F DATASET=メインフレームデータセット名 ①
RECORD LENGTH=レコード長 ②
  REC# ③  OFFSET ④  TYPE ⑤  ERR-CODE ⑥
XXXXXXXXXX△△XXXXXXXX△△1B-CODE△△0Xxx
      ⋮
XXXXXXXXXX△△XXXXXXXX△△2B-CODE△△0Xxxxx

*** TOTAL RECORD      : xxxxxxxxxxxx⑦
*** ERROR RECORD COUNT : xxxxxxxxxxxx⑧
*** ERROR CODE COUNT  : xxxxxxxxxxxx⑨
<説明>
①：処理対象データセット名
②：レコード長
③：未定義コードを検出したレコード番号
④：未定義コードのレコードの先頭からのオフセット。レコード先頭を0とした値。
⑤：未定義コードが半角の場合『1B-CODE』、全角の場合『2B-CODE』を表示します。
⑥：未定義コードを表示します。
⑦：処理したレコード件数を表示します。
⑧：未定義コードがあったレコード数を表示します。
⑨：未定義コード件数を表示します。
```

##### ◆動作条件

下記条件を全て満たす場合に動作します。

- ・編集オプションファイルのコード変換モード指定に TRAN\_CODECHK あるいは TRAN\_CODE\_UNDEFCHK が指定されている。
- ・ファイル定義ファイルに漢字変換フィルタ（フィルタ属性が「KANJI」）が指定されている。

##### ◆ログファイル名

ログファイル名は、環境変数『CODE\_CONV\_ERR』で指定します。省略時は、UNIX の場合「/tmp/code\_conv\_err」、Windows の場合カレントディレクトリの「code\_conv\_err」を仮定します。

##### ◆ログファイル容量

ログファイル容量は1MB以内です。

### 3. 機能

#### (2) SO/SI ペアチェックログ 出力処理

メインフレーム環境からオープン環境へ文字コード変換(mto)する場合、SO/SI のペアチェックを行ないペアになっていなければ、ログファイルにSO/SI ログ情報を出力する機能です。

本機能は、編集オプションファイルのコード変換モード指定に TRAN\_SOSI\_PAIRCHK が指定された時に動作します。

#### ◆ログファイル出力内容

```
*** CODE CONVERT LIST ***  DATE:日付け及び時刻
```

```
M/F DATASET=メインフレームデータセット名 ①
```

```
RECORD LENGTH=レコード長 ②
```

```
REC# : xxxxxxxxxxxx③SO/SI pair unmatched
```

```
⋮
```

```
*** TOTAL RECORD      : xxxxxxxxxxxx ④
```

```
*** ERROR RECORD COUNT : xxxxxxxxxxxx ⑤
```

```
*** ERROR CODE COUNT   : xxxxxxxxxxxx ⑥
```

<説明>

①：処理対象データセット名

②：レコード長

③：SOSI ペア不一致を検出したレコード番号

④：処理したレコード件数を表示します。

⑤：SOSI ペア不一致があったレコード数を表示します。

⑥：SOSI ペア不一致件数を表示します。

#### ◆動作条件

下記条件を全て満たす場合に動作します。

- ・ mto 変換である。
- ・ 編集オプションファイルのコード変換モード指定に TRAN\_SOSI\_PAIRCHK が指定されている。
- ・ フィールド定義ファイルに漢字変換フィールド（フィールド属性が「KANJI」）が指定されている。

#### ◆ログファイル名

ログファイル名は、環境変数『CODE\_CONV\_ERR』で指定します。省略時は、UNIX の場合「/tmp/code\_conv\_err」、Windows の場合カレントディレクトリの「code\_conv\_err」を仮定します。

#### ◆ログファイル容量

ログファイル容量は1 MB以内です。

### 3. 機能

#### (3) オープンデータ切り捨てログ出力処理

オープン環境からメインフレーム環境へ文字コード変換(otm)する場合、オープンデータの切り捨てチェックを行ないデータの切り捨てが発生していれば、ログファイルにデータ切り捨てログ情報を出力する機能です。

本機能は、編集オプションファイルのコード変換モード指定に TRAN\_DATA\_SIZECHK が指定された時に動作します。

#### ◆ログファイル出力内容

```
*** KANJI CODE CONVERT LIST ***  DATE:日付け及び時刻  
M/F DATASET=メインフレームデータセット名①  
RECORD LENGTH=レコード長②  
*** TOTAL RECORD          : xxxxxxxxxx ④  
*** ERROR RECORD COUNT    : xxxxxxxxxx ⑤  
*** ERROR CODE COUNT      : xxxxxxxxxx ⑥  
REC#: xxxxxxxxxx Data was omitted. (IN_field_size:xxxxxxByte, Omission_size:xxxxxxByte)
```

#### <説明>

- ①: 処理対象データセット名
- ②: レコード長
- ③: データ切捨てを検出したレコード番号
- ④: 処理したレコード件数を表示する。
- ⑤: データ切捨てがあったレコード件数を表示する。
- ⑥: データ切捨て件数を表示する。

(注) ⑤、⑥の件数には、未定義コードチェック (TRAN\_CODE\_UNDEFCHK) 機能を同時に指定した場合は、その件数も含まれます。

#### ◆動作条件

下記条件を全て満たす場合に動作する。

- ・ otm 変換である。
- ・ 編集オプションファイルのコード変換モード指定に TRAN\_DATA\_SIZECHK が指定されている。

#### ◆ログファイル名

ログファイル名は、環境変数『CODE\_CONV\_ERR』で指定する。省略時は、UNIX の場合「/tmp/code\_conv\_err」、Windows の場合カレントディレクトリの「code\_conv\_err」を仮定する。

#### ◆ログファイル容量

ログファイル容量は1MB以内とする。

### 3. 機能

#### 3.7. コード変換に関わる編集諸機能

##### (1) 未定義文字コード処理 (半角)

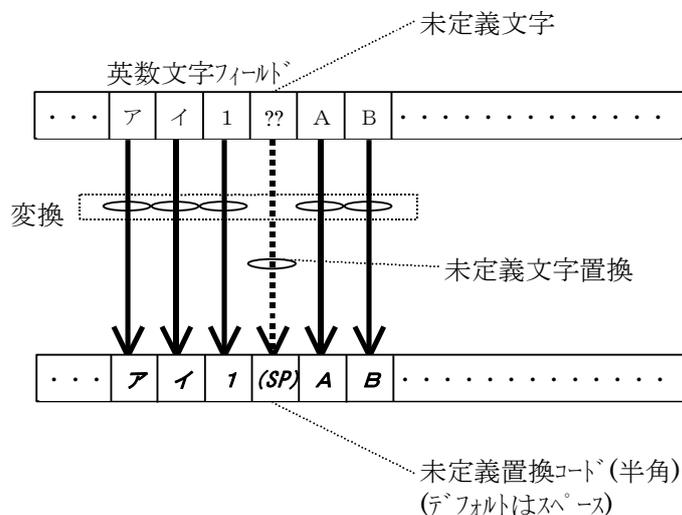


図 3.7.1 未定義文字コード処理 (半角) 概要

##### [説明]

英数文字変換の途中で現れた未定義文字(有効な変換先文字が定義されていない文字)は、未定義置換コード(半角スペース)で置換します。

編集オプションファイルのコード変換モード指定で TRAN\_MINIMUM を指定した場合、半角カタカナコード(EUCコードセット3)は未定義置換コードに置換されますのでご注意ください。

##### (2) 未定義文字コード処理 (全角)

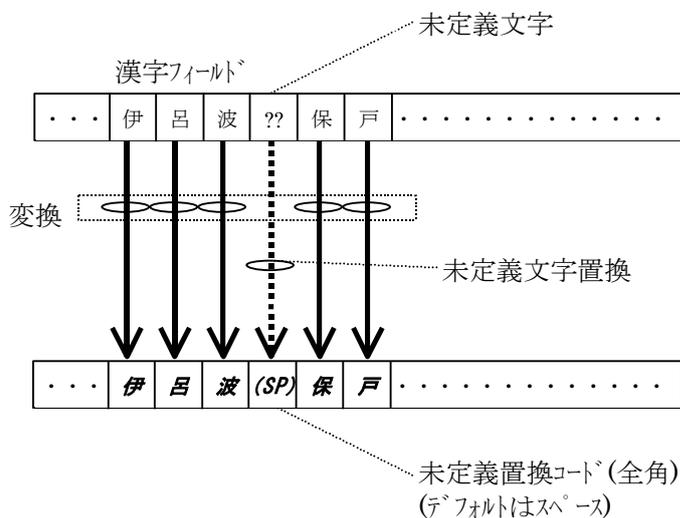


図 3.7.2 未定義文字コード処理 (全角) 概要

##### [説明]

漢字変換の途中で現れた未定義文字(有効な変換先文字が定義されていない文字)は、未定義置換コード(全角スペース)で置換します。

編集オプションファイルのコード変換モード指定で TRAN\_MINIMUM を指定した場合、変換後コードの EUCコードセット3は未定義置換コードに置換されますのでご注意ください。

なお、半角文字が混在する場合は、場所に応じて未定義置換コード(半角スペース/全角スペース)を使い分けます。

### 3. 機能

#### (3) シフトコード削除処理 (全半角混在時)

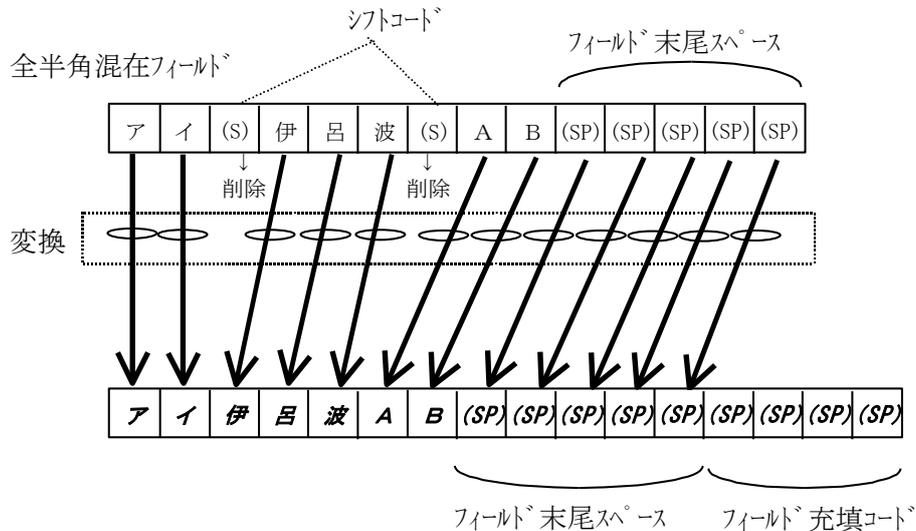


図 3.7.3 シフトコード削除処理 (全半角混在時) 概要

#### [説明]

シフトコードのあるコード体系からシフトコードのないコード体系へ漢字変換(全半角混在)する際に現れたシフトコード(全角/半角文字の境界を示す特殊コード)は削除し、前詰めで出力します。出力先の余ったエリアにはフィールド充填コード(全角/半角スペース)を充填します。

#### (4) シフトコード挿入処理 (全半角混在時)

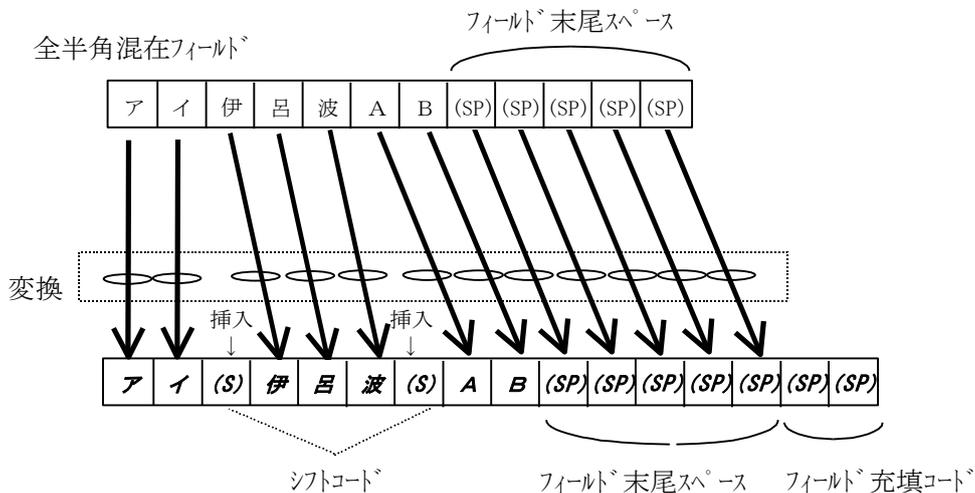


図 3.7.4 シフトコード挿入処理 (全半角混在時) 概要

#### [説明]

シフトコードのないコード体系からシフトコードのあるコード体系へ漢字変換(全半角混在)する際には、必要に応じてシフトコード(全角/半角文字の境界を示す特殊コード)を間に挿入しながら出力することができます。それでも出力先にエリアが余った場合には、フィールド充填コード(全角/半角スペース)を充填します。間にシフトコードを挿入した結果、出力先エリアが不足した場合には、入力データの末尾が切り捨てられますので、ご注意ください。

### 3. 機能

#### (5) シフトコード挿入抑止処理（全半角混在時）

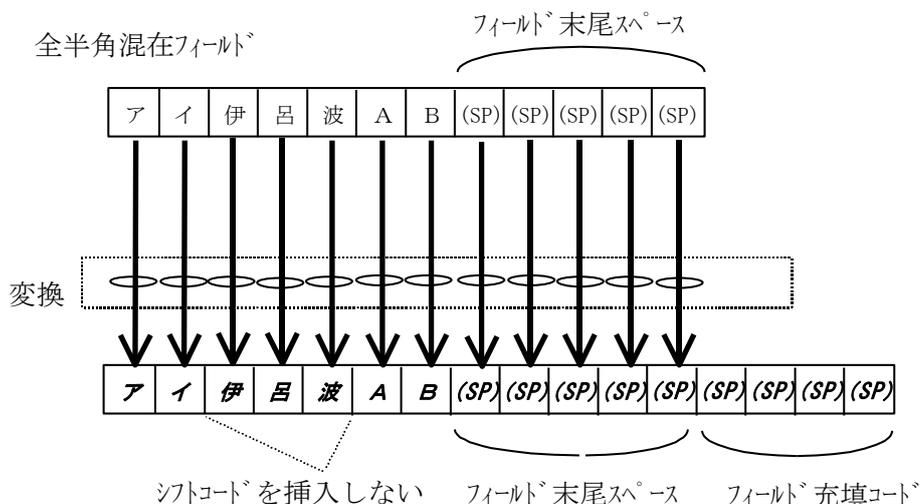


図 3.7.5 シフトコード挿入抑止処理（全半角混在時）概要

#### [説明]

シフトコードのないコード体系からシフトコードのあるコード体系へ漢字変換(全半角混在)する際でも、シフトコード(全角/半角文字の境界を示す特殊コード)を間に挿入せずに出力することが選択できます。

(本指定は特殊処理にあたるため、「編集オプションファイル」のコード変換モード指定に"TRAN\_SHIFT\_NO"を追加指定する必要があります。通常は、シフトコードを間に挿入して運用ください。)

#### (6) 出力先フィールドの方が短い場合の処理

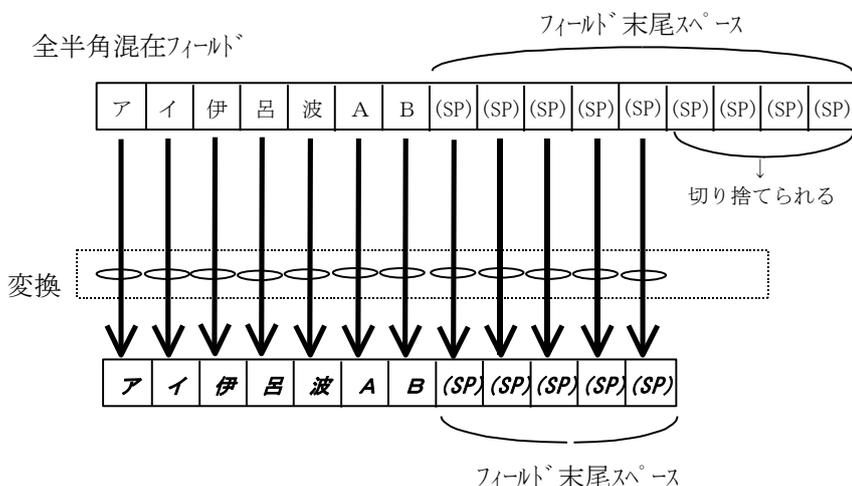


図 3.7.6 出力先フィールドの方が短い場合の処理概要

#### [説明]

「フィールド定義ファイル」で出力フィールド長を短く指定した場合や、間にシフトコードを挿入した結果、出力先エリアが不足するようになった場合には、入力データの末尾が切り捨てられますので、ご注意ください。

### 3. 機能

#### 3.8. EUCコード変換処理

本機能を使用するには、オプション製品のCode Converter-EUCが必要です。

EUCコード (JIS X0208-1990 : 注1) のコード体系を図 3.8.1に示します。

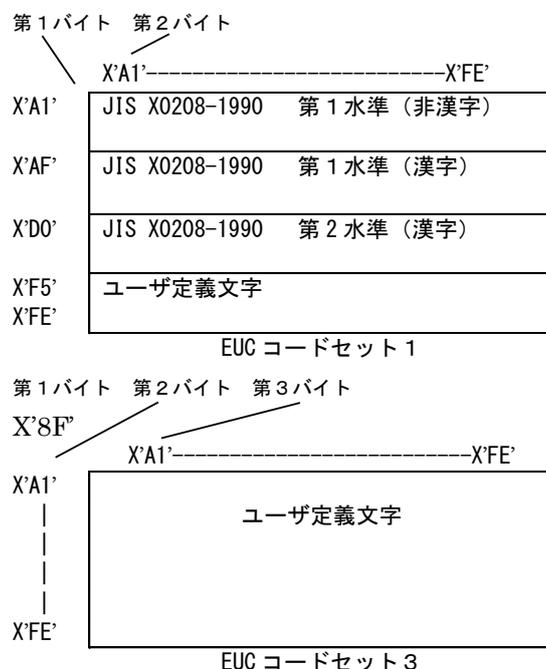


図 3.8.1 EUCコード体系

注1: EUCコードは、様々なUNIXワークステーションに搭載されていますが、メーカーや装置により対応している規格年度の違いや、メーカーが独自に搭載した文字があり必ずしも統一されていません。

Solaris2.6の日本語EUCコードの定義を表 3.8.1に示します。

表 3.8.1 日本語EUCコードの定義

文字セット	コードセット	EUCコード表現	文字コード長
JIS X 0201 ローマ字	0	(0xxxxxxx) 2	1バイト
JIS X 0208-1990 漢字/外字	1	(1xxxxxxx 1xxxxxxx) 2	2バイト
JIS X 0201 片仮名	2	(SS2 1xxxxxxx) 2	2バイト
JIS X 0212-1990 補助漢字/外字	3	(SS3 1xxxxxxx 1xxxxxxx) 2	3バイト

SS2: シングルシフト 2=(10001110) 2=(8E) 1 6

SS3: シングルシフト 3=(10001111) 2=(8F) 1 6

Solaris2.6の日本語EUCコードには、JIS X 0201 ローマ字、JIS X 0201 片仮名、JIS X 0208-1990 漢字、JIS X 0212 補助漢字が搭載されており、それぞれコードセット0~3と呼んでいます。コードセット0~3の文字コード長はそれぞれ異なります。

### 3. 機能

#### (1) EUCコード変換処理 (半角)

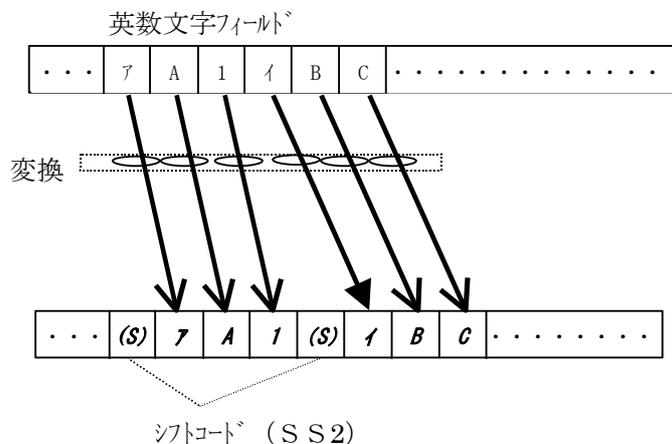


図 3.8.2 EUCコード変換処理 (半角) 概要

#### [説明]

IBM メインフレームコードから EUC コードへの変換時、英数文字フィールドに半角カナコード (EUC コードセット 1) が存在する場合、シフトコード (SS2) を付加して出力します。シフトコード (SS2) を付加した結果、出力先エリアが不足した場合には、入力データの末尾が切り捨てられますので、ご注意ください。

逆に、EUC コードから IBM メインフレームコードへの変換時、シフトコード (SS2) を削除して前詰めで出力します。出力先の余ったエリアにはフィールド充填コード (半角スペース) を充填します。

#### (2) EUCコード変換処理 (全角)

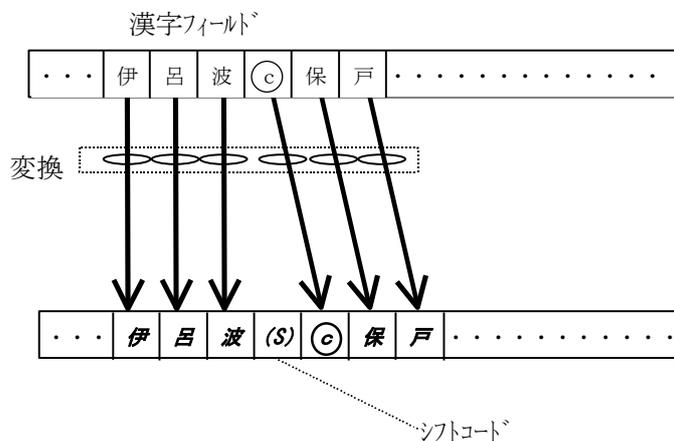


図 3.8.3 EUCコード変換処理 (全角) 概要

#### [説明]

IBM メインフレームコードから EUC コードへの変換時、漢字フィールドに EUC コードセット 3 が存在する場合、シフトコード (SS3) を付加して出力します。シフトコード (SS3) を付加した結果、出力先エリアが不足した場合には、入力データの末尾が切り捨てられますので、ご注意ください。

逆に、EUC コードから IBM メインフレームコードへの変換時、シフトコード (SS3) を削除して前詰めで出力します。出力先の余ったエリアにはフィールド充填コード (全角スペース) を充填します。

### 3. 機能

#### 3.9. Unicode コード変換処理

本機能を使用するには、オプション製品の Code Converter-UNICODE が必要です。

国際標準化機構 (ISO) は、世界中の主要な文字を一括して扱う多重言語文字セット規格 UCS (Universal multi-octet coded Character Set) を 1993 年に制定しました。これが ISO10646 と呼ばれるもので、4 バイトによる方式と 2 バイトによる方式の 2 通りがあります。

Unicode は、ISO10646 である UCS のサブセットで、UCS の群番号、面番号が 0 の面、すなわち上位 2 バイトが 0 である文字からなる面は、Basic Multilingual Plane (BMP) と呼ばれ、UCS のうち BMP だけを使用するのであれば、1 文字を 2 バイトで表現することができ、このようなサブセットが Unicode です。

コード変換テーブルとして標準提供する Unicode は、UCS-2 (2 バイト方式 Unicode、Windows Unicode) を対象としています。

サポートする JIS 規格年度は、JIS X 0208-1990 です。図 3.9.1 に Unicode の位置付けを示します。

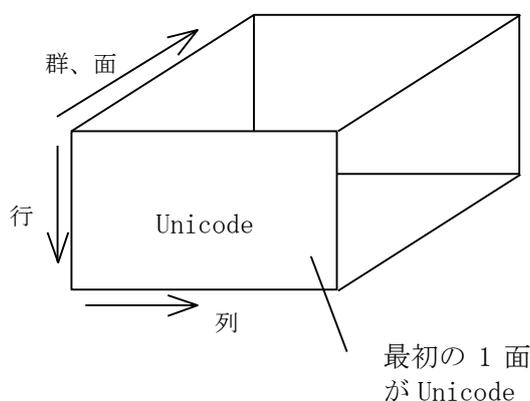


図 3.9.1 Unicode の位置付け

漢字は、この中に中国語、日本語、韓国語を合わせて配置しています。このため、JIS コードとの規則的な変換ルールはありません。

### 3. 機能

#### (1) Unicode変換処理 (半角)

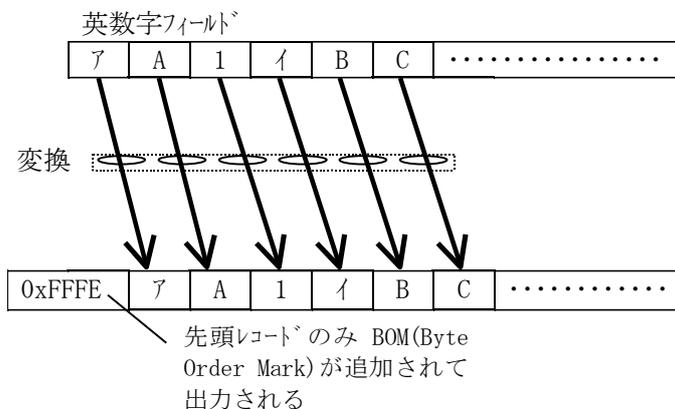


図 3.9.2 Unicode コード変換処理 (半角) 概要

#### [説明]

メインフレームコードから Unicode への変換時、先頭コードのみに BOM(Byte Order Mark)の 0xFFFE が追加されて出力されます。出力先エリアが不足した場合には、入力データの末尾が切り捨てられますので、ご注意ください。

逆に、Unicode からメインフレームコードへの変換時、BOM を削除して前詰めで出力します。出力先の余ったエリアにはフィールド充填コード(半角スペース)を充填します。

#### (2) Unicodeコード変換処理 (全角)

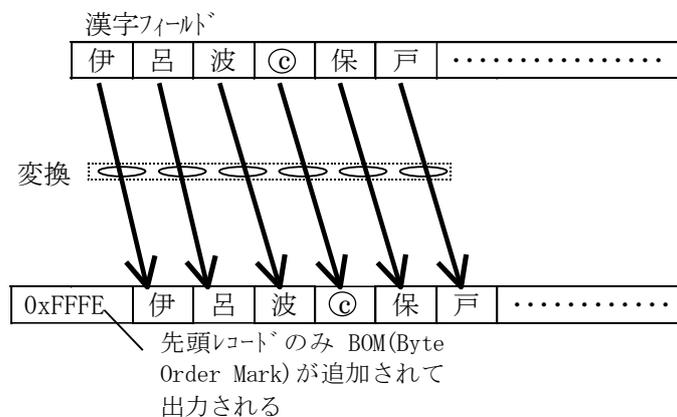


図 3.9.3 Unicode コード変換処理 (全角) 概要

#### [説明]

メインフレームコードから Unicode への変換時、先頭コードのみに BOM(Byte Order Mark)の 0xFFFE が追加されて出力されます。出力先エリアが不足した場合には、入力データの末尾が切り捨てられますので、ご注意ください。

逆に、Unicode からメインフレームコードへの変換時、BOM を削除して前詰めで出力します。出力先の余ったエリアにはフィールド充填コード(全角スペース)を充填します。

## 4. Code Converter の使い方

### 4. Code Converter の使い方

#### 4.1. 提供媒体およびインストール方法概説

Code Converter をお使い頂くためには、まずファイル変換ユーティリティを先にインストールしておき、その後で Code Converter 及び Code Converter オプション製品を追加インストールして頂くという手順が必要です。

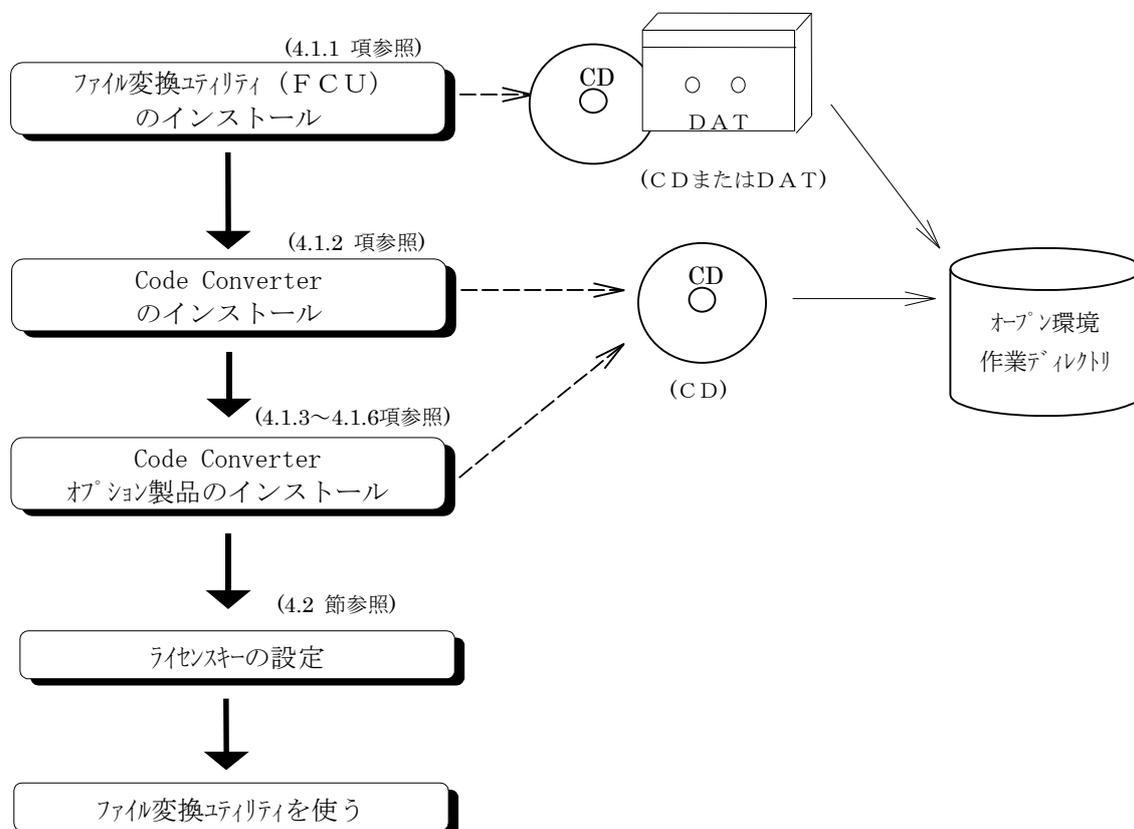


図 4.1.1 インストール方法概説

## 4. Code Converter の使い方

### 4.1.1. FCUのインストール及びアンインストール方法

FCU のインストール及びアンインストール方法については、「Cross-OS File Exchange Function ユーザーズガイド」を参照してください。

### 4.1.2. Code Converter のインストール/アンインストール

(1) UNIX 系の場合

#### [インストール方法]

- ① オープン環境の端末(UNIXサーバ)に root でログインしてください。
  - ② Code Converter の CD をドライブに入れてマウントしてください。
  - ③ カレントディレクトリをインストール先の作業ディレクトリへ移動し、次の tar コマンドを使って提供媒体に入っている全てのファイルを、提供媒体から作業ディレクトリへ複製してください。
- 4.1.6項(2)で示した Code Converter 用のファイルが、オープン環境の作業ディレクトリへ格納されます。

```
#tar xvf xxxxxx/libuoc.tar
```

… こうキーインしてください。

CD のデバイスファイル名 : 表 4.1.1 に示す CD のデバイスファイル名を指定してください。

表 4.1.1 CD のデバイスファイル名

項番	OS	媒体	デバイスファイル名
1	AIX 5.2 以降	CD	(マウントポイント)/PROGRAM/CONVERT/MAIN/AIX/
2	Solaris 8 以降	CD	(マウントポイント)/PROGRAM/CONVERT/MAIN/SOLARIS/
3	Red Hat Linux AS3.0 以降 SuSE Linux ES9 以降	CD	(マウントポイント)/PROGRAM/CONVERT/MAIN/LINUX/
4	RHEL (IA-64) ・ AS3.0 以降 ・ AS4.0(Update5) 以降 ・ AS5.1 以降 SuSE Linux (IA-64) ・ SLES9 以降	CD	(マウントポイント)/PROGRAM/CONVERT/MAIN64/LINUX/
5	HP-UX 11iV2(IA-64)	CD	(マウントポイント)/PROGRAM/CONVERT/MAIN64/HP_UX/IA
6	AIX 5.2 以降(64Bit)	CD	(マウントポイント)/PROGRAM/CONVERT/MAIN64/AIX/
7	Solaris 8 以降(64Bit)	CD	(マウントポイント)/PROGRAM/CONVERT/MAIN64/SOLARIS/

対応 OS の詳細については、弊社へ個別にご相談ください。

注 : CD のマウントの仕方によりファイル名/ディレクトリ名の大文字/小文字の見え方が異なり、またファイル名に ';' 等が付加される場合 (この場合は、ファイル名をシングルクォーテーション ( ' ) で囲む必要あり) があります。必ず、ls コマンドにて確認してください。

- ④ Code Converter の CD をドライブ から取り出してください。
- ⑤ /user/lib にある libuoc.\* を別名で保存してください。  
(注 : 拡張子は表 4.1.4 を見て、機種により変えてください。)
- ⑥ 共有ライブラリの移動。( /usr/lib/ に置く。これを行わないと fcu が動きません。)

```
# mv -f libuoc.* /usr/lib
```

… こうキーインしてください。

(注 : 拡張子は表 4.1.5 を見て、機種により変えてください。)

RHEL (IA-64) 対応の 64 ビットライブラリを使用する場合は、以下のコマンドを実行してください。

```
# mv -f libuoc64.so.1 /usr/lib/
```

… こうキーインしてください。

HP-UX(IA) 対応の 64 ビットライブラリを使用の場合は、以下のコマンドを実行してください。

## 4. Code Converter の使い方

```
# mv -f libuoc64.so /usr/lib/hpux64
```

… こうキーインしてください。

Solaris 対応の 64 ビットライブラリを使用の場合は、以下のコマンドを実行してください。

```
# mv -f libuoc64.so.1 /usr/lib/sparcv9
```

… こうキーインしてください。

- ⑦ バージョン情報ファイルの移動。( /usr/lib/ に置く。 これを行わないと fcU が動きません。)

```
# mv -f libuoc.ver /usr/lib
```

… こうキーインしてください。

- ⑧ ラテンキー入力ファイルの移動。( /usr/bin/ に置く。)

```
# mv -f ppkeyset /usr/bin
```

… こうキーインしてください。

64 ビットライブラリを使用の場合は、以下のコマンドを実行してください。

```
# mv -f ppkeyset64 /usr/bin
```

… こうキーインしてください。

- ⑨ 一度ログアウトし、ログインし直してください。

- ⑩ 以上で Code Converter のインストールは完了です。

### [アンインストール方法]

Code Converter 製品のアンインストールは、**FAL/FCU 製品をアンインストールする前に実施**します。

- ① オープン環境の端末(UNIX サーバ)に root でログインしてください。
- ② 4.1.6項(2)の内容で記載されているファイルを全て手動で削除してください。  
特に /usr/lib ディレクトリに入れた libuoc の実行ファイルとバージョン情報ファイルは、必ず削除してください。  
以下にその例を示します。

```
# cd /usr/lib
```

```
# rm libuoc.* libuoc.ver
```

注) 拡張子は表 4.1.4 を見て機種により変えてください。

- ③ インストール手順⑤で保存したファイルのファイル名を、元の libuoc.\* に戻してください。  
(これを行わないと fcU が動きません。拡張子は表 4.1.4 を見て機種により変えてください。)

### [バージョンアップ/ダウン方法]

- ① オープン環境の端末(UNIX サーバ)に root でログインしてください。
- ② Code Converter 提供の関連ファイルをユーザが変更したファイルがあれば退避します。
  - ・ 編集オプションファイル (EditOpFile)
- ③ Code Converter をインストールします。
- ④ ②で退避した関連ファイルを回復します。
- ⑤ ラテンキーコードを再入力します。
- ⑥ 一度ログアウトし、ログインし直してください。
- ⑦ 以上で Code Converter のバージョンアップ/ダウンは完了です。

## 4. Code Converter の使い方

(2) Windows 系の場合

### [インストール方法]

- ① PC の Windows を立ち上げて、administrator で開いてください。
- ② Code Converter の CD をドライブに入れてください。
- ③ エクスプローラー等で Code Converter の CD 中下記ディレクトリ内の "setup.exe" をダブルクリックすると、セットアップが始まります。  
ディレクトリ：(CDドライブ) %PROGRAM%CONVERT%MAIN%WIN%DISK1  
画面の案内に従って、ユーザーの情報/インストール先/プログラムフォルダ等を指定・選択する操作を行なってください。インストール先は、FCU と同じディレクトリを指定してください。
- ④ 最後に「InstallShield ウィザードの完了」画面が表示されたら、「完了」ボタンをクリックして Code Converter のインストールは完了です。

### [アンインストール方法]

Code Converter 製品のアンインストールは、**FAL/FCU 製品をアンインストールする前に実施**します。

- ① PC の Windows を立ち上げて、administrator で開いてください。
- ② 「スタート」をクリックし、「設定」→「コントロールパネル」をクリックしてください。
- ③ 「アプリケーションの追加と削除」をダブルクリックし、「プログラムの変更と削除」のソフトウェア一覧より「Code Converter」をクリックし、「変更/削除」ボタンをクリックするとアンインストールが開始します。  
以降は画面の案内に従って、操作してください。
- ④ 最後に「メンテナンスの完了」画面が表示されたら、「完了」ボタンをクリックして Code Converter のアンインストールは完了です。

### [バージョンアップ/ダウン方法]

- ① PC の Windows を立ち上げて、administrator で開いてください。
- ② Code Converter 提供の関連ファイルをユーザーが変更したファイルがあれば回避します。
  - ・ 編集オプションファイル (EditOpFile)
- ③ Code Converter を提供媒体からインストールします。
- ④ ②で回避した関連ファイルを回復します。
- ⑤ ライセンスコードを再入力します。
- ⑥ 以上で Code Converter のバージョンアップ/ダウンは完了です。

### 4.1.3. Code Converter-EUC のインストール/アンインストール

(1) UNIX 系の場合

### [インストール方法]

- ① オープン環境の端末(UNIXサーバ)に root でログインしてください。
- ② Code Converter-EUC の CD をドライブに入れてマウントしてください。
- ③ 次に、カレントディレクトリをインストール先の作業ディレクトリに移動し、次の tar コマンドを使って提供媒体に入っている全てのファイルを、提供媒体から作業ディレクトリへ複製してください。  
すると、4.1.6項(3)で示した Code Converter-EUC のファイルが、オープン環境の作業ディレクトリへ格納されます。

```
#tar xvf xxxxxxx/uoceuc.tar
```

… こうキーインしてください。

CD のディレクトリ名：表 4.1.2 に示す CD のディレクトリ名を指定してください。

## 4. Code Converter の使い方

表 4.1.2 CD のデバイスファイル名

項番	OS	媒体	デバイスファイル名
1	AIX 5.2 以降	CD	(マウントポイント)/PROGRAM/CONVERT/EUC/AIX/
2	Solaris 8 以降	CD	(マウントポイント)/PROGRAM/CONVERT/EUC/SOLARIS/
3	Red Hat Linux 7.2 以降 SuSE Linux ES9 以降	CD	(マウントポイント)/PROGRAM/CONVERT/EUC/LINUX/
4	RHEL (IA-64) ・AS3.0 以降 ・AS4.0(Update5)以降 ・AS5.1 以降 SuSE Linux (IA-64) ・SLES9 以降	CD	(マウントポイント)/PROGRAM/CONVERT/EUC64/LINUX/
5	HP-UX 11iV2 (IA-64)	CD	(マウントポイント)/PROGRAM/CONVERT/EUC64/HP_UX/
6	AIX 5.2 以降(64Bit)	CD	(マウントポイント)/PROGRAM/CONVERT/EUC64/AIX/
7	Solaris 8 以降(64Bit)	CD	(マウントポイント)/PROGRAM/CONVERT/EUC64/SOLARIS/

対応 OS の詳細については、弊社へ個別にご相談ください。

注：CD のマウントの仕方によりファイル名/ディレクトリ名の大文字/小文字の見え方が異なり、またファイル名に ';' 等が付加される場合（この場合は、ファイル名をシングルクォーテーション（' '）で囲む必要あり）があります。必ず、ls コマンドにて確認してください。

- ④ Code Converter-EUC の CD をドライブから取り出してください。
- ⑤ 一度ログアウトし、ログインし直してください。
- ⑥ 以上で Code Converter-EUC のインストールは完了です。

### [アンインストール方法]

Code Converter 製品のアンインストールは、**FAL/FCU 製品をアンインストールする前に実施**します。

- ① オープン環境の端末 (UNIX サーバ) に root でログインしてください。
- ② 4.1.6項(3)の内容で記載されているファイルを全て手で削除してください。

### [バージョンアップ/ダウン方法]

- ① オープン環境の端末 (UNIX サーバ) に root でログインしてください。
- ② Code Converter-EUC 提供の関連ファイルをユーザが変更したファイルがあれば退避します。
  - ・ 編集オプションファイル (EditOpEUC)
- ③ Code Converter-EUC をインストールします。
- ④ ②で退避した関連ファイルを回復します。
- ⑤ 一度ログアウトし、ログインし直してください。
- ⑥ 以上で Code Converter-EUC のバージョンアップ/ダウンは完了です。

### (2) Windows 系の場合

#### [インストール方法]

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② Code Converter-EUC の CD をドライブに入れてください。
- ③ エクスプローラ等で Code Converter-EUC の CD 中下記ディレクトリの "setup.exe" をダブルクリックすると、セットアップが始まります。
  - ディレクトリ: (CD ドライブ) %PROGRAM%CONVERT\EUC%WIN%DISK1
 画面の案内に従って、ユーザの情報/インストール先/プログラムフォルダ等を指定・選択する操作を行なってください。インストール先は、Code Converter と同じディレクトリを指定してください。

## 4. Code Converter の使い方

- ④ 最後に「InstallShield ウィザード」の完了」画面が表示されたら、「完了」ボタンをクリックして Code Converter-EUC のインストールは完了です。

### [アンインストール方法]

Code Converter 製品のアンインストールは、**FAL/FCU 製品をアンインストールする前に実施**します。

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② 「スタート」をクリックし、「設定」→「コントロールパネル」をクリックしてください。
- ③ 「アプリケーションの追加と削除」をダブルクリックし、「プログラムの変更と削除」のソフトウェア一覧より「Code Converter-EUC」をクリックし、「変更/削除」ボタンをクリックするとアンインストールが開始します。  
以降は画面の案内に従って、操作してください。
- ④ 最後に「メンテナンスの完了」画面が表示されたら、「完了」ボタンをクリックして Code Converter-EUC のアンインストールは完了です。

### [バージョンアップ/ダウン方法]

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② Code Converter-EUC 提供の関連ファイルをユーザが変更したファイルがあれば退避します。
  - ・ 編集オプションファイル (EditOpEUC)
- ③ Code Converter-EUC を提供媒体からインストールします。
- ④ ②で退避した関連ファイルを回復します。
- ⑤ 以上で Code Converter-EUC のバージョンアップ/ダウンは完了です。

### 4.1.4. Code Converter-CODECHK のインストール/アンインストール

(1)UNIX 系の場合

#### [インストール方法]

- ① オープン環境の端末(UNIX サーバ)に root でログインしてください。
- ② Code Converter-CODECHK の CD をドライブに入れてマウントしてください。
- ③ 次に、カレントディレクトリをインストール先の作業ディレクトリに移動し、次の tar コマンドを使って提供媒体に入っている全てのファイルを、提供媒体から作業ディレクトリへ複製してください。  
すると、4.1.6項(4)で示した Code Converter-CODECHK のファイルが、オープン環境の作業ディレクトリへ格納されます。

```
#tar xvf xxxxxx/uocchk.tar
```

… こうキーインしてください。

CD のデバイスファイル名 : 表 4.1.3 に示す CD のデバイスファイル名を指定してください。

## 4. Code Converter の使い方

表 4.1.3 CD のデバイスファイル名

項番	OS	媒体	デバイスファイル名
1	AIX 5.2 以降	CD	(マウントポイント)/PROGRAM/CONVERT/CODECHK/AIX/
2	Solaris 8 以降	CD	(マウントポイント)/PROGRAM/CONVERT/CODECHK/SOLARIS/
3	Red Hat Linux 7.2 以降 SuSE Linux ES9 以降	CD	(マウントポイント)/PROGRAM/CONVERT/CODECHK/LINUX/
4	RHEL (IA-64) ・AS3.0 以降 ・AS4.0(Update5) 以降 ・AS5.1 以降 SuSE Linux (IA-64) ・SLES9 以降	CD	(マウントポイント)/PROGRAM/CONVERT/CODECHK64/LINUX/
5	HP-UX 11iV2(IA-64)	CD	(マウントポイント)/PROGRAM/CONVERT/CODECHK64/HP_UX/
6	AIX 5.2 以降(64Bit)	CD	(マウントポイント)/PROGRAM/CONVERT/CODECHK64/AIX/
7	Solaris 8 以降(64Bit)	CD	(マウントポイント)/PROGRAM/CONVERT/CODECHK64/SOLARIS/

対応 OS の詳細については、弊社へ個別にご相談ください。

注：CD のマウントの仕方によりファイル名/ディレクトリ名の大文字/小文字の見え方が異なり、またファイル名に `;` 等が付加される場合（この場合は、ファイル名をシングルクォーテーション（`'`）で囲む必要あり）があります。必ず、`ls` コマンドにて確認してください。

- ④ Code Converter-CODECHK の CD をドライブから取り出してください。
- ⑤ オプション機能プロジェクトの移動（/usr/lib に置きます。これを行わないと CODECHK オプション機能が動きません）。

```
#mv -f libuoc.sb1 /usr/lib
```

… こうキーインしてください。

- ⑥ 一度ログアウトし、ログイン直してください。
- ⑦ 以上で Code Converter-CODECHK のインストールは完了です。

### [アンインストール方法]

Code Converter 製品のアンインストールは、**FAL/FCU 製品をアンインストールする前に実施**します。

- ① オープン環境の端末(UNIXサーバ)に root でログインしてください。
- ② 4.1.6項(4)の内容で記載されているファイルを全て手動で削除してください。

### [バージョンアップ/ダウン方法]

- ① オープン環境の端末(UNIXサーバ)に root でログインしてください。
- ② Code Converter-CODECHK をインストールします。
- ③ 一度ログアウトし、ログイン直してください。
- ④ 以上で Code Converter-CODECHK のバージョンアップ/ダウンは完了です。

### (2) Windows 系の場合

#### [インストール方法]

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② Code Converter-CODECHK の CD をドライブに入れてください。
- ③ エクスプローラー等で Code Converter-CODECHK の CD 中下記ディレクトリの "setup.exe" をダブルクリックすると、セットアップが始まります。

ディレクトリ：(CD ドライブ) %PROGRAM%CONVERT%CODECHK%WIN%DISK1

画面の案内に従って、ユーザーの情報/インストール先/プログラムフォルダ等を指定・選択する操作を行なってくだ

## 4. Code Converter の使い方

さい。インストール先は、Code Converter と同じディレクトリを指定してください。

- ④ 最後に「InstallShield ウィザード」の完了」画面が表示されたら、「完了」ボタンをクリックして Code Converter-CODECHK のインストールは完了です。

### [アンインストール方法]

Code Converter 製品のアンインストールは、**FAL/FCU 製品をアンインストールする前に実施**します。

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② 「スタート」をクリックし、「設定」→「コントロールパネル」をクリックしてください。
- ③ 「アプリケーションの追加と削除」をダブルクリックし、「プログラムの変更と削除」のソフトウェア一覧より「Code Converter-CODECHK」をクリックし、「変更/削除」ボタンをクリックするとアンインストールが開始します。以降は画面の案内に従って、操作してください。
- ④ 最後に「メンテナンスの完了」画面が表示されたら、「完了」ボタンをクリックして Code Converter-CODECHK のアンインストールは完了です。

### [バージョンアップ/ダウン方法]

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② Code Converter-CODECHK を提供媒体からインストールします。
- ③ 以上で Code Converter-CODECHK のバージョンアップ/ダウンは完了です。

### 4.1.5. Code Converter-UNICODE のインストール/アンインストール

(1) Windows 系の場合

#### [インストール方法]

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② 次に、Code Converter-UNICODE の CD をドライブに入れてください。
- ③ エクスプローラ等で Code Converter-UNICODE の CD 中下記ディレクトリの "setup.exe" をダブルクリックすると、セットアップが始まります。  
ディレクトリ: (CD ドライブ) %PROGRAM%CONVERT%UNICODE%WIN%DISK1  
画面の案内に従って、ユーザーの情報/インストール先/プログラムフォルダ等を指定・選択する操作を行なってください。インストール先は、Code Converter と同じディレクトリを指定してください。
- ④ 最後に「InstallShield ウィザード」の完了」画面が表示されたら、「完了」ボタンをクリックして Code Converter-UNICODE のインストールは完了です。

#### [アンインストール方法]

Code Converter 製品のアンインストールは、**FAL/FCU 製品をアンインストールする前に実施**します。

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② 「スタート」をクリックし、「設定」→「コントロールパネル」をクリックしてください。
- ③ 「アプリケーションの追加と削除」をダブルクリックし、「プログラムの変更と削除」のソフトウェア一覧より「Code Converter-UNICODE」をクリックし、「変更/削除」ボタンをクリックするとアンインストールが開始します。以降は画面の案内に従って、操作してください。
- ④ 最後に「メンテナンスの完了」画面が表示されたら、「完了」ボタンをクリックして Code Converter-UNICODE のアンインストールは完了です。

#### [バージョンアップ/ダウン方法]

- ① PC の Windows を立ち上げて、administrator でログインしてください。
- ② Code Converter-UNICODE 提供の関連ファイルをユーザーが変更したファイルがあれば回避します。
  - ・ 編集オプションファイル (EditOpUNI)
- ③ Code Converter-UNICODE を提供媒体からインストールします。
- ④ ②で回避した関連ファイルを回復します。

## 4. Code Converter の使い方

⑤ 以上で Code Converter-UNICODE のバージョンアップ/ダウンは完了です。

### 4.1.6. 提供媒体

#### [UNIX 系の場合]

(1) ファイル変換ユーティリティ (FCU) の提供媒体

FCU の提供媒体については、「Cross-OS File Exchange Function ユーザーズガイド」を参照してください。

(2) Code Converter の提供媒体

Code Converter は、CD に tar 形式で次のファイル群を格納してご提供致します。

[提供媒体の内容]

#### ■ 32 ビット対応の提供媒体

項番	ファイル名	内 容	インストール場所	
1	libuoc.* (注1)	UOC 漢字変換機能オブジェクト (FAL/FCU 提供ダミーオブジェクトと置換する)	/usr/lib	
2	libuoc.ver (注6)	バージョン情報ファイル		
3	ppkeyset	ライセンスコードを入力する実行ファイル	/usr/bin	
4	codetran.ini	漢字変換ルーチンが使用する ini ファイル	FAL/FCU 起動ディレクトリ	
5	EBCASCII.tbl	EBCDIC→ASCII 変換テーブル		mtm 方向標準変換 テーブル
6	IBMSJIS.tbl	IBM→シフト JIS 変換テーブル		
7	JEFSJIS.tbl	JEF→シフト JIS 変換テーブル		
8	KS78SJIS.tbl	KEIS78→シフト JIS 変換テーブル		
9	KS90SJIS.tbl	KEIS83/90→シフト JIS 変換テーブル		
10	ASCIIEBC.tbl	ASCII→EBCDIC 変換テーブル		otm 方向標準変換 テーブル
11	SJISIBM.tbl	シフト JIS→IBM 変換テーブル		
12	SJISJEF.tbl	シフト JIS→JEF 変換テーブル		
13	SJISKS78.tbl	シフト JIS→KEIS78 変換テーブル		
14	SJISKS90.tbl	シフト JIS→KEIS83/90 変換テーブル		
15	EditOpFile	編集オプションファイル		
16	ccnv (注2)	コード変換テーブルをカスタマイズする ccnv コマンドの実行ファイル		
17	libuocccnv.ver	ccnv のバージョン情報ファイル		
18	uoc.c (注3)	標準提供ソース	Code Converter を使わず ユーザ独自の UOC 関数を作 成するためのファイル	
19	uocerror.h (注3)	標準提供ヘッダファイル		
20	libuoc.mak (注3)	メークファイル		
21	uoc.exp (注3)(注4)	モデル定義ファイル		

(注1) 拡張子 ".\*" は、使用するマシンによって表 4.1.4 に示すように異なります。

(注2) 項番 17 の ccnv は、EBCDIC/EBCDIK にあって JIS8 にない文字や、外字を追加登録したい場合など、変換テーブルのカスタマイズが必要なときにご使用ください。

(注3) 項番 18~21 の 4 つのファイルは、Code Converter をご使用頂く際には、必要ありません。

(注4) uoc.exp は AIX 版のみ提供されています。

(注5) 対応 OS の詳細については弊社へ個別にご相談ください。

(注6) libuoc.ver は、/usr/lib/に置いてください。

## 4. Code Converter の使い方

### ■ 6 4 ビット対応の提供媒体

項番	ファイル名	内 容		インストール場所
1	Libuoc64.* (注1)	UOC 漢字変換機能オブジェクト (FAL/FCU 提供ダミーオブジェクトと置換する)		/usr/lib (注7)
2	libuoc.ver (注6)	バージョン情報ファイル		/usr/lib
3	ppkeyset64	ライセンスコードを入力する実行ファイル		/usr/bin
4	codetran.ini	漢字変換ルーチンが使用する ini ファイル		FAL/FCU 起動ディレクトリ
5	EBCASCII.tbl	EBCDIC→ASCII 変換テーブル	mto 方向標準変換 テーブル	
6	IBMSJIS.tbl	IBM→シフト JIS 変換テーブル		
7	JEFSJIS.tbl	JEF→シフト JIS 変換テーブル		
8	KS78SJIS.tbl	KEIS78→シフト JIS 変換テーブル		
9	KS90SJIS.tbl	KEIS83/90→シフト JIS 変換テーブル		
10	ASCIIIBC.tbl	ASCII→EBCDIC 変換テーブル	otm 方向標準変換 テーブル	
11	SJISIBM.tbl	シフト JIS→IBM 変換テーブル		
12	SJISJEF.tbl	シフト JIS→JEF 変換テーブル		
13	SJISKS78.tbl	シフト JIS→KEIS78 変換テーブル		
14	SJISKS90.tbl	シフト JIS→KEIS83/90 変換テーブル		
15	EditOpFile	編集オプションファイル		
16	ccnv64 (注2)	コード変換テーブルをカスタマイズする ccnv コマンドの実行ファイル		
17	libuocccnv.ver	ccnv64 のバージョン情報ファイル		
18	uoc.c (注3)	標準提供ソース	Code Converter を使わず 独自に UOC 関数を作成す るためのファイル	
19	uocerror.h (注3)	標準提供ヘッダファイル		
20	libuoc64.mak (注3)	メークファイル		
21	uoc.exp (注3)(注4)	モジュール定義ファイル		

(注1) 拡張子 ".\*" は、使用するマシンによって表 4.1.4 に示すように異なります。

(注2) 項番 17 の ccnv64 は、EBCDIC/EBCDIC にあって JIS8 にはない文字や、外字を追加登録したい場合など、変換テーブルのカスタマイズが必要なきにご使用ください。

(注3) 項番 18～21 の 4 つのファイルは、Code Converter をご使用頂く際には、必要ありません。

(注4) uoc.exp は AIX 版のみ提供されています。

(注5) 対応 OS の詳細については弊社へ個別にご相談ください。

(注6) libuoc.ver は、/usr/lib/に置いてください。

(注7) 64 ビットライブラリのインストール場所は、以下の通りです。

- Solaris : /usr/lib/sparcv9
- HP-UX (IA) : /usr/lib/hpux64
- AIX : /usr/lib
- Linux : /usr/lib

## 4. Code Converter の使い方

表 4.1.4 使用するマシンによる拡張子の違い

項番	区分	メーカー	OS	拡張子
1	UNIX サーバ	HP	HP-UX 11iV2(IA-64)	.so
2		IBM	AIX 5.2以降	.a
3		SUN	Solaris 8以降	.so.1
4		Red Hat	Linux AS3.0以降	.so.1
5		SuSE	Linux ES9以降	.so.1

### (3) Code Converter-EUC の提供媒体

Code Converter-EUC は、CD に tar 形式で次のファイル群を格納してご提供致します。

[提供媒体の内容]

項番	ファイル名	内 容		インストール場所
1	IBMEUC.tbl	IBM→EUC 変換テーブル	mto 方向標準変換テーブル	FAL/FCU 起動ディレクトリ
2	KS78EUC.tbl	KEIS78→EUC 変換テーブル		
3	KS90EUC.tbl	KEIS83/90→EUC 変換テーブル		
4	EUCIBM.tbl	EUC→IBM 変換テーブル	otm 方向標準変換テーブル	
5	EUCKS78.tbl	EUC→KEIS78 変換テーブル		
6	EUCKS90.tbl	EUC→KEIS83/90 変換テーブル		
7	EditOpEUC	EUC 用編集オプションファイル		
8	libuoeuc.ver	バージョン情報ファイル		

### (4) Code Converter-CODECHK の提供媒体

Code Converter-CODECHK は、CD に tar 形式で次のファイル群を格納してご提供致します。

[提供媒体の内容]

項番	ファイル名	内 容	インストール場所
1	libuoc.sbl	エラーログ出力機能プロジェクト	/usr/lib (ディレクトリ固定)
2	libuocchk.ver	バージョン情報ファイル	FAL/FCU 起動ディレクトリ

## 4. Code Converter の使い方

### [Windows の場合]

#### (1) ファイル変換ユーティリティ (FCU)の提供媒体

FCUの提供媒体については、「Cross-OS File Exchange Function ユーザーガイド」を参照してください。

#### (2) Code Converter の提供媒体

Code Converter は、CD に Windows 用のインストール専用形式 (Install Shield 形式) で、次のファイル群を格納してご提供致します。

#### [提供媒体の内容]

項番	ファイル名	内 容		インストール場所
1	uoc.dll	UOC 漢字変換機能オブジェクト (FAL/FCU 提供ダミーオブジェクトと置換する)		FAL/FCU 起動フォルダ (C:\FCU)
2	codetran.ini	漢字変換ルーチンが使用する ini ファイル		
3	EBCASCII.tbl	EBCDIC→ASCII 変換テーブル	mto 方向標準変換 テーブル	
4	IBMSJIS.tbl	IBM→シフト JIS 変換テーブル		
5	JEFSJIS.tbl	JEF→シフト JIS 変換テーブル		
6	KS78SJIS.tbl	KEIS78→シフト JIS 変換テーブル		
7	KS90SJIS.tbl	KEIS83/90→シフト JIS 変換テーブル		
8	ASCIIEBC.tbl	ASCII→EBCDIC 変換テーブル	otm 方向標準変換 テーブル	
9	SJISIBM.tbl	シフト JIS→IBM 変換テーブル		
10	SJISJEF.tbl	シフト JIS→JEF 変換テーブル		
11	SJISKS78.tbl	シフト JIS→KEIS78 変換テーブル		
12	SJISKS90.tbl	シフト JIS→KEIS83/90 変換テーブル		
13	EditOpFile	編集オプションファイル		
14	ccnv.exe (注1)	コード変換ファイルをカスタマイズする ccnv コマンドの実行ファイル		
15	ppkeyset.exe	ライセンスコードを入力する実行ファイル		
16	libuoccnv.ver	ccnv.exe のバージョン情報ファイル		
17	libuoc.ver (注3)	バージョン情報ファイル		
18	uoc.c (注2)	標準提供ソース	Code Converter を使わず ユーザ独自の UOC 関数を作 成するためのファイル	FAL/FCU フォルダ→UOC フォルダ - (C:\FCU\UOC)
19	uocerror.h (注2)	標準提供ヘッダファイル		
20	Uoc.mak (注2)	メイクファイル		
21	Uoc.dsp (注2)	プロジェクトファイル		
22	Uoc.def (注2)	def ファイル		

(注1) 項番 14 の ccnv は、EBCDIC/EBCDIK にあって JIS8 がない文字や、外字を追加登録したい場合など、変換テーブルのカスタマイズが必要なおきにご使用ください。

(注2) 項番 18～22 の5つのファイルは、Code Converter をご使用頂く際には、必要ありません。これらのファイルは、インストール先の作業ディレクトリの下にできる"Uoc"というサブディレクトリに、まとめて格納されます。

(注3) Windows システムディレクトリに格納されます。(例)C:\WINDOWS。

#### 4. Code Converter の使い方

##### (3) Code Converter-EUC の提供媒体

Code Converter-EUC は、CD に Windows 用のインストール専用形式 (Install Shield 形式) で、次のファイル群を格納してご提供致します。

[提供媒体の内容]

項番	ファイル名	内 容		インストール場所
1	IBMEUC.tbl	IBM→EUC 変換テーブル	mto 方向標準変換テーブル	FAL/FCU 起動フォルダ (C:\FCU)
2	KS78EUC.tbl	KEIS78→EUC 変換テーブル		
3	KS90EUC.tbl	KEIS83/90→EUC 変換テーブル		
4	EUCIBM.tbl	EUC→IBM 変換テーブル	otm 方向標準変換テーブル	
5	EUCKS78.tbl	EUC→KEIS78 変換テーブル		
6	EUCKS90.tbl	EUC→KEIS83/90 変換テーブル		
7	EditOpEUC	EUC 用編集オプションファイル		
8	libuocuc.ver	バージョン情報ファイル		

##### (4) Code Converter-CODECHK の提供媒体

Code Converter-CODECHK は、CD に Windows 用のインストール専用形式 (Install Shield 形式) で、次のファイル群を格納してご提供致します。

[提供媒体の内容]

項番	ファイル名	内 容	インストール場所
1	libuoc.sbl	エラーログ出力機能プロジェクト	FAL/FCU 起動フォルダ (C:\FCU)
2	libuocchk.ver	バージョン情報ファイル	

##### (5) Code Converter-UNICODE の提供媒体

Code Converter-UNICODE は、CD に Windows 用のインストール専用形式 (Install Shield 形式) で、次のファイル群を格納してご提供致します。

[提供媒体の内容]

項番	ファイル名	内 容		インストール場所
1	IBMUNI.tbl	IBM→UNICODE 変換テーブル	mto 方向標準変換 テーブル	FAL/FCU 起動フォルダ (C:\FCU)
2	KS78UNI.tbl	KEIS78→UNICODE 変換テーブル		
3	KS90UNI.tbl	KEIS83/90→UNICODE 変換テーブル		
4	JEFUNI.tbl	JEF→UNICODE 変換テーブル	otm 方向標準変換 テーブル	
5	UNIIBM.tbl	UNICODE→IBM 変換テーブル		
6	UNIKS78.tbl	UNICODE→KEIS78 変換テーブル		
7	UNIKS90.tbl	UNICODE→KEIS83/90 変換テーブル		
8	UNIJEF.tbl	UNICODE→JEF 変換テーブル		
9	EditOpUNI	UNICODE 用編集オプションファイル		
10	libuocuni.ver	バージョン情報ファイル		

## 4. Code Converter の使い方

### 4.2. ライセンスキーコードの入力

Code Converter、Code Converter-EUC、Code Converter-CODECHK、Code Converter-UNICODE を動作させるにはライセンスキーコードを入力する必要があります。ppkeyset コマンドを用いて、ライセンスキーコードを入力してください。

ライセンスキーコードの入力は、新規インストール及びバージョンアップ/ダウン時に必要となります。

- (注) 1. AIX の時、プログラム動作後、ppkeyset を行うとエラーとなることがあります。slibclean コマンド発行後再度 ppkeyset の操作を行ってください。
2. Code Converter のライセンスキーコードは、File Exchange (FAL/FCU) と同じ以下のライセンスキーコードを入力してください。
- Hitachi Cross-OS File Exchange software (P-012R-JDS1K)
3. Code Converter オプション (Code Converter-EUC、CODECHK、UNICODE) のライセンスキーコードは、以下のライセンスキーコードを入力してください。
- Hitachi Cross-OS File Exchange software Code Converter option set (P-F012R-JE12SK)
- このライセンスキーコードを入力すると、上記の3つのオプション製品が動作できます。

#### (1) ライセンスキーコード設定方法

コマンドラインより、下記に示す通りに ppkeyset (32 ビット用) コマンドまたは ppkeyset64 (64 ビット用) コマンドを起動してください。

コマンドラインより、下記に示す通りに ppkeyset コマンドを起動してください。

[UNIX 系の場合]

##### ① HP-UX の場合

```
ppkeyset64 <ライセンスキーコード> /usr/lib/hpux64/libuoc64.so (IA-64 の場合)
```

##### ② Solaris の場合

```
ppkeyset <ライセンスキーコード> /usr/lib/libuoc.so.1
```

```
ppkeyset64 <ライセンスキーコード> /usr/lib/sparcv9/libuoc64.so.1
```

##### ③ AIX の場合

```
ppkeyset <ライセンスキーコード> /usr/lib/libuoc.a
```

```
ppkeyset64 <ライセンスキーコード> /usr/lib/libuoc64.a
```

##### ④ Linux の場合

```
ppkeyset <ライセンスキーコード> /usr/lib/libuoc.so.1
```

```
ppkeyset64 <ライセンスキーコード> /usr/lib/libuoc64.so.1
```

##### ⑤ FAL/FCU が提供している autopkeyset コマンドを使用する場合

```
autopkeyset <ライセンスキーファイル>
```

```
autopkeyset64 <ライセンスキーファイル>
```

[Windows 系の場合]

##### ① FAL/FCU インストール先ディレクトリ内の、ppkeyset.exe をダブルクリックします。

##### ② MS-DOS プロンプト画面が表示され、以下のメッセージが表示されます。

```
Please enter <License key> [Installed Directory or File]... or <quit> for exit.
```

##### ③ <ライセンス> <Code Converter インストール先ディレクトリ> と入力して Enter キーを押下してください。但しディレクトリ名にスペースを含む場合、直接ファイル名を指定してください。 (<ライセンス> uoc.dll)

##### ④ 再度 Please enter <License key> [Installed Directory or File]... or <quit> for exit. が表示されますので、"quit"を入力して Enter キーを押下してください。

ライセンスキーファイルを使用する場合は、コマンドプロンプト (DOS 窓) を開き、下記の示す通りに FAL/FCU が提供している autopkeyset.exe コマンドを起動してください。

```
autopkeyset <ライセンスキーファイル>
```

## 4. Code Converter の使い方

### (2) ライセンスコード入力例

Code Converter 及び Code Converter オプションのライセンスコード入力の例を示します。また、Code Converter の PP コードを下表に示します。

[Code Converter の PP コード]

項番	製品	PP コード
1	Code Converter	0 8 0 1
2	Code Converter オプション	0 8 0 4

[UNIX の場合]

HP-UX を例に ppkeyset コマンドによるライセンスコード入力例を示します。

#### ①Code Converter のライセンスコード入力例

##### ■ライセンスコード入力が正常終了した場合

```
# ppkeyset <ライセンスコード> /usr/lib/libuoc.sl
[/usr/lib/libuoc.sl] skipped for inequality to PP_CODE[0804]. ... Code Converter オプションスキップ
[/usr/lib/libuoc.sl] set in available for all days. ... ライセンスコード入力完了
#
```

##### ■ライセンスコード入力後に再度ライセンスコードを入力した場合

```
# ppkeyset <ライセンスコード> /usr/lib/libuoc.sl
[/usr/lib/libuoc.sl] skipped for inequality to PP_CODE[0804]. ... Code Converter オプションスキップ
[/usr/lib/libuoc.sl][0801] available for all days on vhdlda01 on Ser# 12345. ... 2重入力
#
```

#### ②Code Converter オプションのライセンスコード入力例

##### ■ライセンスコード入力が正常終了した場合

```
# ppkeyset <ライセンスコード> /usr/lib/libuoc.sl
[/usr/lib/libuoc.sl] set in available for all days. ... ライセンスコード入力完了
[/usr/lib/libuoc.sl] skipped for inequality to PP_CODE[0801]. ... Code Converter スキップ
#
```

##### ■ライセンスコード入力後に再度ライセンスコードを入力した場合

```
# ppkeyset <ライセンスコード> /usr/lib/libuoc.sl
[/usr/lib/libuoc.sl][0804] available for all days on vhdlda01 on Ser# 12345. ... 2重入力
[/usr/lib/libuoc.sl] skipped for inequality to PP_CODE[0801]. ... Code Converter スキップ
#
```

#### ③Code Converter 以外のライセンスコードを入力した場合

```
# ppkeyset <ライセンスコード> /usr/lib/libuoc.sl
[/usr/lib/libuoc.sl] skipped for inequality to PP_CODE[0804]. ... Code Converter オプションスキップ
[/usr/lib/libuoc.sl] skipped for inequality to PP_CODE[0801]. ... Code Converter スキップ
#
```

## 4. Code Converter の使い方

[Windows 系場合]

コマンドプロンプト (DOS 窓) で ppkeyset.exe コマンドによるライセンスコード入力例を示します。

### ① Code Converter のライセンスコード入力例

#### ■ライセンスコード入力が正常終了した場合

```
C:\%FCU> ppkeyset.exe <ライセンスコード> uoc.dll
[uoc.dll] set in available for all days.      ... ライセンスコード入力完了
[uoc.dll] skipped for inequality to PP_CODE[0804]. ... Code Converter オプションスキップ
C:\%FCU>
```

#### ■ライセンスコード入力後に再度ライセンスコードを入力した場合

```
C:\%FCU> ppkeyset.exe <ライセンスコード> uoc.dll
[uoc.dll][0801] available for all days on vhdlda01 on Ser# 12345. ... 2重入力
[uoc.dll] skipped for inequality to PP_CODE[0804]. ... Code Converter オプションスキップ
C:\%FCU>
```

### ② Code Converter オプションのライセンスコード入力例

#### ■ライセンスコード入力が正常終了した場合

```
C:\%FCU> ppkeyset.exe <ライセンスコード> uoc.dll
[uoc.dll] skipped for inequality to PP_CODE[0801]. ... Code Converter スキップ
[uoc.dll] set in available for all days.      ... ライセンスコード入力完了
C:\%FCU>
```

#### ■ライセンスコード入力後に再度ライセンスコードを入力した場合

```
C:\%FCU> ppkeyset.exe <ライセンスコード> uoc.dll
[uoc.dll] skipped for inequality to PP_CODE[0801]. ... Code Converter スキップ
[uoc.dll][0804] available for all days on vhdlda01 on Ser# 12345. ... 2重入力
C:\%FCU>
```

### ③ Code Converter 以外のライセンスコードを入力した場合

```
C:\%FCU> ppkeyset.exe <ライセンスコード> uoc.dll
[uoc.dll] skipped for inequality to PP_CODE[0801]. ... Code Converter スキップ
[uoc.dll] skipped for inequality to PP_CODE[0804]. ... Code Converter オプションスキップ
C:\%FCU>
```

## 4. Code Converter の使い方

### 4.3. 使い方概説

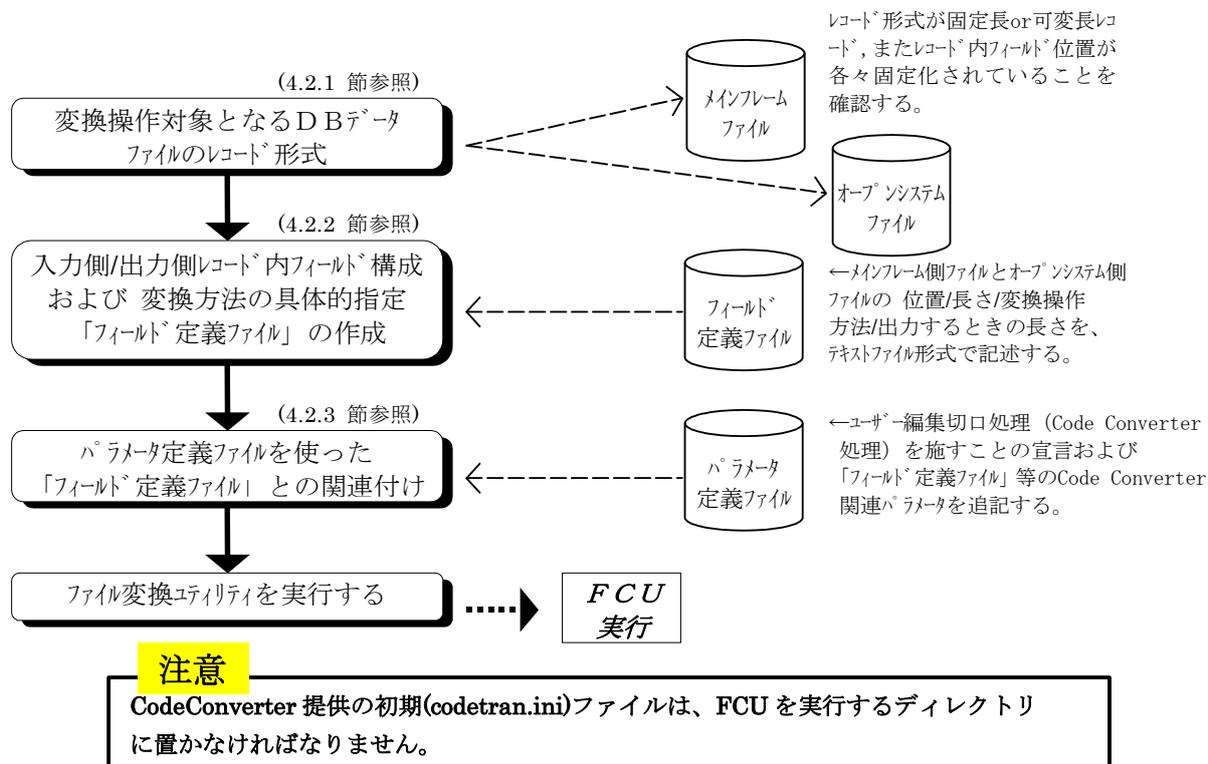


図 4.3.1 使い方概説

#### (1) 変換操作対象となるDBデータファイルのレコード形式

Code Converter を使って漢字変換などの変換操作を施す場合、DBデータファイルは、固定長レコード及び可変長レコードの順編成ファイル(SAM)を対象にしています。

また、レコード内フィールド位置が各々固定化されている必要があります。

まず、メインフレーム側ファイルとオープンシステム側ファイルの各々のレコード形式を確認してください。

#### (2) 入力側/出力側レコード内フィールド構成および変換方法の具体的指定 (「フィールド定義ファイル」の作成)

次に、入力側レコードのどの位置にどんな変換操作を必要とするフィールドがあるのか、また、その変換結果を出力側レコードにどんな長さのフィールドとして編集出力すれば良いのかを具体的に指定します。

本指定のために作成するのが「フィールド定義ファイル」です。

フィールド定義ファイルは、変換操作対象となるDBデータファイル毎に異なるレコード形式(フィールド構成)に対応させて作成するもので、入力側レコードに含まれる全フィールドについて、その(位置/長さ/変換操作方法/出力するときの長さ)を、一定の書式に従って、ユーザーがテキストファイル形式で記述するものです。

#### (3) パラメータ定義ファイルを使った「フィールド定義ファイル」との関連付け

変換操作対象となるDBデータファイルと「フィールド定義ファイル」を関連付けるため、ファイル変換ユーティリティが mto 変換または otm 変換するDBデータファイル(メインフレーム側 SAM ファイル と オープンシステム側 UNIX/PC ファイル の1セット)を指定するのに用いている「パラメータ定義ファイル」を使います。

「パラメータ定義ファイル」に対しては、ユーザー編集切口処理 (Code Converter 処理) を施すことの宣言、および、「フィールド定義ファイル」等のCode Converter 関連パラメータを追記します。

(「パラメータ定義ファイル」も、一定の書式に従って、ユーザーがテキストファイル形式で記述するものです。)

#### (4) 以上で準備完了です。あとは、ファイル変換ユーティリティを実行してください。

## 4. Code Converter の使い方

### 4.3.1. 変換操作対象となるDBデータファイルのレコード形式

Code Converter を使って漢字変換などの変換操作を施す場合、DBデータファイルは、固定長レコード及び可変長レコードの順編成ファイル(SAM)を対象にしています。

また、レコード内フィールド位置は各々固定化されている必要があります。

まず、メインフレーム側ファイルとオープンシステム側ファイルの各々のレコード形式を確認してください。

Code Converter が対象とする SAM レコード形式は、各レコードの固定位置から入力データを切り出してくるため、図 4.3.2及び図 4.3.3に示すように、各フィールドの始まるレコード内相対バイト位置とフィールドバイト長が、全レコード統一されていることが必要です。(例：DB2 の REORG エイリティが出力した SAM ファイルなど)

		レコード長					
位置 0		6	10	18	58	78	
		制御情報(6B)	ハズリ(4B)	リソースデータ(8B)	英数文字(40B)	漢字(20B)	パック10進(6B)
レコード 1		X'00006B000301'	X'80000001'	00000001	ABCDEFGF..	日立..	X'F00000000001'
レコード 2		X'00006B000302'	X'80000002'	00000002	BCDEFGH..	亜阿..	X'F00000000002'
レコード 3		X'00006B000303'	X'80000003'	00000003	CDEFGHI..	伊医..	X'F00000000003'
レコード 4		X'00006B000304'	X'80000004'	00000004	DEFGHIJ..	A B C..	X'F00000000004'

図 4.3.2 入力対象SAMレコード形式 (固定長レコードの場合)

		レコード長					
位置 0		6	10	18	58	78	
		制御情報(6B)	ハズリ(4B)	リソースデータ(8B)	英数文字(40B)	漢字(20B)	パック10進(6B)
レコード 1		X'00006B000301'	X'80000001'	00000001	ABCDEFGF..	日立..	
レコード 2		X'00006B000302'	X'80000002'				
レコード 3		X'00006B000303'	X'80000003'	00000003	CDEFGHI..	伊医..	X'F00000000003'
レコード 4		X'00006B000304'	X'80000004'	00000004			

図 4.3.3 入力対象SAMレコード形式 (可変長レコードの場合)

Code Converter で提供しているレコード形式とコード変換形態を表 4.3.1に示します。

表 4.3.1 レコード形式とコード変換形態

変換	メインフレームレコード			UNIX/PCレコード			参照
mto	固定長レコード	→	(変換)	→	固定長レコード	(1)	
	可変長レコード	→	(変換)	→	可変長レコード	(2) (a)	
otm	固定長レコード	←	(変換)	→	固定長レコード	(1)	
	可変長レコード	←	(変換)	→	可変長レコード	(2) (b)	
	固定長レコード	←	(変換)	→	可変長レコード	(2) (c)	

#### 4. Code Converter の使い方

##### (1) 固定長レコード形式のコード変換

メインフレームレコード及びUNIX/PCレコードが共に固定長レコードである場合、mto方向及びotm方向のコード変換を図4.3.4に示します。

mto方向でのコード変換時、デリミタ付加指定があれば、UNIX/PC側各レコードの最終フィールドの後ろにデリミタを付加して出力します。コード変換により、UNIX/PCレコード長が、メインフレームレコード長と異なる場合、必ずデリミタを付加してください。デリミタを付加しないとotm変換ができなくなります。

また、各フィールドの入力フィールド長の合計が、メインフレームレコード長と一致するように、フィールド定義をしてください。

otm方向でのコード変換時、デリミタ付加指定があれば、デリミタを検索してUNIX/PCレコードを入力し、各フィールド単位にコード変換後メインフレームレコードへ出力します。なお、デリミタはメインフレームレコードに出力しません。

デリミタ付加指定がなければ、メインフレームレコード長と同じ長さでコード変換します。

また、各フィールドの出力フィールド長の合計が、メインフレームレコード長と一致するように、フィールド定義をしてください。

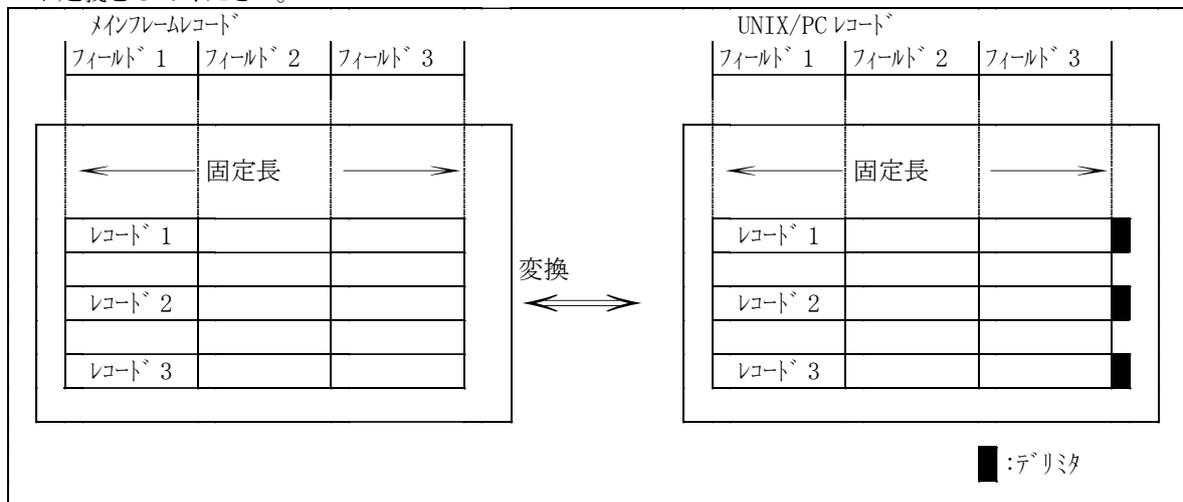


図 4.3.4 固定長レコードのコード変換概要

## 4. Code Converter の使い方

### (2) 可変長レコード形式のコード変換

可変長レコードのコード変換時は、下記の注意が必要です。

- 必ずデリミタ指定をしてください。
- フィールド定義文で指定する各フィールド長の合計は、メインフレームの最大レコード長-4と一致するように指定してください。
- フィールド定義文で指定する各フィールドは、メインフレームの各レコード内で完結するように指定してください。
- 最終フィールドが可変長の場合（入力フィールド長として end を指定した場合）当該フィールドに数値変換指定はできません。数値コード（パック形式/ゾーン形式データ）は固定フィールドとして扱っています。

#### (a) メインフレームの可変長レコードをUNIX/PCの可変長レコードへコード変換 (m t o)

メインフレームの可変長レコードをフィールド毎にコード変換し、UNIX/PCの可変長レコードとして出力します。この場合、フィールド定義の end 文にて、必ずデリミタを指定してください。

##### ①フィールド長が固定の場合

フィールド定義文で指定する各フィールドが固定の長さである場合(入力フィールド長として end を指定しない場合) mto 方向のコード変換を図 4.3.5に示します。

メインフレームの可変長は、最大レコード長-4以内の長さです。

[説明]

- レコード 1：フィールド 1のみコード変換します。フィールド 2~3は処理しません。
- レコード 2：フィールド 1~3に対してコード変換します。
- レコード 3：フィールド 1~2に対してコード変換し、フィールド 3は処理しません。

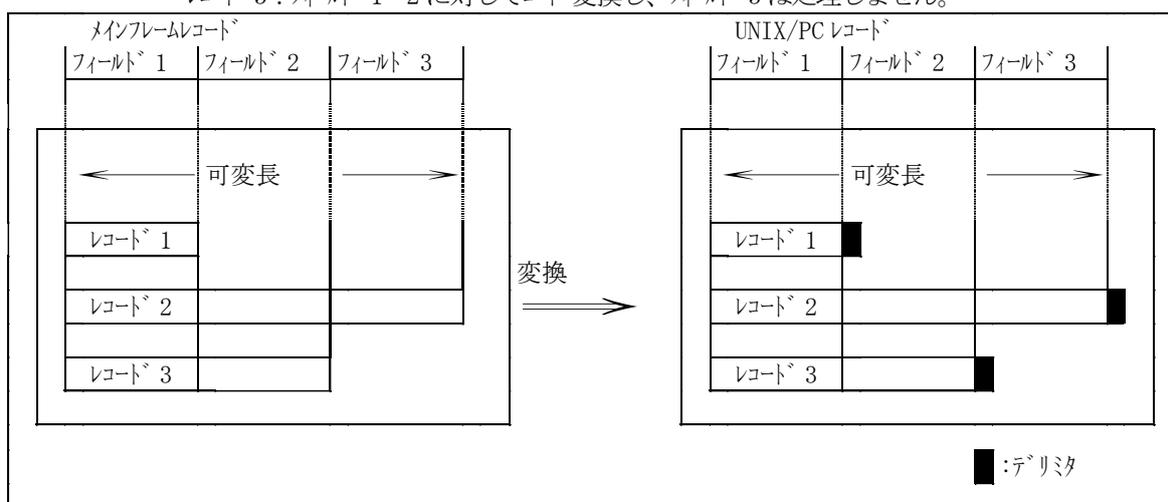


図 4.3.5 可変長レコード (end 指定なし) のコード変換概要

##### ②最終フィールド長が可変の場合

フィールド定義文で指定する最終フィールドが可変の長さである場合（入力フィールド長として end を指定した場合） mto 方向のコード変換を図 4.3.6に示します。

メインフレームの可変長は、最大レコード長-4以内の長さです。

[説明]

レコード中の最終フィールドに対してのみ、フィールド定義文にて end 指定ができます。この場合、最終フィールドはフィールドの開始位置からレコードの終端までがコード変換対象データとなります。■で示すフィールド2が end 指定です。

#### 4. Code Converter の使い方

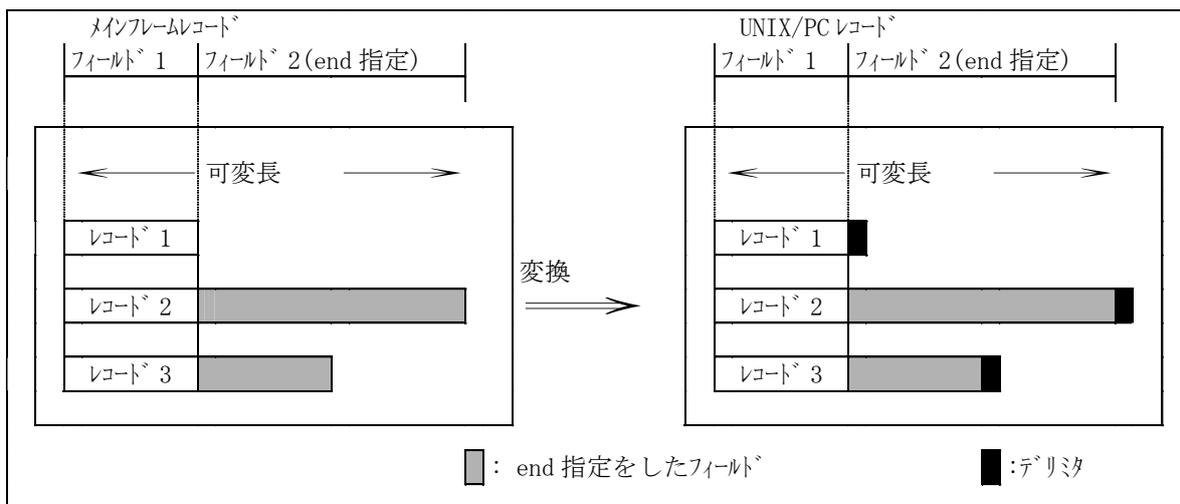


図 4.3.6 可変長レコード (end 指定あり) のコード変換概要

(b) UNIX/PC の可変長レコードをメインフレームの可変長レコードへコード変換 (o t m)

UNIX/PC の可変長レコードのデリミタを削除しフィールド毎にコード変換後、可変長レコードとしてメインフレームに出力します。この場合、フィールド定義の end 文にて必ずデリミタを指定してください。また、コード変換後のレコード長はメインフレームで定義した最大レコード長 - 4 以内の長さにしてください。

①フィールド長が固定の場合

フィールド定義文で指定する各フィールドが固定の長さである場合(入力フィールド長として end を指定しない場合) otm 方向のコード変換を図 4.3.7に示します。デリミタは、メインフレームレコードへは出力しません。メインフレームの可変長は、最大レコード長 - 4 以内の長さです。

[説明]

- レコード 1 : フィールド 1 のみコード変換します。フィールド 2~3 は処理しません。
- レコード 2 : フィールド 1~3 に対してコード変換します。
- レコード 3 : フィールド 1~2 に対してコード変換し、フィールド 3 は処理しません。

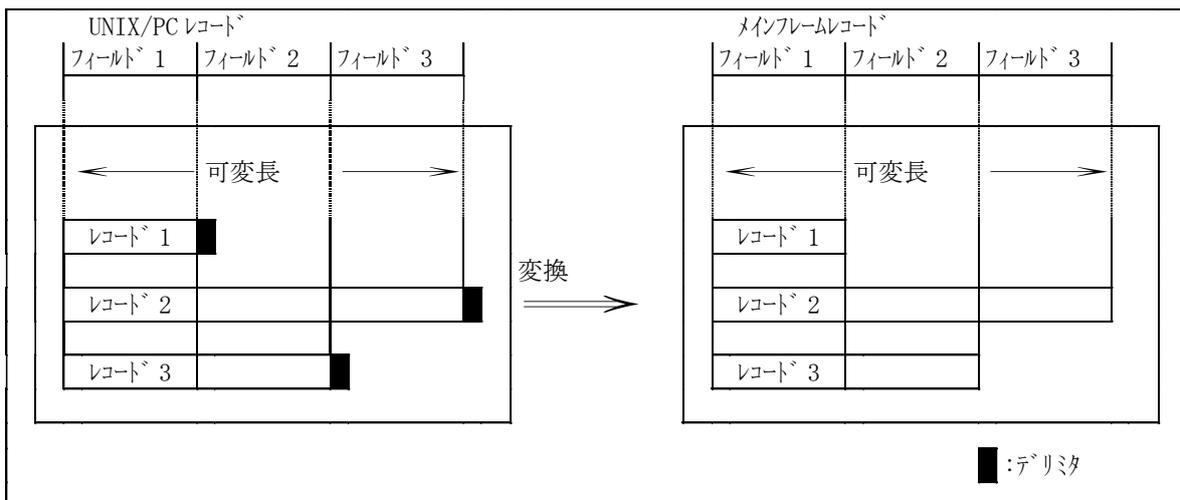


図 4.3.7 可変長レコード (end 指定なし) のコード変換概要

## 4. Code Converter の使い方

### ②最終フィールド長が可変の場合

フィールド定義文で指定する最終フィールドが可変の長さである場合（入力フィールド長として end を指定した場合）otm 方向のコード変換を図 4.3.8 に示します。

#### [説明]

レコード中の最終フィールドに対してのみ、フィールド定義文にて end 指定ができます。この場合、最終フィールドはフィールドの開始位置からレコードの終端までがコード変換対象データとなります。■ で示すフィールド 2 が end 指定です。

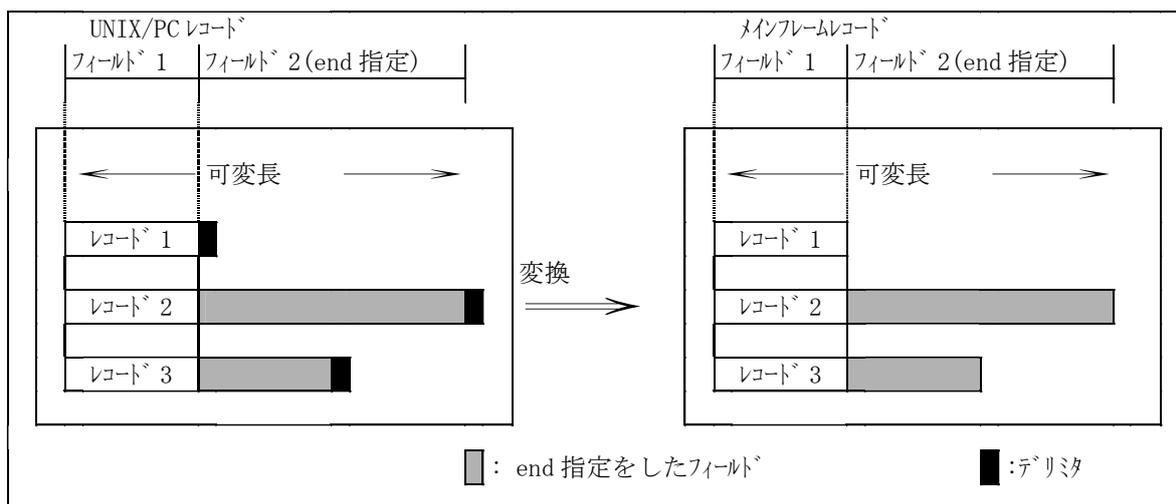


図 4.3.8 可変長レコード (end 指定あり) のコード変換

### (c) UNIX/PC の可変長レコードをメインフレームの固定長レコードへコード変換 (otm)

UNIX/PC の可変長レコードのデリミタを削除し、フィールド毎にコード変換後、固定長レコードとしてメインフレームに出力します。この場合、フィールド定義の end 文にて、必ずパディングとデリミタを指定してください。コード変換後、メインフレームレコードで余ったエリアには、パディング指定コードを設定します。なお、フィールド単位で余ったエリアには、編集オプションファイルのフィールド充填コードを充填します。

また、各フィールドの出力フィールド長の合計が、メインフレームの固定長以内になるようにフィールド定義をしてください。

(注) パディングされたメインフレームレコードを、mto 変換にて、UNIX/PC レコードに戻すことはできません。

#### ①フィールド長が固定の場合

フィールド定義文で指定する各フィールドが固定の長さである場合(入力フィールド長として end を指定しない場合) otm 方向のコード変換を図 4.3.9 に示します。

デリミタは、メインフレームレコードへは出力されません。

#### [説明]

- レコード 1 : フィールド 1 のみコード変換し、メインフレームレコードの終端まで、パディングコードを充填します。フィールド 2~3 は処理しません。
- レコード 2 : フィールド 1~3 に対してコード変換します。
- レコード 3 : フィールド 1~2 に対してコード変換し、メインフレームレコードの終端まで、パディングコードを充填します。フィールド 3 は処理しません。

#### 4. Code Converter の使い方

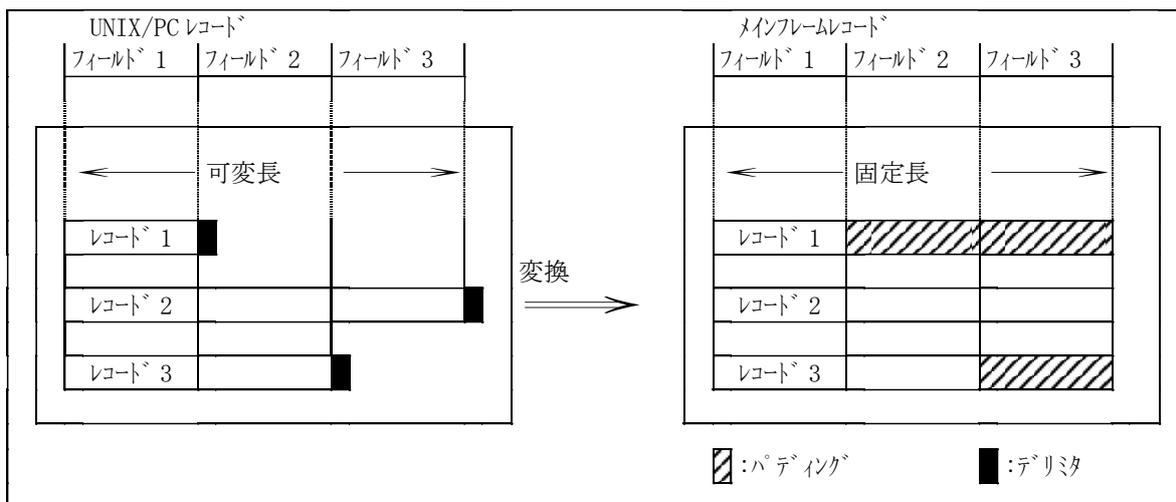


図 4.3.9 可変長レコード (end 指定なし) のコード変換

#### ②最終フィールド長が可変の場合

フィールド定義文で指定する最終フィールドが可変の長さである場合 (入力フィールド長として end を指定した場合) otm 方向のコード変換を図 4.3.10 に示します。

【説明】

レコード中の最終フィールドに対してのみ、フィールド定義文にて end 指定ができます。この場合、最終フィールドはフィールドの開始位置からレコードの終端までがコード変換対象データとなります。■で示すフィールド2が end 指定です。

- レコード 1 : フィールド 1 のみコード変換し、メインフレームレコードの終端まで、パディングコードを充填します。フィールド 2 は処理しません。
- レコード 2 : フィールド 1~2 に対してコード変換します。
- レコード 3 : フィールド 1~2 に対してコード変換し、メインフレームレコードの終端まで、パディングコードを充填します。

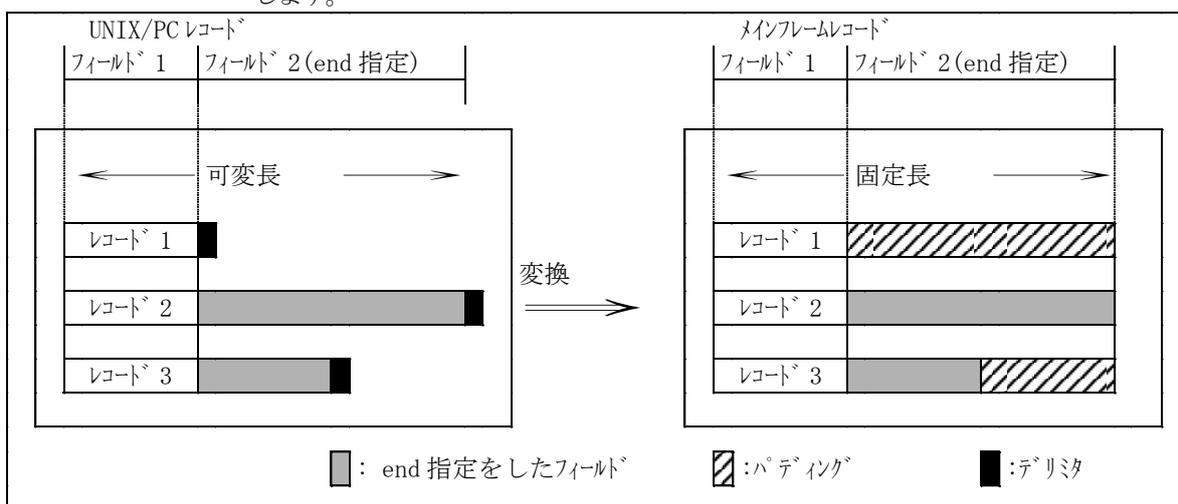


図 4.3.10 可変長レコード (end 指定あり) のコード変換

## 4. Code Converter の使い方

### 4.3.2. フィールド定義ファイルの作成

入力側レコードのどの位置にどんな変換操作を必要とするフィールドがあるのか、また、その変換結果を出力側レコードにどんな長さのフィールドとして編集出力すれば良いのかを、具体的に指定します。

本指定のために作成するのが「フィールド定義ファイル」です。

フィールド定義ファイルは、変換操作対象となるDBデータファイル毎に異なるレコード形式（フィールド構成）に対応させて作成するもので、入力側レコードに含まれる全フィールドについて、その（位置／長さ／変換操作方法／出力するときの長さ）を、以下に示す書式に従って記述します。

フィールド定義ファイルは、vi などのエディタを使いテキスト形式のファイルとして作成し、コメントも含め最大 1000 行まで記述できるようになっています。作成したファイルは、オープン環境 (UNIX/PC サーバ) のカレントディレクトリに格納して使用します。

#### [書式]

フィールド定義ファイルは、フィールド定義文、end 文、および コメント の 3 つの要素から構成されます。

1 行 1 文で 127 カラム、1000 行まで (コメント含む) 記述できます。

#### (1) フィールド定義文

フィールド定義文は、入力レコード中の全フィールドについて、先頭フィールドから順に 1 行 / 1 フィールドずつ記述します。

<code>fld</code> △ レコード内相対バイト位置 △ 入力フィールドバイト長 △ フィールド属性 △ [出力フィールドバイト長]
---

△ : 半角 1 バイト以上の空白を示す。 [] で囲ったパラメータは省略可能であることを示す。

- ① `fld` … この行がフィールド定義文であることを示すキーワード (予約語) です。
- ② レコード内相対バイト位置 … 当該フィールドのレコード内相対開始バイト位置を 0 スタートの十進数で記述します。
- ③ 入力フィールドバイト長 … 当該フィールドの入力側 (変換元) バイト長を十進数で記述します。
- ④ フィールド属性 … 当該フィールドに対する変換操作方法を、次のように指定します。
  - ・出力抑止フィールドの場合 → SKIP
  - ・無変換フィールドの場合 → NOP
  - ・英数文字変換フィールドの場合 → CHAR : OPnn  
(OPnn は、表 4.3.3～表 4.3.6、表 4.3.9～表 4.3.15 を参照願います)
  - ・漢字変換フィールドの場合 → KANJI : OPnn  
(OPnn は、表 4.3.3～表 4.3.6、表 4.3.9～表 4.3.15 を参照願います)
  - ・数値変換フィールドの場合 → NUM : OPnn  
(OPnn は、表 4.3.8、表 4.3.15-1 を参照願います)

**数値変換フィールドは、必ず出力フィールドバイト長を記述してください。**
- ⑤ 出力フィールドバイト長 … 当該フィールドの出力先 (変換先) エリアのバイト長を十進数で記述します。  
本指定は、入力フィールドバイト長と同じ長さにする場合、省略できます。  
(注意) 無変換フィールド (NOP) と出力抑止フィールド (SKIP) では、出力フィールドバイト長の指定を必ず省略してください。

なお、当該フィールドがレコード内最終かつ、その位置より後ろ全てを入力フィールドとして良い場合に限り、次のように入力フィールドバイト長を "end" と省略して書くことができます。

(具体的な値を書かずに済むので、最終フィールドだけ長さが異なる複数のデータセット・レコードに対して、共通に使用可能なフィールド定義文 (フィールド定義ファイル) を作成することができます。)

<code>fld</code> △ レコード内相対バイト位置 △ <code>end</code> △ フィールド属性 △ [出力フィールドバイト長]
--

△ : 半角 1 バイト以上の空白を示す。 [] で囲ったパラメータは省略可能であることを示す。

## 4. Code Converter の使い方

- ① fld, ② レコード内相対バイト位置, ④ フィールド属性, ⑤ 出力フィールドバイト長 … 非省略形の場合と同じです。  
 ③ end … 当該フィールドより後ろ全てを入力フィールドとして定義します。  
 また、レコード形式により下記のように相違します。
- ・固定長レコードの場合：(入力フィールドバイト長) = (入力レコード長) - (レコード内相対バイト位置)
  - ・可変長レコードの場合：(入力フィールドバイト長) = (コード変換後の長さ)
- (注意) 可変長レコードの場合、end 指定をしたフィールドに対して、数値変換指定はできません。  
 数値変換は固定長レコードとして扱っています。

各フィールド属性に対する変換モード指示パラメータを表 4.3.2 に示します。

表 4.3.2 フィールド種別・フィールド属性対応一覧

項番	フィールド種別	フィールド属性	OPnn 参照表		
			シフト JIS/JIS8	EUC コード	Unicode
1	英数文字変換フィールド	CHAR : OPnn	表 4.3.3~	表 4.3.9~	表 4.3.12~
2	漢字変換フィールド	KANJI : OPnn	表 4.3.6	表 4.3.11	表 4.3.15
3	数値変換フィールド	NUM : OPnn	表 4.3.8		表 4.3.15-1
4	無変換フィールド	NOP	—		
5	出力抑止フィールド	SKIP	—		

### (2) end文

end 文は、入力レコード中の全フィールドについてフィールド定義文を記述し終わった後、最後に1行記述します。

```
end Δ [Delimiter=デリミタ指定コード][, Padding=パディング指定コード]
```

Δ：半角1バイト以上の空白を示す。□ で囲ったパラメータは省略可能であることを示す。

- ① end … 全てのフィールド定義文記述が終了したことを示すキーワード(予約語)です。  
 ② Delimiter=デリミタ指定コード … オープンシステム側レコードの末尾にデリミタを付加する場合、デリミタ指定コードを次のように指定します。可変長データセットの変換の場合は、必ずデリミタ指定をつけてください。
- ・CR=(OD) を付加する場合 → Delimiter=CR
  - ・LF=(OA) を付加する場合 → Delimiter=LF
  - ・CRLF=(ODOA) を付加する場合 → Delimiter=CRLF (\*1)
  - ・UNICODE の CR=(OD00) を付加する場合 → Delimiter=UNICR
  - ・UNICODE の LF=(OA00) を付加する場合 → Delimiter=UNILF
  - ・UNICODE の CRLF=(OD00OA00) を付加する場合 → Delimiter=UNICRLF (\*1)

(\*1) Windows の場合のみです。

- ③ Padding=パディング指定コード … オープンシステム側の可変長レコードを、メインフレーム側の固定長レコードへコード変換する場合、変換後の最終フィールドの終端から、メインフレーム側レコードの終端までの領域が余った時の充填コードを、2桁の16進文字で指定します。固定長データセットのOTM変換の場合は、必ずパディング指定コードを付けてください。

(注意) デリミタを付加した場合、出力レコード長が増減しますので、出力先容量確保の際、十分ご注意ください。  
 また、編集レコードに半角/全角文字以外のバイトデータを含む場合は、そのバイト部分にデリミタと一致するデータを含む危険性があるため、特にご注意ください。デリミタによるレコード切り出しができない場合があります。

## 4. Code Converter の使い方

### (3) コメント

コメントは、フィールド定義文と同一行内（フィールド定義文より後方）または、独立したコメント行として適宜付加できるようにになっています。

- フィールド定義文と同一行内では、フィールド定義文必要項目数(5パラメータ)より後方がコメント記述可能エリアです。
- 出力フィールドバ付長を省略した場合は、#を記述して以後をコメント記述エリアとして使用できます。
- 行の先頭に#を記述すると、行全体がコメントになります。
- また、空白行もコメントとして扱います。

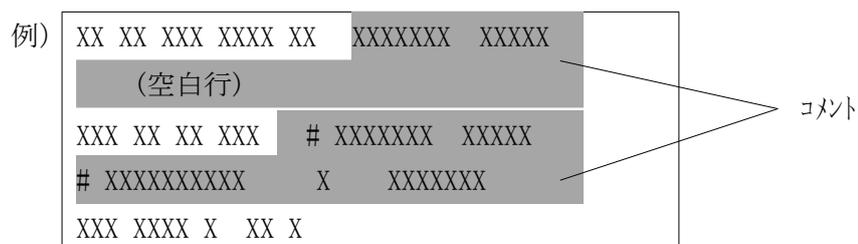


図 4.3.11 フィールド定義ファイル内のコメント記述

## 4. Code Converter の使い方

[記述例]

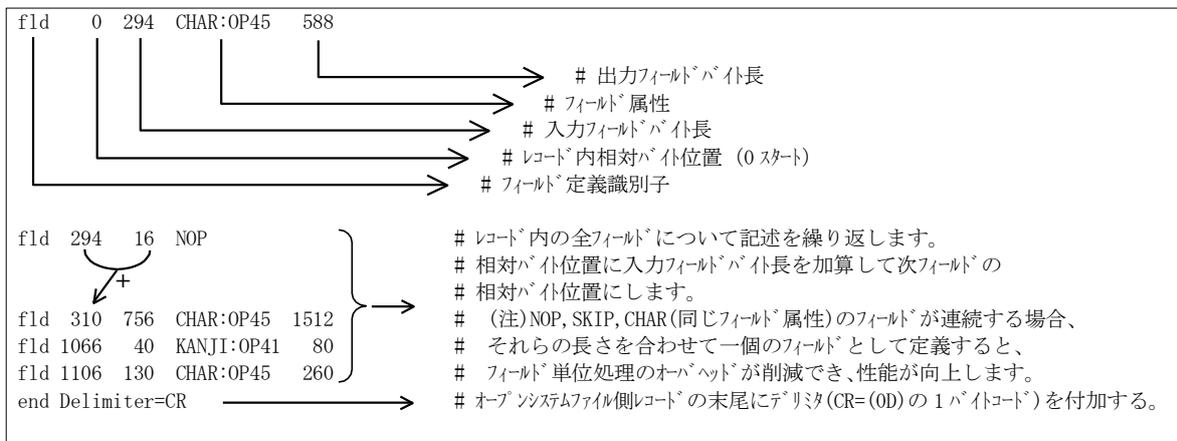


図 4.3.12 フィールド定義ファイル記述例

<説明>

この例では、入力レコード長は、 $294+16+756+40+130 = 1236$  バイト

出力レコード長は、 $588+16+1512+80+260 = 2456$  に、デリミタ1バイトを加えて2457 バイトです。

- ・ 第一フィールドは、先頭から294バイトのデータであり、半角の英数文字変換をして出力します。変換は、編集オプションファイルの [CHAR:OP45] で指示されたモード指示情報に従います。出力フィールドバイト長は、588バイトです。
- ・ 第二フィールドは、後続16バイトのデータであり、そのまま無変換で出力します。出力フィールドバイト長は、16バイトです。
- ・ 第三フィールドは、後続756バイトのデータであり、半角の英数文字変換をして出力します。変換は、編集オプションファイルの [CHAR:OP45] で指示されたモード指示情報に従います。出力フィールドバイト長は、1512バイトです。
- ・ 第四フィールドは、後続40バイトのデータであり、全角のみの漢字変換をして出力します。変換は、編集オプションファイルの [KANJI:OP41] で指示されたモード指示情報に従います。出力フィールドバイト長は、80バイトです。
- ・ 第五フィールドは、後続130バイトのデータであり、半角の英数文字変換をして出力します。変換は、編集オプションファイルの [CHAR:OP45] で指示されたモード指示情報に従います。出力フィールドバイト長は、260バイトです。

1行1文で127カラム、1000行まで(コメント含む)記述できます。

## 4. Code Converter の使い方

[フィールド属性一覧]

(1) オープンシステムが「シフトJIS / JIS8」の場合の変換モード指示パラメータ

表 4.3.3 IBM 漢字コード体系を基調とするフィールド属性一覧

オープンシステム側 (全角コード: シフトJIS) (半角コード: JIS8)			変換モード指示パラメータ	
メインフレーム側 (全角コード: IBM漢字) (半角コード: EBCDIC/EBCDIK)			mto変換用	otm変換用
漢字変換	全角のみ		OP11	OP16
	全/半角混在	半角に英小文字含む	OP12	OP17
		「」 含まない	OP13	OP18
英数文字変換	半角のみ	半角に英小文字含む	OP14	OP19
		「」 含まない	OP15	OP1A

表 4.3.4 KEIS78 コード体系を基調とするフィールド属性一覧

オープンシステム側 (全角コード: シフトJIS) (半角コード: JIS8)			変換モード指示パラメータ	
メインフレーム側 (全角コード: KEIS78) (半角コード: EBCDIC/EBCDIK)			mto変換用	otm変換用
漢字変換	全角のみ		OP21	OP26
	全/半角混在	半角にカナ文字含まない	OP22	OP27
		「」 含む	OP23	OP28
英数文字変換	半角のみ	半角にカナ文字含まない	OP24	OP29
		「」 含む	OP25	OP2A

表 4.3.5 KEIS83/90 コード体系を基調とするフィールド属性一覧

オープンシステム側 (全角コード: シフトJIS) (半角コード: JIS8)			変換モード指示パラメータ	
メインフレーム側 (全角コード: KEIS83/90) (半角コード: EBCDIC/EBCDIK)			mto変換用	otm変換用
漢字変換	全角のみ		OP31	OP36
	全/半角混在	半角にカナ文字含まない	OP32	OP37
		「」 含む	OP33	OP38
英数文字変換	半角のみ	半角にカナ文字含まない	OP34	OP39
		「」 含む	OP35	OP3A

表 4.3.6 JEF 漢字コード体系を基調とするフィールド属性一覧

オープンシステム側 (全角コード: シフトJIS) (半角コード: JIS8)			変換モード指示パラメータ	
メインフレーム側 (全角コード: JEF漢字) (半角コード: EBCDIC/EBCDIK)			mto変換用	otm変換用
漢字変換	全角のみ		OP41	OP46
	全/半角混在	半角にカナ文字含まない	OP42	OP47
		「」 含む	OP43	OP48
英数文字変換	半角のみ	半角にカナ文字含まない	OP44	OP49
		「」 含む	OP45	OP4A

#### 4. Code Converter の使い方

さらに Code Converter では、ファイル変換ユーティリティ (FCU) が EBCDIC (カナ文字含まない) と ASCII コードとの間で行なっているのと同じの ASCII コード変換機能を用意しています。使用する場合は、次のフィールド属性を指定してください。

- mto 変換用 → CHAR:OP1
- otm 変換用 → CHAR:OP6

このとき適用される EBCDIC⇔ASCII コード変換表を次に示します。この変換テーブルをさらにカスタマイズして使用する場合には、ccnv コマンドをお使いください。(ccnv コマンドの使い方は、10 章を参照ください。)

表 4.3.7 EBCDIC ⇔ ASCII コード変換表

EBCDICコード		+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
00	ASCII	00	01	02	03	9C	09	86	7F	97	8D	8E	0B	0C	0D	0E	0F
	文字	NUL	SOH	STX	ETX	PF	HT	LC	DEL	GE	RLF	SMM	VT	FF	CR	SO	SI
10	ASCII	10	11	12	13	9D	85	08	87	18	19	92	8F	1C	1D	1E	1F
	文字	DLE	DC1	DC2	DC3	TM		BS	IL	CAN	EM	CC	CUI	IFS	IGS	IRS	IUS
20	ASCII	80	81	82	83	84	0A	17	1B	88	89	8A	8B	8C	05	06	07
	文字	DS	SOS	FS		BYP	NL	ETB	ESC			SW	CUI		ENQ	ACK	BEL
30	ASCII	90	91	16	93	94	95	96	04	98	99	9A	9B	14	15	9E	1A
	文字			SYN		PN	RS	UC	EOT				CU3	DC4	NAK		SUB
40	ASCII	20	A0	A1	A2	A3	A4	A5	A6	A7	A8	D5	2E	3C	28	2B	7C
	文字	SP											.	<	(	+	
50	ASCII	26	A9	AA	AB	AC	AD	AE	AF	B0	B1	21	24	2A	29	3B	7E
	文字	&										!	\$	*	)	;	~
60	ASCII	2D	2F	B2	B3	B4	B5	B6	B7	B8	B9	CB	2C	25	5F	3E	3F
	文字	-	/										,	%	_	>	?
70	ASCII	BA	BB	BC	BD	BE	BF	C0	C1	C2	60	3A	23	40	27	3D	22
	文字										`	:	#	@	'	=	"
80	ASCII	C3	61	62	63	64	65	66	67	68	69	C4	C5	C6	C7	C8	C9
	文字		a	b	c	d	e	f	g	h	i						
90	ASCII	CA	6A	6B	6C	6D	6E	6F	70	71	72	5E	CC	CD	CE	CF	D0
	文字		j	k	l	m	n	o	p	q	r	^					
A0	ASCII	D1	E5	73	74	75	76	77	78	79	7A	D2	D3	D4	5B	D6	D7
	文字			s	t	u	v	w	x	y	z				[		
B0	ASCII	D8	D9	DA	DB	DC	DD	DE	DF	E0	E1	E2	E3	E4	5D	E6	E7
	文字														]		
C0	ASCII	7B	41	42	43	44	45	46	47	48	49	E8	E9	EA	EB	EC	ED
	文字	{	A	B	C	D	E	F	G	H	I						
D0	ASCII	7D	4A	4B	4C	4D	4E	4F	50	51	52	EE	EF	F0	F1	F2	F3
	文字	}	J	K	L	M	N	O	P	Q	R						
E0	ASCII	5C	9F	53	54	55	56	57	58	59	5A	F4	F5	F6	F7	F8	F9
	文字	¥		S	T	U	V	W	X	Y	Z						
F0	ASCII	30	31	32	33	34	35	36	37	38	39	FA	FB	FC	FD	FE	FF
	文字	0	1	2	3	4	5	6	7	8	9						

#### 4. Code Converter の使い方

表 4.3.8 数値コード体系を基調とするフィールド属性一覧

オープンシステム側 (文字形式データ)		変換モード指示パラメータ		
		mto変換用	otm変換用	
メインフレーム側 (パック形式データ) (ゾーン形式データ)	数値変換	パック形式変換	OPA1	OPA6
		ゾーン形式変換	OPA2	OPA7

(2) オープンシステムが「EUCコード」の場合の変換モード指示パラメータ

表 4.3.9 IBM 漢字コード体系を基調とするフィールド属性一覧

オープンシステム側 (EUCコード)		変換モード指示パラメータ		
		mto変換用	otm変換用	
メインフレーム側 (全角コード : IBM漢字) (半角コード : EBCDIC/EBCDIK)	漢字変換	全角のみ	OPC1	OPC6
	全/半角混在	半角に英小文字含む	OPC2	OPC7
〃 含まない		OPC3	OPC8	
英数文字変換	半角のみ	半角に英小文字含む	OPC4	OPC9
		〃 含まない	OPC5	OPCA

表 4.3.10 KEIS78 漢字コード体系を基調とするフィールド属性一覧

オープンシステム側 (EUCコード)		変換モード指示パラメータ		
		mto変換用	otm変換用	
メインフレーム側 (全角コード : KEIS78漢字) (半角コード : EBCDIC/EBCDIK)	漢字変換	全角のみ	OPD1	OPD6
	全/半角混在	半角にカナ文字含まない	OPD2	OPD7
〃 含む		OPD3	OPD8	
英数文字変換	半角のみ	半角にカナ文字含まない	OPD4	OPD9
		〃 含む	OPD5	OPDA

表 4.3.11 KEIS83/90 漢字コード体系を基調とするフィールド属性一覧

オープンシステム側 (EUCコード)		変換モード指示パラメータ		
		mto変換用	otm変換用	
メインフレーム側 (全角コード : KEIS83/90漢字) (半角コード : EBCDIC/EBCDIK)	漢字変換	全角のみ	OPE1	OPE6
	全/半角混在	半角にカナ文字含まない	OPE2	OPE7
〃 含む		OPE3	OPE8	
英数文字変換	半角のみ	半角にカナ文字含まない	OPE4	OPE9
		〃 含む	OPE5	OPEA

#### 4. Code Converter の使い方

(3) オープンシステムが「Unicode」の場合の変換モード指示パラメータ

表 4.3.12 IBM 漢字コード体系を基調とするフィールド属性一覧

オープンシステム側 (UNICODE) メインフレーム側 (全角コード: IBM 漢字) (半角コード: EBCDIC/EBCDIK)		変換モード指示パラメータ		
		mt0 変換用	otm 変換用	
漢字変換	全角のみ	OP61	OP66	
	全/半角混在	半角英小文字含む	OP62	OP67
		半角英小文字含まない	OP63	OP68
英数文字変換	半角のみ	半角英小文字含む	OP64	OP69
		半角英小文字含まない	OP65	OP6A

表 4.3.13 KEIS78 コード体系を基調とするフィールド属性一覧

オープンシステム側 (UNICODE) メインフレーム側 (全角コード: KEIS78) (半角コード: EBCDIC/EBCDIK)		変換モード指示パラメータ		
		mt0 変換用	otm 変換用	
漢字変換	全角のみ	OP71	OP76	
	全/半角混在	半角カナ文字含まない	OP72	OP77
		半角カナ文字含む	OP73	OP78
英数文字変換	半角のみ	半角カナ文字含まない	OP74	OP79
		半角カナ文字含む	OP75	OP7A

表 4.3.14 KEIS83/90 コード体系を基調とするフィールド属性一覧

オープンシステム側 (UNICODE) メインフレーム側 (全角コード: KEIS83/90) (半角コード: EBCDIC/EBCDIK)		変換モード指示パラメータ		
		mt0 変換用	otm 変換用	
漢字変換	全角のみ	OP81	OP86	
	全/半角混在	半角カナ文字含まない	OP82	OP87
		半角カナ文字含む	OP83	OP88
英数文字変換	半角のみ	半角カナ文字含まない	OP84	OP89
		半角カナ文字含む	OP85	OP8A

#### 4. Code Converter の使い方

表 4.3.15 JEF 漢字コード体系を基調とするフィールド属性一覧

オープンシステム側 (UNICODE)		変換モード指示パラメータ		
		mt0 変換用	otm 変換用	
メインフレーム側 (全角コード: JEF 漢字) (半角コード: EBCDIC/EBCDIK)				
漢字変換	全角のみ	OP91	OP96	
	全/半角混在	半角カナ文字含まない	OP92	OP97
		半角カナ文字含む	OP93	OP98
英数文字変換	半角のみ	半角カナ文字含まない	OP94	OP99
		半角カナ文字含む	OP95	OP9A

表 4.3.15-1 数値コード体系を基調とするフィールド属性一覧

オープンシステム側 (文字形式データ)		変換モード指示パラメータ	
		mt0変換用	otm変換用
メインフレーム側 (パック形式データ) (ゾーン形式データ)			
数値変換	パック形式変換	OPB1	OPB6
	ゾーン形式変換	OPB2	OPB7

## 4. Code Converter の使い方

### 4.3.3. パラメータ定義ファイルを使った「フィールド定義ファイル」との関連付け

変換操作対象となるDBデータファイルと「フィールド定義ファイル」を関連付けるため、ファイル変換ユーティリティが mto 変換または otm 変換するDBデータファイル（メインフレーム側 SAM ファイル と オープンシステム側 UNIX/PC ファイル の 1 セット）を指定するのに用いている「パラメータ定義ファイル」を使います。

「パラメータ定義ファイル」に対しては、ユーザー編集切口処理（Code Converter 処理）を施すことの宣言、および「フィールド定義ファイル」等の Code Converter 関連パラメータを追記します。  
 （「パラメータ定義ファイル」も、一定の書式に従って、ユーザーがテキストファイル形式で記述するものです。）

#### [追記要領]

次の2つのパラメータを追加記述してください。

- ① USER-EDIT 記述 → ユーザー編集切口処理(Code Converter 処理)を施すことを宣言する予約語キーワードです。既存のファイル変換ユーティリティ用パラメータの後方に、1 個以上の半角スペースを空けてこの通り記述してください。
- ② フィールド定義ファイル名称 および 編集オプションファイル名 の記述 → Code Converter 関連パラメータです。4.3.2節の説明に従って作成したフィールド定義ファイルの名称、および編集オプションファイルの名称(EditOpFile/EditOpEUC)に各々"./"（当該ファイルがカレントディレクトリ上に格納されていることを示すパス記述）を付加し、さらに両者を","(コンマ)で連結したものを ①の USER-EDIT の後に 1 個以上の半角スペースを空けて記述してください。("./" フィールド定義ファイル名称 "," 編集オプションファイル名 "./" の間にスペースを入れないでください)

[追記例] (△: 半角1バイト以上の空白を示す。)

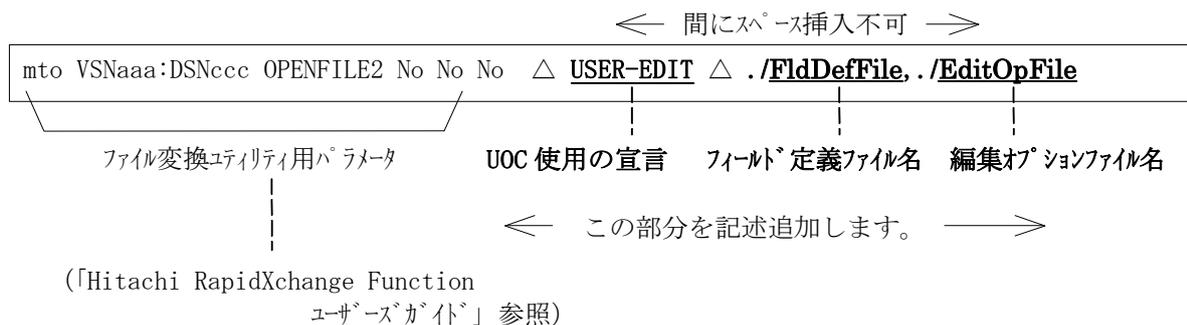


図 4.3.13 パラメータ定義ファイルを使った「フィールド定義ファイル」との関連付け

## 4. Code Converter の使い方

### 4.3.4. 編集オプションファイル

編集オプションファイルは、Code Converter で支援している変換様式(表 4.3.3～表 4.3.6、表 4.3.9～表 4.3.15 に示したフィールド属性)の全種類について詳細を定義するもので、オープン環境(UNIX/PC サーバ)のカレントディレクトリ内に格納して使います。

編集オプションファイルは、オープンシステム側の文字コードによって、ファイルが異なります。

- シフト JIS / JIS8 : EditOpFile
- EUC コード : EditOpEUC
- Unicode : EditOpUNI

未定義コードの置換に使用するスペースコードを独自の特殊コードに変更したい等の状況も考えられますので、この編集オプションファイルで定義している内容について簡単な説明をしておきます。

パラメータ定義ファイル

```
mt0 VSNaaa:DSNccc OPENFILE2 No No No USER-EDIT ./FldDefFile,./EditOpFile
```

編集オプションファイル名

./EditOpFile (編集オプションファイル) … コメントも含め、最大 999 行のテキスト形式ファイルです。

<pre>[KANJI:OP41] #...変換モード指示 = KANJI : OP41 の定義ラベルヘッダ JEFtoSJIS #...JEF 漢字コードからシフト JIS コードに変換する ./ #...コード変換テーブルは ./ ディレクトリに存在する JEFSJIS #...JEFSJIS.tbl というコード変換テーブルを使用する TRAN_DBLSPACE #...当該フィールド内に半角文字コードがあるなら EBCDIK コード # (か文字を含む)であり、スペース文字は全角スペースコードに変換する 20 #...半角未定義文字は X' 20' に置換する 8140 #...全角未定義文字は X' 8140' に置換する 20 #...半角/全角混在時、変換先領域が余れば X' 20' で充填する 8140 #...全角のみの時、変換先領域が余れば X' 8140' で充填する TRAN_DOUBLE #...当該フィールドは 2 バイト文字(漢字)のみのフィールドである [CHAR:OP45] #...変換モード指示 = CHAR : OP45 の定義ラベルヘッダ EBKJEFtoJIS8 #...JEF 版 EBCDIK から JIS8 コードに変換する(か文字を含む) ./ #...コード変換テーブルは ./ ディレクトリに存在する JEFSJIS #... JEFSJIS.tbl というコード変換テーブルを使用する 20 #...未定義文字は X' 20' に置換する 20 #...変換先領域が余れば X' 20' で充填する [] #...終了記述子</pre>	<p>変換モード指示 =KANJI : OP41 …JEF 漢字→シフト JIS 漢字変換 (全角のみ) の場合の指定例</p> <p>変換モード指示 = CHAR : OP45 …EBCDIK→JIS8 英数文字変換 (か文字を含む半角のみ) の場合の指定例</p>
---	--

図 4.3.14 編集オプションファイルによる変換モード指示の記述様式 概念図

#### 4. Code Converter の使い方

編集オプションファイルの中では、各行単位に必要な項目数(1 パラメタ)より後方および #文字以降をコメントとして扱っています。また、空白行もコメント扱いです。

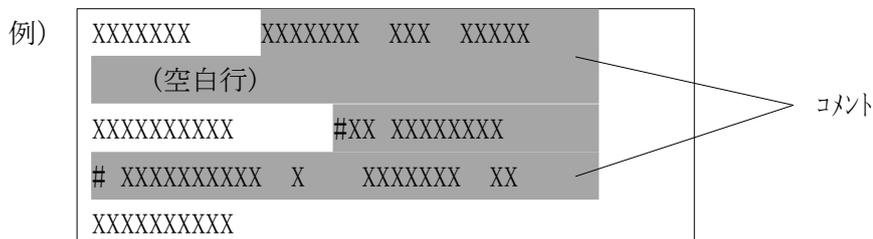


図 4.3.15 編集オプションファイル内のコメント記述

##### [変換モード指示の記述書式・概要]

- ① このファイルには、英数文字変換や漢字変換に係るモード指定などの情報を、フィールド定義ファイルの中で使用している変換モード指示(OPnn)の種類数分だけ、順不同で全て記述してあります。
- ② 各種の変換モード指示の記述情報は”[CHAR : OPnn]” ”[KANJI : OPnn]” ”[NUM : OPnn]”を記述した定義ヘッダ行で始まり、次の定義ヘッダ ”[xxxx : OPnn]”が現れるまでの間に定義されます。(最後は、終了記述子(“[]”の2文字)で終わります。)
- ③ 後に示す表 4.3.16～表 4.3.18で決められた個数 (=行数) のモード指定パラメタを1パラメタ/1行ずつ、表項番と同じ順番で定義しています。(この時、行数はパラメタを含む有効行のみで数え、コメントのみの行は数えていません。)
- ④ 最終行には、終了記述子(“[]”の2文字)があり、これをもって編集オプションファイルの終了と識別しています。
- ⑤ 1行1文で127カラム、999行まで(コメント含む)記述できます。

##### [編集オプションファイルの編集方法]

標準提供の変換指示パラメタを変更するか、コピーして新たな変換指示パラメタを作成し、パラメタの変更を行ってください。

[KANJI:OP41]	#...変換モード指示 = KANJI : OP41 の定義ヘッダ	
JEFtoSJIS	#...JEF 漢字コードからシフト JISコードに変換する	
./	#...コード変換テーブルは ./ ディレクトリに存在する	
JEFSJIS	#...JEFSJIS.tb1 というコード変換テーブルを使用する	
TRAN_DBLSPACE	#...当該フィールド内に半角文字コードがあるなら EBCDIKコード	
	# (か文字を含む)であり、スペース文字は全角スペースコードに変換する	
20	#...半角未定義文字は X' 20' に置換する	
8140	#...全角未定義文字は X' 8140' に置換する	
20	#...半角/全角混在時、変換先領域が余れば X' 20' で充填する	
8140	#...全角のみの時、変換先領域が余れば X' 8140' で充填する	
TRAN_DOUBLE	#...当該フィールドは2バイト文字(漢字)のみのフィールドである	
[KANJI:USR41]	#...変換モード指示 = KANJI : USR41 の定義ヘッダ	コピーして[KANJI:USR41]を作成
JEFtoSJIS	#...JEF 漢字コードからシフト JISコードに変換する	
./	#...コード変換テーブルは ./ ディレクトリに存在する	
JEFSJIS	#...JEFSJIS.tb1 というコード変換テーブルを使用する	
TRAN_DBLSPACE TRAN_SHIFT_28 TRAN_NOREC	#...当該フィールド内に半角文字コードがあるなら EBCDIK	追加したいパラメタを追記
	# (か文字を含む)であり、スペース文字は全角スペースコードに変換する	
20	#...半角未定義文字は X' 20' に置換する	
8140	#...全角未定義文字は X' 8140' に置換する	
20	#...半角/全角混在時、変換先領域が余れば X' 20' で充填する	
8140	#...全角のみの時、変換先領域が余れば X' 8140' で充填する	
TRAN_DOUBLE	#...当該フィールドは2バイト文字(漢字)のみのフィールドである	
[]	#...終了記述子	

表 4.3.16～表 4.3.18 に、Code Converter が使用している変換モード指示パラメタの一覧を示します。

#### 4. Code Converter の使い方

表 4.3.16 英数文字変換” [CHAR : OPnn]”用 変換コード指示パラメータ一覧 (パラメータ数: 6 個)

指定行	指示パラメータ	記述内容	意味	加文字		
1 行目	コード変換種別	m t o 方 向	EBCIBMtoJIS8	IBM版EBCDIC→JIS8コード変換を指示	含む	
			EBKIBMtoJIS8	IBM版EBCDIK→JIS8コード変換を指示	含む	
			EBCKS78toJIS8	KEIS78版EBCDIC→JIS8コード変換を指示	含まない	
			EBKKS78toJIS8	KEIS78版EBCDIK→JIS8コード変換を指示	含む	
			EBCKS90toJIS8	KEIS90版EBCDIC→JIS8コード変換を指示	含まない	
			EBKKS90toJIS8	KEIS90版EBCDIK→JIS8コード変換を指示	含む	
			EBCJEFtoJIS8	JEF版EBCDIC→JIS8コード変換を指示	含まない	
			EBKJEFtoJIS8	JEF版EBCDIK→JIS8コード変換を指示	含む	
			EBCIBMtoEUC	IBM版EBCDIC→EUCコード変換を指示	含む	
			EBKIBMtoEUC	IBM版EBCDIK→EUCコード変換を指示	含む	
			EBCKS78toEUC	KEIS78版EBCDIC→EUCコード変換を指示	含まない	
			EBKKS78toEUC	KEIS78版EBCDIK→EUCコード変換を指示	含む	
			EBCKS90toEUC	KEIS90版EBCDIC→EUCコード変換を指示	含まない	
			EBKKS90toEUC	KEIS90版EBCDIK→EUCコード変換を指示	含む	
			EBCIBMtoUNI	IBM版EBCDIC→Unicodeコード変換を指示	含む	
			EBKIBMtoUNI	IBM版EBCDIK→Unicodeコード変換を指示	含む	
			o t m 方 向	JIS8toEBCIBM	JIS8コード→IBM版EBCDIC変換を指示	含む
				JIS8toEBKIBM	JIS8コード→IBM版EBCDIK変換を指示	含む
	JIS8toEBCKS78	JIS8コード→KEIS78版EBCDIC変換を指示		含まない		
	JIS8toEBKKS78	JIS8コード→KEIS78版EBCDIK変換を指示		含む		
	JIS8toEBCKS90	JIS8コード→KEIS90版EBCDIC変換を指示		含まない		
	JIS8toEBKKS90	JIS8コード→KEIS90版EBCDIK変換を指示		含む		
	JIS8toEBCJEF	JIS8コード→JEF版EBCDIC変換を指示		含まない		
	JIS8toEBKJEF	JIS8コード→JEF版EBCDIK変換を指示		含む		
	EUCtoEBCIBM	EUCコード→IBM版EBCDIC変換を指示		含む		
	EUCtoEBKIBM	EUCコード→IBM版EBCDIK変換を指示		含む		
		EUCtoEBCKS78	EUCコード→KEIS78版EBCDIC変換を指示	含まない		
		EUCtoEBKKS78	EUCコード→KEIS78版EBCDIK変換を指示	含む		
EUCtoEBCKS90		EUCコード→KEIS90版EBCDIC変換を指示	含まない			
EUCtoEBKKS90		EUCコード→KEIS90版EBCDIK変換を指示	含む			
UNItoEBCIBM		Unicodeコード→IBM版EBCDIC変換を指示	含む			
UNItoEBKIBM		Unicodeコード→IBM版EBCDIK変換を指示	含む			
2 行目		テーブル格納ディレクトリ	英数文字列	コード変換テーブル格納ディレクトリの名称を指示		
3 行目		変換テーブル名称	英数文字列	当該ファイルで使用するコード変換テーブルの名称を指示		

#### 4. Code Converter の使い方

指定行	指示パラメータ	記述内容	意味	加文字
4行目	未定義置換コード	2桁の16進文字	未定義文字はこのコードで置換することを指示 UNICODE変換の場合：未定義置換コードの前に、0x00を付加したコードにて置換します。また、半角カコードは指定出来ません。 UNICODE以外の場合：指定された未定義置換コードにて、置換します	
5行目	フィールド充填コード	2桁の16進文字	変換先領域が余ればこのコードで充填することを指示 UNICODE変換の場合：フィールド充填コードの前に、0x00を付加したコードにて置換します。また、半角カコードは指定出来ません UNICODE以外の場合：指定されたフィールド充填コードにて、置換します	
6行目	コード変換モード指定	TRAN_NOREC	固定長データセットで、otm方向の変換時に0バイト長のデータを許可することを指示。 なお、TRAN_NORECを指定しない場合は6行目は必要ありません。	
		TRAN_CODECHK	未定義コードのログ出力を指示(省略時は、ログを出力しない) <b>(*3) (*4)</b>	
		TRAN_CODE_UNDEFCHK	未定義コードのログ出力を指示(省略時は、ログを出力しない) <b>(*3) (*4)</b>	
		TRAN_DATA_SIZECHK	OPEN側データの切り捨て情報のログ出力を指示 <b>(*2) (*4)</b>	
		TRAN_MAXIMUM	コード変換対象を全てのコード範囲にする。 <b>(*1)</b>	
		TRAN_MINIMUM	コード変換対象を、EBCDIC (TRAN_EBCDICの場合) or EBCDIK (TRAN_EBCDIKの場合) とJIS第1水準/第2水準の範囲に限定して変換する。それ以外のコードは、未定義コードに置換する。TRAN_MAXIMUMと比較して変換性能が良い。 <b>(*1)</b>	

(\*1) EUCコード変換のときのみ有効

(\*2) 表 4.3.17の(\*6)を参照してください。

(\*3) 表 4.3.17の(\*4)を参照してください。

(\*4) 指定するにはオプション製品 Code Converter-CODECHK が必要です。

#### 4. Code Converter の使い方

表 4.3.17 漢字変換” [KANJI : OPnn]”用 変換モード指示パラメータ一覧 (パラメータ数 : 9 個)

指定行	指示パラメータ	記述内容	意味	
1 行目	コード変換種別	m t o 方 向	IBMtoSJIS	全角文字はIBM漢字→シフトJISコード変換を指示
			JEFtoSJIS	全角文字はJEF漢字→シフトJISコード変換を指示
			KS78toSJIS	全角文字はKEIS78漢字→シフトJISコード変換を指示
			KS90toSJIS	全角文字はKEIS83/90漢字→シフトJISコード変換を指示
			IBMtoEUC	全角文字はIBM漢字→EUCコード変換を指示
			KS78toEUC	全角文字はKEIS78漢字→EUCコード変換を指示
			KS90toEUC	全角文字はKEIS83/90漢字→EUCコード変換を指示
			IBMtoUNI	全角文字はIBM漢字→Unicode変換を指示
			JEFtoUNI	全角文字はJEF漢字→Unicode変換を指示
			KS78toUNI	全角文字はKEIS78漢字→Unicode変換を指示
			KS90toUNI	全角文字はKEIS83/90漢字→Unicode変換を指示
			o t m 方 向	SJIStoIBM
		SJIStoJEF		全角文字はシフトJISコード→JEF漢字変換を指示
		SJIStoKS78		全角文字はシフトJISコード→KEIS78漢字変換を指示
		SJIStoKS90		全角文字はシフトJISコード→KEIS83/90漢字変換を指示
		EUCtoIBM		全角文字はEUCコード→IBM漢字変換を指示
		EUCtoKS78		全角文字はEUCコード→KEIS78漢字変換を指示
		EUCtoKS90		全角文字はEUCコード→KEIS83/90漢字変換を指示
		UNItoIBM		全角文字はUnicode→IBM漢字変換を指示
		UNItoJEF	全角文字はUnicode→JEF漢字変換を指示	
UNItoKS78	全角文字はUnicode→KEIS78漢字変換を指示			
UNItoKS90	全角文字はUnicode→KEIS83/90漢字変換を指示			
2 行目	テーブル格納ディレクトリ	英数文字列	コード変換テーブル格納ディレクトリの名称を指示	
3 行目	変換テーブル名称	英数文字列	当該ファイルで使用されるコード変換テーブルの名称を指示	
4 行目	コード変換モード指定	TRAN_EBCDIC	半角文字があるならばカ文字を含まないことを指示	
		TRAN_EBCDIK	半角文字があるならばカ文字を含むことを指示	
		TRAN_SGLSPACE	スペースは半角スペースコードに変換することを指示 TRAN_DBLSPACEと共に指定できない	
		TRAN_DBLSPACE	スペースは全角スペースコードに変換することを指示 TRAN_SGLSPACEと共に指定できない	
		TRAN_MIXED_SPACE	スペースは半角/全角スペースコードに変換することを指示 TRAN_SGLSPACE, TRAN_DBLSPACEと指定時は当指定が優先する (KEIS⇔SJISコード変換でのみ有効)	
		TRAN_START_SINGLE	当該ファイルは半角文字から開始することを指示(*1)	
		TRAN_START_DOUBLE	当該ファイルは全角文字から開始することを指示(*1)	
		TRAN_SHIFT_NO	シフトコードを挿入しないことを指示	
		TRAN_SHIFT_28	シフトコードを0x28にすることを指示(JEFコードでのみ有効)	
		TRAN_SHIFT_38	シフトコードを0x38にすることを指示(JEFコードでのみ有効)	
		TRAN_SHIFT_SPACE	全角文字列中の半角スペースに対してもシフトコードを付けることを指示(*2)	

#### 4. Code Converter の使い方

指定行	指示パラメータ	記述内容	意味
		TRAN_SHIFT_DBLCHAR	OTM 変換時にコード変換フィールドに全角文字列がある場合、その全角文字列の前後にシフトコードを付けることを指示 (全/半角混在の漢字変換を指示したときのみ有効) <b>(*2) (*3)</b>
		TRAN_NOREC	固定長データセットで、otm方向の変換時に0バイト長のデータを許可することを指示。
		TRAN_NOPAD	SO/SI削除分フィールド充填コードを充填しないことを指示
		TRAN_SOSI_PAIRCHK	SO/SIのペアチェック結果をログファイルに出力を指示 <b>(*7)</b>
		TRAN_CODECHK	未定義コードのログ出力を指示(省略時は、ログを出力しない) <b>(*4) (*7)</b>
		TRAN_CODE_UNDEFCHK	未定義コードのログ出力を指示(省略時は、ログを出力しない) <b>(*4) (*7)</b>
		TRAN_DATA_SIZECHK	OPEN側データの切り捨て情報のログ出力を指示 <b>(*6) (*7)</b>
		TRAN_MAXIMUM	コード変換対象を全てのコード範囲にする。 <b>(*5)</b>
		TRAN_MINIMUM	コード変換対象を、EBCDIC (TRAN_EBCDICの場合) or EBCDIK (TRAN_EBCDIKの場合) とJIS第1水準/第2水準の範囲に限定して変換する。それ以外のコードは、未定義コードに置換する。TRAN_MAXIMUMに比較して変換性能が良い。 <b>(*5)</b>
5行目	未定義置換コード 半角	2桁の16進文字	半角の未定義文字はこのコードで置換することを指示 UNICODE変換の場合：未定義置換コードの前に、0x00を付加したコードにて置換します。また、半角がコードには指定出来ません。 UNICODE以外の場合：指定された未定義置換コードにて、置換します。
6行目	未定義置換コード 全角	4桁の16進文字	全角の未定義文字はこのコードで置換することを指示
7行目	フィールド充填コード 半角	2桁の16進文字	半角/全角混在時、変換先領域が余ればこのコードで充填することを指示 UNICODE変換の場合：フィールド充填コードの前に、0x00を付加したコードにて置換します。また、半角がコードには指定出来ません UNICODE以外の場合：指定されたフィールド充填コードにて、置換します。
8行目	フィールド充填コード 全角	4桁の16進文字	全角のみの時、変換先領域が余ればこのコードで充填することを指示
9行目	半角/全角混在有無	TRAN_DOUBLE	全角(2Byteコード)のみのフィールドであることを指示
		TRAN_MIX	半角/全角が混在するフィールドであることを指示

(\*1)フィールド先頭がシフトイン/シフトアウトコードから始まる場合には、本指示は無視され、シフトコードが優先されます。

#### 4. Code Converter の使い方

(\*2) TRAN\_SHIFT\_SPACE と TRAN\_SHIFT\_DBLCHAR の相違を以下に示す。

レコード形式	TRAN_SHIFT_SPACE	TRAN_SHIFT_DBLCHAR
固定長	OTM 変換時に全角文字列中の半角スペースに対してシフトコードを付ける。	固定長/可変長で OTM 変換時に全角文字列の前後にシフトコードを付ける。
可変長	OTM 変換時に全角文字列中の半角スペースに対してシフトコードを付ける。 注) 可変長レコードを OTM 変換時に変換フィールド終端の文字列が全角文字列に対してシフト IN コード (X'0F') が付加されない。	

また、TRAN\_SHIFT\_SPACE と TRAN\_SHIFT\_DBLCHAR を共に指定した場合は、TRAN\_SHIFT\_DBLCHAR が有効となります。

(\*3) 出力フィールドの後端に全角文字列をコード変換したとき、出力フィールドの最後が漢字文字列の途中で終わる場合はシフトコード (X'0F') が付加されません (下の表を参照)。したがって、メインフレーム側の出力フィールド長を以下の計算値を超える十分な長さにしてください。

- OPEN 側の入力フィールド長 + 入力フィールドに含まれる漢字フィールド数 × n  
n : シフトコード分 (IBM, JEF の場合は 2、KEIS の場合は 4)

全角文字で終わるケース	全角文字の途中で終わるケース
1-----8-----15 Byte OPENデータ: abc あい de <u>うえお</u> 16Byteへ変換: ↓ 「え」で終わり、シフトコードは付加されない。	1-----8-----15 Byte OPENデータ: abc あい de <u>うえお</u> 17Byteへ変換: ↓ 「お」の途中で終わると、シフトコードが付加される。
1-----9-----16 Byte M/Fデータ: abc▲あい▼de▲うえ	1-----9-----17 Byte M/Fデータ: abc▲あい▼de▲うえ▼

凡例. ▲: シフト OUT (X'0E')、▼: シフト IN (X'0F')

- ①左の例では、「え」が出力フィールドの終端で終わるため、「お」が切り捨てられてシフト IN コードが付加されない。
- ②右の例では「お」の途中で終わるため、「お」が切り捨てられて「お」の右1バイト分にシフト IN コードが付加される。

(\*4) TRAN\_CODECHK と TRAN\_CODE\_UNDEFchk の相違を以下に示す。

項目	TRAN_CODECHK	TRAN_CODE_UNDEFchk
未定義コードの検出がなかった	ログファイルを作成する	ログファイルを作成しない
ログファイルが1MBを超えた	処理は異常終了する。	ログ出力を停止して処理を継続する。
ログ出力形式	相違点なし	相違点なし

(注) ログファイルは、上書きモードで作成する。

(\*5) EUC コード変換のときのみ有効

#### 4. Code Converter の使い方

(\*6) OTM 変換の出力フィールドが全角文字列で終わる変換の場合は、以下に示す変換状態になります。

① 全角文字の最終文字で終わるケース	② 全角文字の途中で終わるケース
1-----8-----15 Byte OPENデータ: abcあいde <u>うえお</u> 18Byteへ変換: ”お”で終わるケース ↓ 1-----9-----18 Byte M/Fデータ: abc▲あい▼de▲ <u>うえお</u>	1-----8-----15 Byte OPENデータ: abcあいde <u>うえお</u> 17Byteへ変換: ”お”の途中で終わるケース ↓ 1-----9-----17 Byte M/Fデータ: abc▲あい▼de▲ <u>うえ△</u>

**【説明】**

- ①のケースでは、”お”が出力フィールドの終端で終わるため、シフトIN(X'0F)コードは付加されない。
- ②のケースでは、”お”の途中で終わるため、1バイト空白を付加して”お”は切り捨てられる。

**【TRAN\_SHIFT\_DBLCHARとの関連】**

TRAN\_DATA\_SIZECHKとTRAN\_SHIFT\_DBLCHARとの関連を以下に示す。

項目	ケース	TRAN_SHIFT_DBLCHARなし	TRAN_SHIFT_DBLCHARあり
シフトコード	①	付加しない	付加しない
	②	付加しない	付加する
”お”の1バイト部分	②	空白を付加する “abc▲あい▼de▲ <u>うえ△</u> ”	シフトIN(X'0F)コードを付加する “abc▲あい▼de▲ <u>うえ▼</u> ”
データ切捨てログ情報	①	出力しない	出力する
	②	出力する	出力する
切り捨てた長さ	①	—	0 (**1)
	②	2 (**2)	2 (**2)

(\*\*1) シフトコードだけが付加されなかった状態を示す長さである。

(\*\*2) “お”を切り捨てた長さである。

凡例. ▲: シフトOUT(X'0E)、▼: シフトIN(X'0F)、△: 空白

本オプションを用いてOTMコード変換したときにデータの切り捨てが発生した場合、全レコードのコード変換が終了したときにエラーメッセージ (-767) を出力し異常終了する。

(注) 当エラーメッセージは、Cross-OS File Exchange Function Ver. 01-04-60/21より適用され、それ以前のバージョンでは正常終了しますのでご注意願います。

(\*7) 指定するにはオプション製品 Code Converter-CODECHK が必要です。

#### 4. Code Converter の使い方

表 4.3.18 数値変換” [NUM:OPnn]”用 変換モード指示パラメータ一覧 (パラメータ数: 7個)

指定行	指示パラメータ	記述内容	意味	
1行目	コード変換種別	mto方向	PACKtoCHAR	バック形式→文字形式に変換を指示
			ZONEtoCHAR	ゾーン形式→文字形式に変換を指示
			PACKtoUNI	バック形式→Unicode文字形式に変換を指示
			ZONEtoUNI	ゾーン形式→Unicode文字形式に変換を指示
		otm方向	CHARtoPACK	文字形式→バック形式に変換を指示
			CHARtoZONE	文字形式→ゾーン形式に変換を指示
			UNItoPACK	Unicode文字形式→バック形式に変換を指示
			UNItoZONE	Unicode文字形式→ゾーン形式に変換を指示
2行目	符号位置	TRAN_SIGN_F	先頭に符号を付ける	
		TRAN_SIGN_E	終端に符号を付ける	
		TRAN_SIGN_V	有効桁の前に符号を付ける (0サブレスでのみ有効)	
		TRAN_SIGN_NON	符号なし (*1)	
3行目	小数点桁数	1桁の10進文字	小数点桁数0~9を指示	
4行目	0サブレスの処置	TRAN_NON_SUPPRESS	0サブレスしないことを指示	
		TRAN_ZERO_SUPPRESS	0サブレスすることを指示	
5行目	プラス符号種別(*)	TRAN_SIGN_PLUS_C	プラス符号をCにすることを指示 (otmでのみ有効)	
		TRAN_SIGN_PLUS_A	プラス符号をAにすることを指示 (otmでのみ有効)	
		TRAN_SIGN_PLUS_E	プラス符号をEにすることを指示 (otmでのみ有効)	
		TRAN_SIGN_PLUS_F	プラス符号をFにすることを指示 (otmでのみ有効)	
6行目	マイナス符号種別(*)	TRAN_SIGN_MINUS_D	マイナス符号をDにすることを指示 (otmでのみ有効)	
		TRAN_SIGN_MINUS_B	マイナス符号をBにすることを指示 (otmでのみ有効)	
7行目	オールスペースの処置	TRAN_ALLSPACE_NORMAL	otmの場合、オールスペースはエラー(-758)とする mtoの場合、オール0はそのまま0に変換する	
		TRAN_ALLSPACE_ZERO	otmの場合、オールスペースは0に変換する mtoの場合、オール0はオールスペースに変換する	
8行目	バックデータの符号処置	TRAN_PACK_SIGN_NON	符号なしバックデータを変換する (*2) 当指示パラメータを指定しない場合は省略可能です。	

(\*) otmの場合、有効なパラメータ。mtoの場合、符号は自動判別します。

(\*1) TRAN\_SIGN\_NON は、バック形式データに符号があり、文字形式データに符号を付けない時に指定します。

(\*2) TRAN\_PACK\_SIGN\_NON は、バック形式データに符号を付けない時に指定します。

このパラメータは、バック形式データの数値変換のみ有効で、オープン側の符号は、「符号位置」の指定に従って処理します。

## 5. 制限事項・その他

### 5. 制限事項・その他

Code Converter が動作可能な、メインフレームとサーバの一覧を次に示します。

#### (1) 対象メインフレームとサーバ

日立マルチプラットフォーム RAID 及びファイル変換ユーティリティがサポートしている機種&OSのうち、Code Converter は、表 5.1 に示す機種&OSをサポート対象とします。

表 5.1 Code Converter 対象メインフレームとサーバ/PC 一覧

項番	区分	メーカー	機種 (OS)	備考
1	メインフレーム	日立	VOS3/FS, VOS3/AS	
2		IBM	MVS/ESA, MVS/XA	
3		富士通	MSP/EX	
4	UNIX サーバ	HP	HP-UX 11.0 以降	
5		IBM	AIX 5.2 以降	
6		SUN	Solaris 8 以降	
7		Red Hat	Linux AS3.0 以降	
8			Linux RHEL AS3.0 以降、AS4.0(Update5) 以降 AS5.1 以降	*1
9		SuSE	Linux ES9 以降	
10	PC	Microsoft	Windows Server 2003/2008/2008 R2	

対応 OS の詳細については弊社へ個別にご相談ください。

\*1 64ビット版は Itanium(IA64) 対応です。

なお、Code Converter を使用する場合には、ファイル変換ユーティリティが必要です。

#### (2) 使用上の注意

ライセンスコード入力前に mto を行うと 0 バイトのファイルが作成されます。その場合、ライセンスを入力後に再度 mto を行なってください。

### 注意

**Code Converter 提供の初期(codetran.ini)ファイルは、FCU を実行するディレクトリに置かなければなりません。**

## 6. Code Converter エラー情報

### 6. Code Converter エラー情報

#### 6.1. Code Converter エラーコード/エラーメッセージ

表 6.1.1 Code Converter のエラーコード・エラーメッセージ

エラーコード	エラーメッセージ
-603	・"Parameter : CODE_CONV select error(USER-EDIT)" Uoc機能使用時に使えない、CODE_CONVの指定があります。 ⇒ CODE_CONVの指定をしないでください。
-604	・"Parameter : DELIMITER select error(USER-EDIT)" Uoc機能使用時に使えない、DELIMITER の指定があります。 ⇒ DELIMITER の指定をしないでください。
-605	・"Parameter : EMPTY select error(USER-EDIT)" Uoc 機能使用時に使えない、Empty の指定があります。 ⇒EMPTY 指定をしないでください。
-606	・"Parameter : RDW select error(USER-EDIT)" Uoc 機能使用時に使えない、RDW の指定があります。 ⇒RDW 指定をしないでください。
-607	・"Parameter : DELIMITER (No) select error(USER-EDIT)" Uoc 機能使用 (otm)時にデリミタ指定がありません。 ⇒フィールド定義ファイルでデリミタを指定してください。
-615	・"Uoc information buf : Allocation error" Uoc制御情報テーブル用メモリの確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-616	・"Edit log buf : Allocation error" Uoc編集切口用ロギングメモリの確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-617	・"Output log buf : Allocation error" Uoc出力切口/パイプ出力切口用ロギングメモリの確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-640	・"Null pointer (fcuocRCedit) error" fcuからNULLポインタが渡されました。 ⇒ fcuのマークが正しくできているか確認してください。
-641	・"Field definition file open error" フィールド定義ファイルのオープンに失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-642	・"Edit option file open error" 編集オプションファイルのオープンに失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-643	・"Field definition file read error" フィールド定義ファイルのリードに失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-644	・"Edit option file read error" 編集オプションファイルのリードに失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-645	・"Field definition file : Top point/field length over flow error" フィールド定義ファイルパラメータで、先頭開始位置またはフィールドサイズが編集元バッファ長よりも大きくなっています。 ⇒ フィールド定義ファイルを見直してください。
-646	・"Field definition file : Output field size over flow" 出力フィールド長が1レコード最大長(32KB)をオーバーしています。 ⇒ フィールド定義ファイルを見直してください。
-647	・"Edited size error" 編集後のトータルコードサイズが0バイトです。 ⇒ フィールド定義ファイルを見直してください。

## 6. Code Converter エラー情報

エラーコード	エラーメッセージ
-648	<p>・"Memory allocate error"            トレース用メモリの確保が失敗しました。            ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-649	<p>・"Record format error"            データセットが固定長になっていません。            ⇒ データセットの指定をチェックしてください。</p>
-650	<p>・"Field definition file : Top point error"            フィールド定義ファイルパラメータの先頭開始位置の値が不正です。            ⇒ フィールド定義ファイルを見直してください。</p>
-651	<p>・"Field definition file : Field size error"            フィールド定義ファイルパラメータのフィールドサイズが不正です。            ⇒ フィールド定義ファイルを見直してください。</p>
-652	<p>・"Field definition file : Item count error"            フィールド定義ファイルパラメータのfld情報の項目数が不足しています。            ⇒ フィールド定義ファイルを見直してください。</p>
-653	<p>・"Field definition file : Edit kind error"            フィールド定義ファイルパラメータの属性定義値が不正です。            ⇒ フィールド定義ファイルを見直してください。</p>
-655	<p>・"Memory allocate error"            フィールド情報読み込み用メモリの確保が失敗しました。            ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-656	<p>・"Field definition file : Character is not #/fld/end error"            フィールド定義ファイルパラメータで fld/end/#以外を指定しています。            ⇒ フィールド定義ファイルを見直してください。</p>
-657	<p>・"Memory allocate error"            フィールド定義ファイル読み込み用メモリの確保が失敗しました。            ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-659	<p>・"Memory allocate error"            編集オプションファイル読み込み用メモリの確保が失敗しました。            ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-660	<p>・"Edit option file : Invalid format error"            編集オプションファイルの変換モード指示ラベルまたは終了記述子が不正です。            ⇒ 編集オプションファイルを見直してください。(']'で閉じられていない。)</p>
-661	<p>・"Edit option file : Information over flow"            編集オプションファイルの情報量が多すぎるか '['がありません。            ⇒ 編集オプションファイルを見直してください。</p>
-662	<p>・"Edit option file : Invalid format error"            編集オプションファイルの変換モード指示ラベルが不正です。            ⇒ 編集オプションファイルを見直してください。</p>
-663	<p>・"Edit option file : Information count error"            編集オプションファイルのパラメータ数が不足しています。            ⇒ 編集オプションファイルを見直してください。</p>
-664	<p>・"Field definition file : Non end mark or non fld area error"            フィールド定義ファイルに "end" マークが存在しないか "fld" から始まるフィールドがありません。            ⇒ フィールド定義ファイルを見直してください。</p>
-665	<p>・"Edit option file : Edit table option error"            フィールド定義ファイルで指示された変換モードが編集オプションファイルで定義されていません。            ⇒ 編集オプションファイル、または、フィールド定義ファイルを見直してください。</p>
-666	<p>・"Edit option file : Edit kind or end mark error"            編集Blockが存在しないかまたは "end" マークがありません。            ⇒ 編集オプションファイルを見直してください。</p>

## 6. Code Converter エラー情報

エラーコード	エラーメッセージ
-667	<p>・"Memory allocate error" フィールド情報取得用メモリの確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-668	<p>・"Memory allocate error" フィールド定義ファイル名取得用メモリの確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-669	<p>・"Memory allocate error" フィールド情報取得用メモリの確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-670	<p>・"Memory allocate error" 編集オプションファイル名取得用メモリの確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-671	<p>・"Edit option file : Code exchange kind (Zenkaku) error" 編集オプションファイルの全角変換処理でコード変換種別値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-672	<p>・"Edit option file : 1byte code (Zenkaku) error" 編集オプションファイルの全角変換処理で1バイト未定義置換コードの値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-673	<p>・"Edit option file : 2byte code (Zenkaku) error" 編集オプションファイルの全角変換処理で2バイト未定義置換コードの値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-674	<p>・"Edit option file : Code exchange option kind (Zenkaku) error" 編集オプションファイルの全角変換処理でコード変換モード指定の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-675	<p>・"Edit option file : Data string type kind (Zenkaku) error" 編集オプションファイルの全角変換処理で半角/全角混在有無の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-676	<p>・"Kanji exchange : Parameter error" 全角変換処理または、半角変換処理でパラメータが不正です。 ⇒ fcuのマークが正しくできているか確認してください。</p>
-677	<p>・"Kanji exchange : NOT COpen error" 全角変換処理または、半角変換処理がCopen (前処理) で不正になりました。 ⇒ 編集オプションファイルの格納ディレクトリを見直してください。</p>
-678	<p>・"Kanji exchange : Can not find exchange table error" 全角変換処理または、半角変換処理で変換テーブルが存在しません。 ⇒ 変換テーブルを確認してください。</p>
-679	<p>・"Kanji exchange : Interface area error" 全角変換処理または、半角変換処理で変換ルーチンのインタフェース不正になりました。 ⇒ fcuのマークが正しくできているか確認してください。</p>
-680	<p>・"Kanji exchange : Invalid exchange table error" 全角変換処理または、半角変換処理で変換テーブルが不正です。 ⇒ 変換テーブルを確認してください。</p>
-681	<p>・"Kanji exchange : Can not Area get for exchange table error" 全角変換処理または、半角変換処理で変換テーブルを読み込む領域の確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-682	<p>・"Kanji exchange : Exchange table read error" 全角変換処理または、半角変換処理で変換テーブルのリードがエラーになりました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-683	<p>・"Kanji exchange : Exchange table I/O error" 全角変換処理または、半角変換処理で変換テーブル入出力に失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>

## 6. Code Converter エラー情報

エラーコード	エラーメッセージ
-684	・"Kanji exchange : Unmatched exchange kind and exchange table error" 全角変換処理または、半角変換処理で変換種別と変換テーブルがあっていません。 ⇒ 変換テーブルを確認してください。
-685	・"Kanji exchange : Broken exchange table error" 全角変換処理または、半角変換処理で変換テーブルの領域が壊れています。 ⇒ <code>cnv</code> を使って変換テーブルの内容を表示し確認してください。
-686	・"Kanji exchange : Memory allocate error" 全角変換処理または、半角変換処理でメモリの確保に失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-687	・"Kanji exchange : Broken environment area error" 全角変換処理または、半角変換処理で環境設定してある領域が壊れています。 ⇒ <code>fcu</code> のマークが正しくできているか確認してください。
-688	・"Kanji exchange : Exchange length error" 全角変換処理または、半角変換処理で変換長が不正のためにエラーになりました。 ⇒ フィールド定義ファイルを見直してください。(変換を行なうフィールドサイズが0のため。)
-689	・"Kanji exchange : Exchange code read/set non area error" 全角変換処理または、半角変換処理で変換するコードを参照/設定する領域が指定されていないため、エラーになりました。 ⇒ <code>fcu</code> のマークが正しくできているか確認してください。
-690	・"Kanji exchange : Other error" 全角変換処理または、半角変換処理で予期せぬエラーが発生しました。 ⇒ フィールド定義ファイルおよび編集オプションファイルを見直してください。
-691	・"Conflicting with input data" 変換指定内容と入力データが矛盾しているため、変換できませんでした。 ⇒ 入力データおよび編集オプションファイルを見直してください。
-693	・"Edit option file : Code exchange kind (Hankaku) error" 編集オプションファイルの 半角変換処理でコード変換種別の値が不正です。 ⇒ 編集オプションファイルを見直してください。
-694	・"Edit option file : lbyte code (Hankaku) error" 編集オプションファイルの 半角変換処理で1バイト未定義置換コードの値が不正です。 ⇒ 編集オプションファイルを見直してください。
-695	・"Edit option file : Directory (Zenkaku) error" 編集オプションファイルの 全角変換処理でテーブル格納ディレクトリが未指定です。 ⇒ 編集オプションファイルを見直してください。
-696	・"Edit option file : Directory (Hankaku) error" 編集オプションファイルの 半角変換処理でテーブル格納ディレクトリが未指定です。 ⇒ 編集オプションファイルを見直してください。
-697	・"Edit option file : Information space error" 編集オプションファイルに情報がありません。 ⇒ 編集オプションファイルを見直して変換情報を設定してください。
-698	・"Field definition file : Information space error" フィールド定義ファイルにフィールド情報がありません。 ⇒ フィールド定義ファイルを見直してフィールド情報を設定してください。
-700	・"Memory allocate error" フィールド情報切り出し用メモリ確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。
-701	・"Edit option file : lbyte umeji (Zenkaku) error" 編集オプションファイルの 全角変換処理でフィールド充填コード半角が不正です。 ⇒ 編集オプションファイルを見直してください。
-702	・"Edit option file : 2byte umeji (Zenkaku) error" 編集オプションファイルの 全角変換処理でフィールド充填コード全角が不正です。 ⇒ 編集オプションファイルを見直してください。

## 6. Code Converter エラー情報

エラーコード	エラーメッセージ
-703	<p>・"Edit option file : lbyte umeji (Hankaku) error"</p> <p>編集オプションファイルの 半角変換処理でフィールド充填コード半角が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-704	<p>・"Edit option file : Exchange option exclusive error"</p> <p>編集オプションファイルの変換モード指定が排他チェックエラーになりました。 ⇒ 編集オプションファイルの変換オプションを見直してください。</p>
-705	<p>・"Memory allocate error"</p> <p>フィールド情報サブテーブル用メモリ確保が失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-706	<p>・"Field definition file : Too many data"</p> <p>フィールド定義ファイルのサイズが大きすぎます。 ⇒ フィールド定義ファイルの行数をコメントも含め1000行以内にしてください。</p>
-707	<p>・"Edit option file : Too many data"</p> <p>編集オプションファイルのサイズが大きすぎます。 ⇒ 編集オプションファイルを見直してください。("[ ]"が999行目以内に見つかりません)</p>
-708	<p>・"Edit option file : 1 line data over flow"</p> <p>編集オプションファイルの1行の文字数が長すぎます。 ⇒1行127バイト以内にしてください。</p>
-710	<p>・"Edit option file : Code exchange option kind (Hankaku) error"</p> <p>編集オプションファイルの半角変換処理でコード変換モード指定の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-720	<p>・"Field definition file : Delimiter or Padding error"</p> <p>フィールド定義ファイルのデリミタまたはパディング指定が不正です。 ⇒ フィールド定義ファイルのデリミタまたはパディング指定を見直してください。</p>
-721	<p>・"Field define file : 1 line data over flow"</p> <p>フィールド定義ファイルの1行の文字数が長すぎます。 ⇒1行127バイト以内にしてください。</p>
-722	<p>・"Field define file : Total field length unmatched"</p> <p>フィールド定義ファイルの入力フィールド長の合計がデコード長に一致しません。 ⇒フィールド定義ファイルの入力フィールド長の合計を見直してください。</p> <ul style="list-style-type: none"> <li>・固定長の場合は、入力フィールド長の合計 = レコード長を確認する。</li> <li>・可変長の場合は、入力フィールド長の合計 &lt;= レコード長を確認する。</li> </ul>
-723	<p>・"Field define file : Output field size error"</p> <p>フィールド定義ファイルの出力フィールド長が不正です。 ⇒フィールド定義ファイルを見直してください。</p>
-724	<p>・"Field define file : Total output field length error"</p> <p>フィールド定義ファイルの出力フィールド長の合計が不正です。 ⇒フィールド定義ファイルを見直してください。</p>
-725	<p>・"Field definition file : Padding cannot be specified (Un-supporting)"</p> <p>フィールド定義ファイルの中で指定されたパディングはサポートされていません。 ⇒ FAL/FCU のバージョンを見直してください (Ver. 01-01-45 以降サポート)。</p>
-726	<p>・"Field definition file : UNICODE delimiter cannot be specified (Un-supporting)"</p> <p>フィールド定義ファイルの中で指定された UNICODE デリミタは、サポートされていません。 ⇒ FAL/FCU のバージョンを見直してください (Ver. 01-01-49 以降サポート)。</p>
-740	<p>・"otm : Delimiter check error"</p> <p>フィールド定義ファイルのデリミタ指定が不正です。 ⇒ フィールド定義ファイルのデリミタ指定を見直してください。</p>
-745	<p>・" Log file close error"</p> <p>ログファイルのクローズが失敗しました。 ⇒ HelpメニューのErrorでシステムエラーコードを表示し、OSのマニュアルによりエラー内容をチェックしてください。</p>
-749	<p>・"Edit option file : Non pack sign (Number) error"</p> <p>編集オプションファイルの数値変換指示パラメータで「バックデータの符号処置」の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>

## 6. Code Converter エラー情報

エラーコード	エラーメッセージ
-750	<p>・"Edit option file : Code exchange kind (Number) error" 編集オプションファイルの数値変換処理でコード変換種別の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-751	<p>・"Edit option file : Sign position (Number) error" 編集オプションファイルの数値変換処理で符号位置の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-752	<p>・"Edit option file : Decimal point number (Number) error" 編集オプションファイルの数値変換処理で小数点以下の桁数の値 (0～9) が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-753	<p>・"Edit option file : Zero suppress (Number) error" 編集オプションファイルの数値変換処理で0サプレス有無の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-754	<p>・"Edit option file : Sign plus (Number) error" 編集オプションファイルの数値変換処理でプラス符号の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-755	<p>・"Edit option file : Sign minus (Number) error" 編集オプションファイルの数値変換処理でマイナス符号の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-756	<p>・"Edit option file : Zero suppress (TRAN_NON_SUPPRESS) error" 編集オプションファイルの数値変換処理で「0サプレスしない」が指定されています。 ⇒ 有効桁の前に符号を付ける (TRAN_SIGN_V) の場合、0サプレスしないは指定できません。</p>
-757	<p>・"Field define file : Input/Output field size error" 数値変換処理でフィールド定義ファイルの入出力フィールド長が短いため、変換できませんでした。 ⇒ フィールド定義ファイルを見直してください。</p>
-758	<p>・"Number exchange : Sign position (Number) error with input data" 数値変換処理で編集オプションファイルの符号位置が入力データの該当する位置にないため、変換できませんでした。 ⇒ 入力データおよび編集オプションファイルを見直してください。</p>
-759	<p>・"Number exchange : Decimal point (Number) error with input data" 数値変換処理で編集オプションファイルの小数点桁数が入力データの該当する位置にないため、変換できませんでした。 ⇒ 入力データおよび編集オプションファイルを見直してください。</p>
-760	<p>・"Number exchange : Input data (Number) error" 数値変換処理で入力データの内容が不正であるため、変換できませんでした。 ⇒ 入力データを見直してください。</p>
-761	<p>・"Edit option file : All space (Number) error" 編集オプションファイルの数値変換処理でオールスペースの処置の値が不正です。 ⇒ 編集オプションファイルを見直してください。</p>
-762	<p>・" Log file open error" ログファイルのオープンが失敗しました。 ⇒ Help メニューの Error でシステムエラーコードを表示し、OS のマニュアルによりエラー内容をチェックしてください。</p>
-763	<p>・" Log file write error" ログファイルへの書き込みが失敗しました。 ⇒ Help メニューの Error でシステムエラーコードを表示し、OS のマニュアルによりエラー内容をチェックしてください。</p>
-764	<p>・ "Too Many undefined code" ログファイルの容量が 1 MB を超えました。 ⇒ Help メニューの Error でシステムエラーコードを表示し、OS のマニュアルによりエラー内容をチェックしてください。</p>

## 6. Code Converter エラー情報

エラーコード	エラーメッセージ
-765	<p>・”Uoc object file : Kanji exchange object file error” DB Loader 連携機能オブジェクトが開けません。 ⇒ DB Loader 連携機能はサポートしていません。</p>
-766	<p>・”Log file : CODECHK option is not installed” Code Converter-CODECHK オプション製品はインストールされていません。 ⇒ Code Converter-CODECHK オプション製品をインストールしてください。</p>
-767	<p>・”Data exchange : Omission of record data occurs” OTM コード変換処理で、レコードデータの切り捨てが発生しました。 ⇒ ログファイルを参照し入力データおよび入力長を確認して、OTM コード変換をやり直すかチェック下さい。</p>
-780	<p>・”License of Code Converter : It does not have the license.” Code Converter のライセンスキーがありません。 ⇒ Code Converter のライセンスキーを入力してください。</p>
-781	<p>・”License of Code Converter : It expired.” Code Converter のライセンスキー期限が切れています。 ⇒ Code Converter のライセンスキーを入力してください。</p>
-784	<p>・”License of UNICODE option : It does not have the license.” Code Converter-UNICODE のライセンスキーがありません。 ⇒ Code Converter-UNICODE のライセンスキーを入力してください。</p>
-785	<p>・”License of UNICODE option : It expired.” Code Converter-UNICODE のライセンスキー期限が切れています。 ⇒ Code Converter-UNICODE のライセンスキーを入力してください。</p>
-788	<p>・”License of CODECHK option : It does not have the license.” Code Converter-CODECHK のライセンスキーがありません。 ⇒ Code Converter-CODECHK のライセンスキーを入力してください。</p>
-789	<p>・”License of CODECHK option : It expired.” Code Converter-CODECHK のライセンスキー期限が切れています。 ⇒ Code Converter-CODECHK のライセンスキーを入力してください。</p>
-792	<p>・”License of EUC option : It does not have the license.” Code Converter-EUC のライセンスキーがありません。 ⇒ Code Converter-EUC のライセンスキーを入力してください。</p>
-793	<p>・”License of EUC option : It expired.” Code Converter-EUC のライセンスキー期限が切れています。 ⇒ Code Converter-EUC のライセンスキーを入力してください。</p>
-796	<p>・”License of ORALINK option : It does not have the license.” Code Converter-ORALINK のライセンスキーがありません。 ⇒ DB Loader 連携機能はサポートしていません。</p>
-797	<p>・”License of ORALINK option : It expired.” Code Converter-ORALINK のライセンスキー期限が切れています。 ⇒ DB Loader 連携機能はサポートしていません。</p>

## 6. Code Converter エラー情報

### 6.2. トラブル発生時の情報収集

トラブルが発生し、原因が究明できない場合は以下の情報を収集して連絡してください。

[Unix 系の場合]

- ① エラーコード
- ② パラメタ定義ファイル
- ③ ボリューム定義ファイル
- ④ ファイル名定義ファイル
- ⑤ 編集オプションファイル
- ⑥ libuoc.ver  
⇒/usr/lib に格納されています。
- ⑦ libuoc\*.ver ファイル  
⇒Code Converter インストールディレクトリに格納されています。
- ⑧ 可能であれば、データ変換前後のデータ
- ⑨ FAL/FCU のトラブル発生時の情報  
⇒採取方法は FAL/FCU マニュアルを参照してください。

[Windows 系の場合]

- ① エラーコード
- ② パラメタ定義ファイル
- ③ ボリューム定義ファイル
- ④ ファイル名定義ファイル
- ⑤ 編集オプションファイル
- ⑥ libuoc.ver  
⇒C:\WINDOWS に格納されています。
- ⑦ libuoc\*.ver ファイル  
⇒Code Converter インストールディレクトリに格納されています。
- ⑧ 可能であれば、データ変換前後のデータ
- ⑨ FAL/FCU のトラブル発生時の情報  
⇒採取方法は FAL/FCU マニュアルを参照してください。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

この章では、メインフレーム環境からオープン環境への文字コード変換 (m t o) について説明します。

#### 7.1. IBMメインフレームコードからシフトJISコードへの変換

##### 7.1.1. 文字コード変換の概要

IBMメインフレームでは、1バイトコードについては、EBCDIC (英小文字、カナ) コード及びEBCDIK (カナ) コードで表現されています。また、2バイトコードについては、IBM漢字コードで表現されています。

Code Converter では、フィールドの内容に合わせて、表 7.1.1に示す種類の文字コード変換を提供しています。

表 7.1.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EBCDIC コード	EBCDIC から JIS8 コードへ変換	CHAR:OP14
EBCDIK コード	EBCDIK から JIS8 コードへ変換	CHAR:OP15
IBM 漢字コード	IBM 漢字コードからシフトJIS コードへ変換	KANJI:OP11
EBCDIC コードと IBM 漢字コード	EBCDIC と IBM 漢字コードの混在したデータを、JIS8 とシフトJIS コードに変換	KANJI:OP12
EBCDIK コードと IBM 漢字コード	EBCDIK と IBM 漢字コードの混在したデータを、JIS8 とシフトJIS コードに変換	KANJI:OP13

コード変換はすべて、コード変換テーブルを用いて行います。IBM漢字コードは、図 7.1.1に示すようにシフトJISコード (マイクロソフト標準キャラクタセットを標準としています:注1) のコード領域に変換します。

注1:シフトJISコードは、様々なUNIXワークステーション及びPCに搭載されていますが、メーカーにより搭載文字が異なります。基本的には、JIS X0208規格に準拠していますが、装置により対応している規格年度の違いや、メーカーが独自に搭載した文字があり必ずしも統一されていません。このため Code Converter では、データの表示がもっとも頻繁に行われるPC (マイクロソフト標準キャラクタセット:JIS x0208-1990 に準拠) を対象としたシフトJISコードへの変換を標準として提供しています。マイクロソフト標準キャラクタセットの搭載文字については、一般資料「日本語入力ガイド<MS IME for Windows>」を参照ください。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

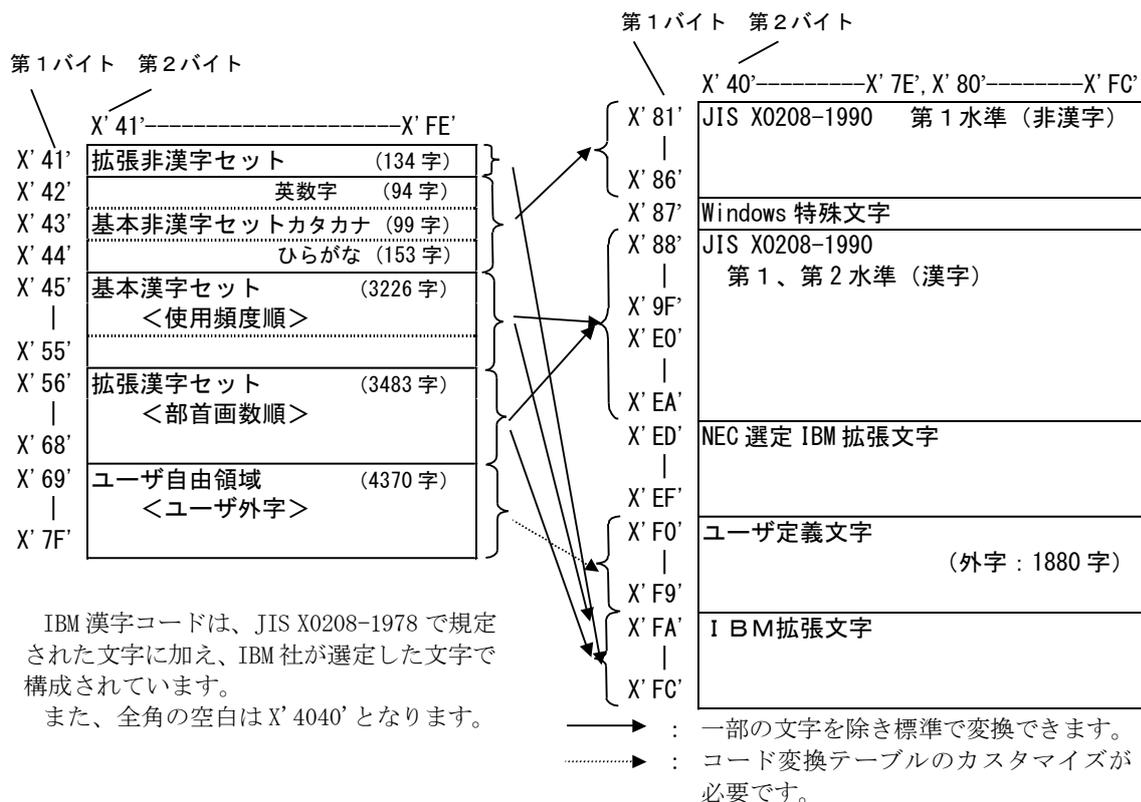


図 7.1.1 IBM漢字コードからシフトJISコードへの変換

### 7.1.2 英数文字フィールド変換 (CHAR:OP14, OP15)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの扱い

シフトコード (X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始) については、未定義コードと同じ扱いとなります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.1.3. 漢字フィールド変換 (KANJI:OP11)

漢字フィールド変換 (OP11) では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FFFF' に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード (コードの値が X'41' ≤ 第1バイト ≤ X'7F' かつ X'41' ≤ 第2バイト ≤ X'FE' の範囲にない文字コード) については、未定義文字コードと同じ扱いとなります。

ただし、空白文字コード (X'40') は除きます。

#### (3) シフトコードの扱い

シフトコード (X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始) についても2バイトコードとして扱いますので、シフトコードとシフトコードに続くデータの2バイトでコード変換を行うため、不当文字コードとして扱います。このため、シフトコード以降のデータは正しいコード変換が行われません。

#### (4) 空白文字コードの扱い

空白文字コード (X'40') の扱いについては、通常は、偶数バイトの X'40' が連続する場合は全角の空白文字として、シフト J I S の全角の空白文字コード (X'8140') に変換し、1バイト X'40' については半角の空白文字として J I S の半角の空白文字コード (X'20') に変換します。

編集オプションファイルに TRAN\_SGLSPACE オプションを指定することにより、X'40' については、全て J I S の半角の空白文字コード (X'20') に変換することもできます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.1.4. 漢字フィールド変換(KANJI:OP12, OP13)

漢字フィールド変換(OP12, OP13)では、フィールドデータ中のシフトコード('X' 0E' : 2バイトコード開始, 'X' 0F' : 1バイトコード開始)により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OP12, OP13)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については、漢字フィールド変換(OP11)に準じますが、以下の点が異なります。

#### (1) シフトコードの扱い

シフト J I S コードではシフトコードがないため、シフトコードは1バイトコード変換、2バイトコード変換の切り替えに使用しますがコード変換の対象としません。

シフトコードがデータ中にある場合のコード変換結果は、データ長が変換前より短くなりますので、左詰めにコード変換の結果を設定し、残りバイト(シフトコードのバイト数分)には'X' FF'を設定します。

なお、編集オプションファイルにフィールド充填コードを指定することにより、残りバイトに設定する値を、任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの仮定

フィールドデータの先頭に、シフトコードが含まれない場合は、1バイトコードとして変換を行います。

先頭が2バイトコードで始まる場合は、編集オプションファイルのコード変換モードにTRAN\_START\_DOUBLEを指定することにより、2バイトコード変換を仮定するよう変更できます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.1.5. コード変換テーブルのカスタマイズ

IBMメインフレームコードからシフトJISコードへ変換を行う場合、表 7.1.2に示す標準提供のコード変換テーブルを使用します。

外字コードを使用する場合や、マイクロソフト標準キャラクタセット以外のシフトJISコードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 7.1.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
IBMSJIS	EBCDIK/EBCDIC コード、IBM 漢字コードから JIS8, シフト JIS コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDICコードからJIS8へのコード変換

標準提供のコード変換テーブルでは、JIS8では規定されていない文字  $\Phi$  (X'4A'),  $\neg$  (X'5F') および  $\pounds$  (X'B1') は未定義文字の定義に、また  $\backslash$  (X'E0') は  $\yen$  (X'5C':JIS8) に代替えておりますので、他の文字に代替える場合、コード変換テーブルのカスタマイズを行ってください。

##### (b) EBCDIKコードからJIS8へのコード変換

標準提供のコード変換テーブルでは、JIS8では規定されていない文字  $\pounds$  (X'4A') は未定義文字の定義になっております。また  $\neg$  (X'5F') については  $\wedge$  (X'5E':JIS8) で代替えていますので、変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (c) IBM漢字コードからシフトJISコードへの変換

標準提供のコード変換テーブルでは、ユーザ定義文字領域 (外字コード) については変換の定義を行っていません。

また、ユーザ定義文字領域 (外字コード) をシフトJISコードに変換する場合は、シフトJISでは、装置により使用できる外字の文字数が異なります。

Code Converter で想定しているPCの場合、外字に使用できる文字数は最大1,880文字となりますので、使用する外字を選定するとともに、使用できない文字の扱いを検討 (例えば、未使用文字を表す外字を1文字決定し、使用できない文字は全て未使用文字に変換する等) する必要があります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.2. 日立メインフレームコードからシフトJISコードへの変換

#### 7.2.1. 文字コード変換の概要

日立メインフレームでは、1バイトコードについては、EBCDIC (英小文字) コード及び EBCDIK (英小文字、カナ) コードで表現されています。また、2バイトコードについては、KEISコードで表現されています。

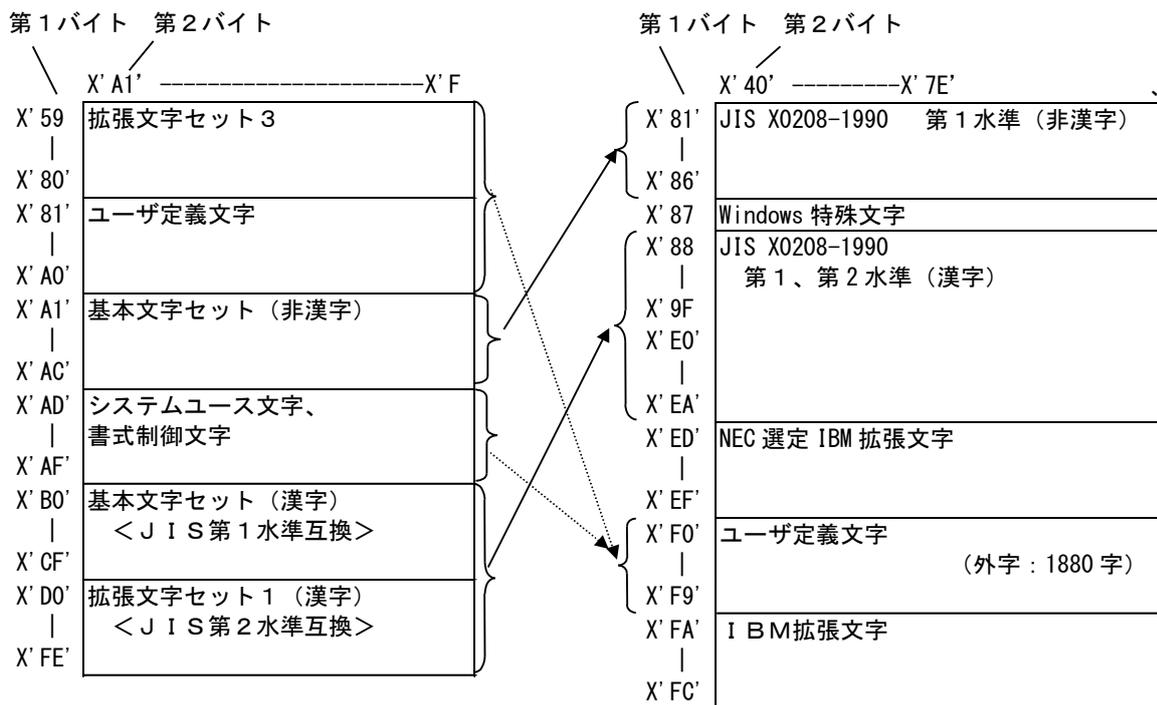
Code Converter では、フィールドの内容に合わせて、表 7.2.1に示す種類の文字コード変換を提供しています。

表 7.2.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EBCDIC コード	EBCDIC から JIS8 コードへ変換	CHAR:OP24, OP34
EBCDIK コード	EBCDIK から JIS8 コードへ変換	CHAR:OP25, OP35
KEIS78 コード	KEIS78 コードからシフトJIS コードへ変換	KANJI:OP21
KEIS83/KEIS90 コード	KEIS83/90 コードからシフトJIS コードへ変換	KANJI:OP31
EBCDIC コードと KEIS78 コード	EBCDIC と KEIS78 コードの混在したデータを、JIS8 とシフトJIS コードに変換	KANJI:OP22
EBCDIC コードと KEIS83/90 コード	EBCDIC と KEIS83/90 コードの混在したデータを、JIS8 とシフトJIS コードに変換	KANJI:OP32
EBCDIK コードと KEIS78 コード	EBCDIK と KEIS78 コードの混在したデータを、JIS8 とシフトJIS コードに変換	KANJI:OP23
EBCDIK コードと KEIS83/90 コード	EBCDIK と KEIS83/90 コードの混在したデータを、JIS8 とシフトJIS コードに変換	KANJI:OP33

コード変換はすべて、コード変換テーブルを用いて行います。KEISコードは、図 7.2.1に示すようにシフトJISコード (マイクロソフト標準キャラクタセットを標準としています) のコード領域に変換します。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)



KEIS コードは、JIS X0208-1978 で規定された文字と日立が選定した拡張文字セット3からなる KEIS78, JIS X0208-1983/90 で規定された文字と日立が選定した拡張文字セット3からなる KEIS83/90 で構成されています。

また、全角スペースは、X' 4040' 及び X' A1A1'

- : 一部の文字を除き標準で変換できます。  
 : コード変換テーブルのカスタマイズが必要です。

図 7.2.1 KEIS コードからシフト JIS コードへの変換

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.2.2. 英数文字フィールド変換 (CHAR:OP24, OP34, OP25, OP35)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード（コード変換テーブルに変換する文字コードが定義されていないコード）については、X'FF'に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの扱い

シフトコード (X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始) については、未定義コードと同じ扱いとなります。

### 7.2.3. 漢字フィールド変換 (KANJI:OP21, OP31)

漢字フィールド変換 (OP21, OP31) では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード（コード変換テーブルに変換する文字コードが定義されていないコード）については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード（コードの値が X'59' ≤ 第1バイト ≤ X'FE' かつ X'A1' ≤ 第2バイト ≤ X'FE' の範囲でない）については、未定義文字コードと同じ扱いとなります。

ただし、空白文字コード (X'40') は除きます。

#### (3) シフトコードの扱い

シフトコード (X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始) についても2バイトコードとして扱いますので、不当文字コードとして扱います。このため、シフトコード以降のデータは正しいコード変換が行われません。

#### (4) 空白文字コードの扱い

偶数バイトの X'40' 及び X'A1A1' は、全角の空白文字として、シフト J I S の全角の空白文字コード (X'8140') に変換し、1バイト X'40' については半角の空白文字として J I S の半角の空白文字コード (X'20') に変換します。

編集オプションファイルに TRAN\_SGLSPACE オプションを指定することにより、全て J I S の半角の空白文字コード (X'20') に変換することもできます。

また、編集オプションファイルに TRAN\_MIXED\_SPACE オプションを指定することにより、X'A1A1' 全角の空白文字はシフト J I S の全角の空白文字コード (X'8140') に変換し、1バイト X'40' については半角の空白文字として J I S の半角の空白文字コード (X'20') に変換します。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.2.4. 漢字フィールド変換 (KANJI:OP22, OP32, OP23, OP33)

漢字フィールド変換 (OP22, OP32, OP23, OP33) では、フィールドデータ中のシフトコード (X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始) により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については、漢字フィールド変換 (OP21, OP31) に準じますが、以下の点が異なります。

#### (1) シフトコードの扱い

シフト J I S コードではシフトコードがないため、シフトコードは1バイトコード変換、2バイトコード変換の切り替えに使用しますがコード変換の対象としません。

シフトコードがデータ中にある場合のコード変換結果は、データ長が短くなりますので、左詰めにコード変換の結果を設定し、残りバイト (シフトコードのバイト数分) には X'FF' を設定します。

なお、編集オプションファイルにフィールド充填コードを指定することにより、残りバイトに設定する値を、任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの仮定

フィールドデータの先頭に、シフトコードが含まれない場合は、1バイトコードとして変換を行います。

先頭が2バイトコードで始まる場合は、編集オプションファイルのコード変換モードに TRAN\_START\_DOUBLE を指定することにより、2バイトコード変換を仮定するよう変更できます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.2.5. コード変換テーブルのカスタマイズ

日立メインフレームコードからシフト J I S コードへ変換を行う場合、表 7.2.2 に示す標準提供のコード変換テーブルを使用します。

外字コード及び拡張文字セット 3 を使用する場合は、マイクロソフト標準キャラクタセット以外のシフト J I S コードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 7.2.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
KS78SJIS	EBCDIK/EBCDIC コード、KE I S 7 8 コードから J I S 8、シフト J I S コードへの変換情報を格納しています。
KS90SJIS	EBCDIK/EBCDIC コード、KE I S 8 3 / 9 0 コードから J I S 8、シフト J I S コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) システムユース文字の扱い

KE I S コードでは、コード領域が  $X'AD' \leq$  第 1 バイト  $\leq X'AF'$  かつ  $X'A1' \leq$  第 2 バイト  $\leq X'FE'$  の中にシステムユース文字 (システムが使用する文字として規定) として丸付き数字等があり、データとして使用している場合があります。

ご使用の場合は、外字コード等に変換するようコード変換テーブルをカスタマイズする必要があります。

##### (b) 拡張文字セット 3、ユーザ定義文字領域の定義

標準提供のコード変換テーブルでは、拡張文字セット 3、ユーザ定義文字領域 (外字コード) については変換の定義を行っていません。

また、シフト J I S コードに変換する場合は、拡張文字セット 3、ユーザ定義文字領域の文字は全て外字コードとして扱うこととなります。シフト J I S では、装置により使用できる外字の文字数が異なります。

Code Converter で想定している P C の場合、外字に使用できる文字数は最大 1, 8 8 0 文字となりますので、使用する外字を選定するとともに、使用できない文字の扱いを検討 (例えば、未使用文字を表す外字を 1 文字決定し、使用できない文字は全て未使用文字に変換する等) する必要があります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.3. 富士通メインフレームコードからシフトJISコードへの変換

#### 7.3.1. 文字コード変換の概要

富士通メインフレームコードでは、1バイトコードについては、EBCDIC (英小文字) コード、EBCDIC (カナ) コード (本書ではEBCDIKと表記しています) 及びEBCDIC (ASCII) コードで表現されています。また、2バイトコードについては、JEFコードで表現されています。

Code Converter では、フィールドの内容に合わせて、表 7.3.1に示す種類の文字コード変換を提供しています。

表 7.3.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EBCDIC コード	EBCDIC から JIS8 コードへ変換	CHAR:OP44
EBCDIK コード	EBCDIK から JIS8 コードへ変換	CHAR:OP45
JEF コード	JEF コードからシフト JIS コードへ変換	KANJI:OP41
EBCDIC コードと JEF コード	EBCDIC と JEF コードの混在したデータを、JIS8 とシフト JIS コードに変換	KANJI:OP42
EBCDIK コードと JEF コード	EBCDIK と JEF コードの混在したデータを、JIS8 とシフト JIS コードに変換	KANJI:OP43

コード変換はすべて、コード変換テーブルを用いて行います。JEFコードは、図 7.3.1に示すようにシフトJISコード (マイクロソフト標準キャラクタセットを標準としています) のコード領域に変換します。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

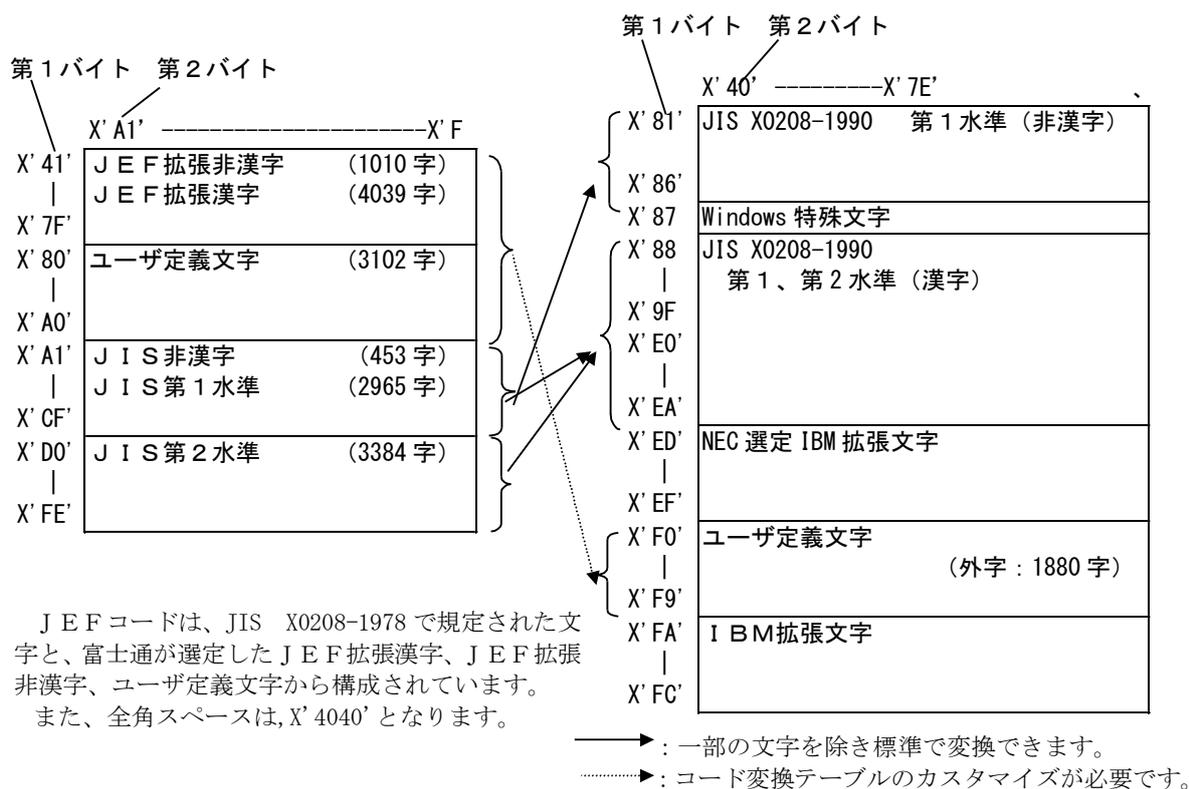


図 7.3.1 J E F コードからシフト J I S コードへの変換

### 7.3.2. 英数文字フィールド変換 (CHAR:OP44, OP45)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X' FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの扱い

シフトコード (X' 28' : 1 2 ポイント 2 バイトコード開始, X' 38' : 9 ポイント 2 バイトコード開始, X' 29' : 1 バイトコード開始) については、未定義コードと同じ扱いとなります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.3.3. 漢字フィールド変換 (KANJI:OP41)

漢字フィールド変換 (OP41) では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行いますが、以下のデータについては扱いが異なります。

(1) 未定義文字コード

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

(2) 不当文字コードの扱い

不当文字コード (コードの値が X'41' ≤ 第1バイト ≤ X'FE' かつ X'A1' ≤ 第2バイト ≤ X'FE' の範囲でない) については、未定義文字コードと同じ扱いとなります。

ただし、空白文字コード (X'40') は除きます。

(3) シフトコードの扱い

シフトコード (X'28':12ポイント2バイトコード開始, X'38':9ポイント2バイトコード開始, X'29':1バイトコード開始) についても2バイトコードとして扱いますので、シフトコードとシフトコードに続くデータの2バイトでコード変換を行うため、不当文字コードとして扱います。このため、シフトコード以降のデータは、正しいコード変換が行われません。

(4) 空白文字コードの扱い

偶数バイトの X'40' 及び X'A1A1' は、全角の空白文字として、シフト J I S の全角の空白文字コード (X'8140') に変換し、1バイト X'40' については半角の空白文字として J I S の半角の空白文字コード (X'20') に変換します。

編集オプションファイルに TRAN\_SGLSPACE オプションを指定することにより、X'40' については、全て J I S の半角の空白文字コード (X'20') に変換することもできます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.3.4. 漢字フィールド変換 (KANJI:OP42, OP43)

漢字フィールド変換 (OP42, OP43) では、フィールドデータ中のシフトコード (X'28' : 1 2ポイント2バイトコード開始, X'38' : 9ポイント2バイトコード開始, X'29' : 1バイトコード開始) により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換 (OP42, OP43) では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については、漢字フィールド変換 (KANJI:OP41) に準じますが、以下の点が異なります。

#### (1) シフトコードの扱い

シフト J I S コードではシフトコードがないため、シフトコードは1バイトコード変換、2バイトコード変換の切り替えに使用しますがコード変換の対象としません。

シフトコードがデータ中にある場合のコード変換結果は、データ長が変換前より短くなりますので、左詰めにコード変換の結果を設定し、残りバイト(シフトコードのバイト数分)にはX'FF'を設定します。

なお、編集オプションファイルにフィールド充填コードを指定することにより、残りバイトに設定する値を、任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの仮定値

フィールドデータの先頭に、シフトコードが含まれない場合は、1バイトコードとして変換を行います。

先頭が2バイトコードで始まる場合は、編集オプションファイルのコード変換モードに TRAN\_START\_DOUBLE を指定することにより、2バイトコード変換を仮定するよう変更できます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.3.5. コード変換テーブルのカスタマイズ

富士通メインフレームコードからシフト J I S コードへ変換を行う場合、通常、表 7.3.2 に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コード、J E F 拡張非漢字/漢字を使用する場合や、マイクロソフト標準キャラクタセット以外のシフト J I S コードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 7.3.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
JEFSJIS	EBCDIK/EBCDIC コード、J E F コードから J I S 8、シフト J I S コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDIC コードから J I S 8 へのコード変換

標準提供のコード変換テーブルでは、J I S 8 では規定されていない文字  $\rightarrow$  (X'5F')、 $\&$  (X'B1') は未定義文字の定義になっておりますので、他の文字に代替える場合、コード変換テーブルのカスタマイズを行ってください。

##### (b) EBCDIK コードから J I S 8 へのコード変換

標準提供のコード変換テーブルでは、J I S 8 では規定されていない文字  $\&$  (X'4A') は未定義文字の定義になっております。また  $\rightarrow$  (X'5F') については  $\wedge$  (X'5E':JIS8) で代替えていますので、変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (c) J E F 拡張漢字、J E F 拡張非漢字、ユーザ定義文字領域の定義

標準提供のコード変換テーブルでは、J E F 拡張漢字、J E F 拡張非漢字、ユーザ定義文字領域 (外字コード) については変換の定義を行っていません。

また、シフト J I S コードに変換する場合は、J E F 拡張漢字、J E F 拡張非漢字、ユーザ定義文字領域の文字は全て外字コードとして扱うこととなります。シフト J I S では、装置により使用できる外字の文字数が異なります。

Code Converter で想定している P C の場合、外字に使用できる文字数は最大 1,880 文字となりますので、使用する外字を選定するとともに、使用できない文字の扱いを検討(例えば、未使用文字を表す外字を 1 文字決定し、使用できない文字は全て未使用文字に変換する等)する必要があります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.4. IBMメインフレームコードからEUCコードへの変換

#### 7.4.1. 文字コード変換の概要

IBMメインフレームでは、1バイトコードについては、EBCDIC (英小文字、カナ) コード及びEBCDIK (カナ) コードで表現されています。また、2バイトコードについては、IBM漢字コードで表現されています。

Code Converter では、フィールドの内容に合わせて、表 7.4.1に示す種類の文字コード変換を提供しています。

表 7.4.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EBCDIC コード	EBCDIC から EUC コードセット 0/2 へ変換	CHAR:OPC4
EBCDIK コード	EBCDIK から EUC コードセット 0/2 へ変換	CHAR:OPC5
IBM 漢字コード	IBM 漢字コードから EUC コードセット 1 へ変換	KANJI:OPC1
EBCDIC コードと IBM 漢字コード	EBCDIC と IBM 漢字コードの混在したデータを、EUC コードセット 0/1/2/3 に変換	KANJI:OPC2
EBCDIK コードと IBM 漢字コード	EBCDIK と IBM 漢字コードの混在したデータを、EUC コードセット 0/1/2/3 に変換	KANJI:OPC3

コード変換はすべて、コード変換テーブルを用いて行います。IBM漢字コードは、図 7.4.1に示すようにEUCコード (JIS X0208-1990:注1) のコード領域に変換します。

注1: EUCコードは、様々なUNIXワークステーションに搭載されていますが、メーカーや装置により対応している規格年度の違いや、メーカーが独自に搭載した文字があり必ずしも統一されていません。このためCode Converter では、JIS X0208-1990規格を対象としたEUCコードへの変換を標準として提供しています。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

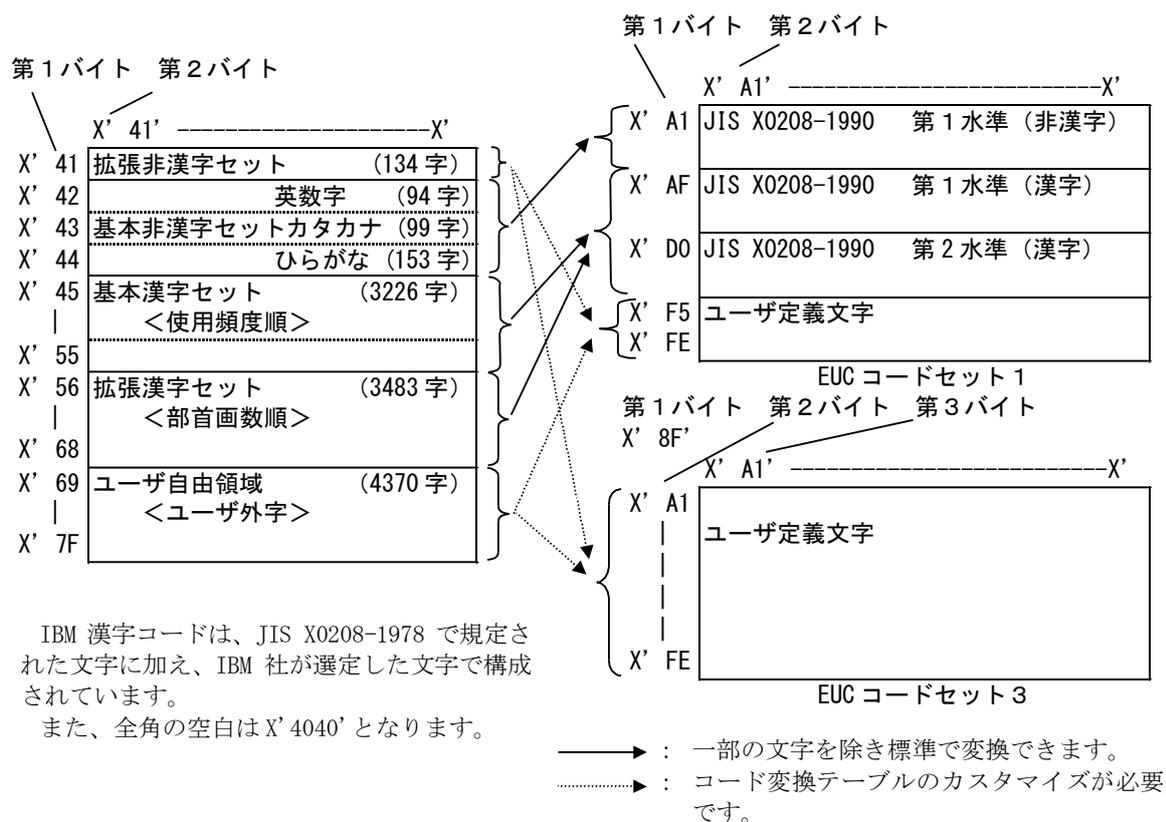


図 7.4.1 IBM漢字コードからEUCコードへの変換

### 7.4.2. 英数文字フィールド変換 (CHAR:OPC4, OPC5)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X' FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの扱い

シフトコード (X' 0E' : 2バイトコード開始, X' 0F' : 1バイトコード開始) については、未定義コードと同じ扱いとなります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.4.3. 漢字フィールド変換(KANJI:OPC1)

漢字フィールド変換(OPC1)では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード(コード変換テーブルに変換する文字コードが定義されていないコード)については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード(コードの値がX'41' ≤ 第1バイト ≤ X'7F' かつ X'41' ≤ 第2バイト ≤ X'FE' の範囲にない文字コード)については、未定義文字コードと同じ扱いとなります。

ただし、空白文字コード(X'40')は除きます。

#### (3) シフトコードの扱い

シフトコード(X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始)についても2バイトコードとして扱いますので、シフトコードとシフトコードに続くデータの2バイトでコード変換を行うため、不当文字コードとして扱います。このため、シフトコード以降のデータは正しいコード変換が行われません。

#### (4) 空白文字コードの扱い

空白文字コード(X'40')の扱いについては、通常は、偶数バイトのX'40'が連続する場合は全角の空白文字として、EUCコードセット1の全角の空白文字コード(X'A1A1')に変換し、1バイトX'40'については半角の空白文字としてEUCコードセット0の半角の空白文字コード(X'20')に変換します。

編集オプションファイルにTRAN\_SGLSPACE オプションを指定することにより、X'40'については、全てEUCコードセット0の半角の空白文字コード(X'20')に変換することもできます。

### 7.4.4. 漢字フィールド変換(KANJI:OPC2, OPC3)

漢字フィールド変換(OPC2, OPC3)では、フィールドデータ中のシフトコード(X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始)により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OPC2, OPC3)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については、漢字フィールド変換(OPC1)に準じますが、以下の点が異なります。

#### (1) シフトコードの扱い

シフトコードは1バイトコード変換、2バイトコード変換の切り替えに使用しますが、EUCコードではシフトコードがないため、コード変換の対象としません。

シフトコードがデータ中にある場合のコード変換結果は、データ長が変換前と一致しない場合があります。そのためコード変換の結果がフィールド長より短くなる場合は、左詰めにコード変換の結果を設定し、更にフィールドの残りバイトにはX'FF'を設定します。また、コード変換の結果がフィールド長を超える場合は、変換可能な範囲までを変換しあふれる部分については変換しません。

なお、編集オプションファイルにフィールド充填コードを指定することにより、残りバイトに設定する値を、任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの仮定

フィールドデータの先頭に、シフトコードが含まれない場合は、1バイトコードとして変換を行います。

先頭が2バイトコードで始まる場合は、編集オプションファイルのコード変換モードにTRAN\_START\_DOUBLEを指定することにより、2バイトコード変換を仮定するよう変更できます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.4.5. コード変換テーブルのカスタマイズ

IBMメインフレームコードからEUCコードへ変換を行う場合、表 7.4.2に示す標準提供のコード変換テーブルを使用します。

外字コードを使用する場合や、JIS X0208-1990規格の文字セット以外のEUCコードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 7.4.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
IBMEUC	EBCDIK/EBCDIC コード、IBM 漢字コードから JIS X0208-1990 規格の文字セットに対応した EUC コードセット 0/1/2/3 への変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDICコードからEUCコードセット0/2へのコード変換

標準提供のコード変換テーブルでは、EUCコードセット0では規定されていない文字  $\text{¢}$  (X'4A'),  $\text{←}$  (X'5F') および  $\text{£}$  (X'B1') は未定義文字の定義に、また  $\text{↵}$  (X'E0') は  $\text{¥}$  (X'5C':EUC コードセット 0) に代替えておりますので、他の文字に代替える場合、コード変換テーブルのカスタマイズを行ってください。

##### (b) EBCDIKコードからEUCコードセット0/2へのコード変換

標準提供のコード変換テーブルでは、EUCコードセット0では規定されていない文字  $\text{£}$  (X'4A') は未定義文字の定義になっております。また  $\text{←}$  (X'5F') については  $\text{^}$  (X'5E':EUC コードセット 0) で代替えておりますので、変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (c) IBM漢字コードからEUCコードセット1/3へのコード変換

標準提供のコード変換テーブルでは、ユーザ定義文字領域(外字コード)と、一部のJIS X0208-1990規格に存在しない文字については変換の定義を行っていません。

また、ユーザ定義文字領域(外字コード)をEUCコードに変換する場合は、装置により使用できる外字の文字数が異なります。

Code Converter で想定しているEUCコードの場合、外字に使用できる文字数は最大9,776(8,836:コードセット3+940:コードセット1)文字となりますが、使用するEUCコードによっては、この文字数以下となる場合があります。使用できる文字数が足りない場合は、使用する外字を選定するとともに、使用できない文字の扱いを検討(例えば、未使用文字を表す外字を1文字決定し、使用できない文字は全て未使用文字に変換する等)をする必要があります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.5. IBMメインフレームコードからUnicodeへの変換

#### 7.5.1. 文字コード変換の概要

IBMメインフレームでは、1バイトコードについては、EBCDIC (英小文字、カナ) コード及びEBCDIK (カナ) コードで表現されています。また、2バイトコードについては、IBM漢字コードで表現されています。

Code Converter では、フィールドの内容に合わせて、表 7.4.1 に示す種類の文字コード変換を提供しています。

表 7.5.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EBCDIC コード	EBCDIC から UNICODE へ変換	CHAR:OP64
EBCDIK コード	EBCDIK から UNICODE へ変換	CHAR:OP65
IBM 漢字コード	IBM 漢字コードから UNICODE へ変換	KANJI:OP61
EBCDIC コードと IBM 漢字コード	EBCDIC と IBM 漢字コードの混在したデータを、UNICODE に変換	KANJI:OP62
EBCDIK コードと IBM 漢字コード	EBCDIK と IBM 漢字コードの混在したデータを、UNICODE に変換	KANJI:OP63

注1: UNICODEは、様々なUNIXワークステーション及びPCに搭載されていますが、メーカーや装置により対応している規格年度の違いや、メーカーが独自に搭載した文字があり必ずしも統一されていません。このため Code Converter では、JIS X0208-1990規格を対象としたUNICODEへの変換を標準として提供しています。

#### 7.5.2. 英数文字フィールド変換(CHAR:OP64, OP65)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

##### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'00FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、1バイト置換文字コードの前に0x00を付加したコードに置き換えることもできます。

但し、半角カナ文字は、UNICODEでは、2バイトコードになるため指定出来ません。

##### (2) シフトコードの扱い

シフトコード (X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始) については、未定義コードと同じ扱いとなります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.5.3. 漢字フィールド変換(KANJI:OP61)

漢字フィールド変換(OP61)では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

(1) 未定義文字コード

未定義文字コード(コード変換テーブルに変換する文字コードが定義されていないコード)については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

(2) 不当文字コードの扱い

不当文字コード(コードの値がX'41' ≤ 第1バイト ≤ X'7F' かつ X'41' ≤ 第2バイト ≤ X'FE' の範囲にない文字コード)については、未定義文字コードと同じ扱いとなります。

ただし、空白文字コード(X'40')は除きます。

(3) シフトコードの扱い

シフトコード(X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始)についても2バイトコードとして扱いますので、シフトコードとシフトコードに続くデータの2バイトでコード変換を行うため、不当文字コードとして扱います。このため、シフトコード以降のデータは正しいコード変換が行われません。

(4) 空白文字コードの扱い

空白文字コード(X'40')の扱いについては、通常は、偶数バイトのX'40'が連続する場合は全角の空白文字として、UNICODEの全角の空白文字コード(X'3000')に変換し、1バイトX'40'については半角の空白文字としての半角の空白文字コード(X'0020')に変換します。

編集オプションファイルにTRAN\_SGLSPACE オプションを指定することにより、X'40'については、全てUNICODEの半角の空白文字コード(X'0020')に変換することもできます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.5.4. 漢字フィールド変換(KANJI:OP62, OP63)

漢字フィールド変換(OP62, OP63)では、フィールドデータ中のシフトコード (X' 0E' : 2バイトコード開始, X' 0F' : 1バイトコード開始) により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OP62, OP63)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については、漢字フィールド変換(OP61) に準じますが、以下の点が異なります。

#### (1) シフトコードの扱い

UNICODEではシフトコードがないため、シフトコードは1バイトコード変換、2バイトコード変換の切り替えに使用しますがコード変換の対象としません。

シフトコードがデータ中にある場合のコード変換結果は、データ長が変換前より短くなりますので、左詰めにコード変換の結果を設定し、残りバイト(シフトコードのバイト数分)にはX' 00FF' を設定します。

なお、編集オプションファイルにフィールド充填コード半角を指定することにより、残りバイトに設定する値を、任意の1バイト文字(フィールド充填コード半角の前に 0x00 を付加したコード)に置き換えることもできます。

但し、半角カナ文字は、UNICODEでは、2バイトコードになるため指定出来ません。

#### (2) シフトコードの仮定

フィールドデータの先頭に、シフトコードコードが含まれない場合は、1バイトコードとして変換を行います。

先頭が2バイトコードで始まる場合は、編集オプションファイルのコード変換モードにTRAN\_START\_DOUBLEを指定することにより、2バイトコード変換を仮定するよう変更できます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.5.5. コード変換テーブルのカスタマイズ

IBMメインフレームコードからUNICODEへ変換を行う場合、表 7.4.2 に示す標準提供のコード変換テーブルを使用します。

外字コードを使用する場合や、JIS X 0208-1990規格の文字セット以外のUNICODEにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 7.5.2 Code Converter の標準提供コード変換

ファイル名	テーブルの内容
IBMUNI	EBCDIK/EBCDIC コード, IBM 漢字コードから UNICODE への変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDICコードからUNICODEへのコード変換

標準提供のコード変換テーブルでは、UNICODEでは規定されていない文字  $\Phi$  (X'4A'),  $\rightarrow$  (X'5F') および  $\&$  (X'B1') は未定義文字の定義に、また  $\backslash$  (X'E0') は  $\forall$  (X'005C':UNICODE) に代替えておりますので、他の文字に代替える場合、コード変換テーブルのカスタマイズを行ってください。

##### (b) EBCDIKコードからUNICODEへのコード変換

標準提供のコード変換テーブルでは、UNICODEでは規定されていない文字  $\&$  (X'B1') は未定義文字の定義になっております。また  $\rightarrow$  (X'5F') については  $\wedge$  (X'005E':UNICODE) で代替えていますので、変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (c) IBM漢字コードからUNICODEへの変換

標準提供のコード変換テーブルでは、ユーザ定義文字領域 (外字コード) については変換の定義を行っていません。

また、ユーザ定義文字領域 (外字コード) をUNICODEに変換する場合は、UNICODEでは、装置により使用できる外字の文字数が異なります。

Code Converter で想定しているPCの場合、外字に使用できる文字数は最大6,400文字となりますので、使用する外字を選定するとともに、使用できない文字の扱いを検討 (例えば、未使用文字を表す外字を1文字決定し、使用できない文字は全て未使用文字に変換する等) する必要があります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.6. 日立メインフレームコードからUNICODEへの変換

#### 7.6.1. 文字コード変換の概要

日立メインフレームでは、1バイトコードについては、EBCDIC (英小文字) コード及び EBCDIK (英小文字、カナ) コードで表現されています。また、2バイトコードについては、KEISコードで表現されています。

Code Converter では、フィールドの内容に合わせて、表 7.5.1 に示す種類の文字コード変換を提供しています。

表 7.6.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EBCDIC コード	EBCDIC から UNICODE へ変換	CHAR:OP74, OP84
EBCDIK コード	EBCDIK から UNICODE へ変換	CHAR:OP75, OP85
KEIS78 コード	KEIS78 コードから UNICODE へ変換	KANJI:OP71
KEIS83/KEIS90 コード	KEIS83/90 コードから UNICODE へ変換	KANJI:OP81
EBCDIC コードと KEIS78 コード	EBCDIC と KEIS78 コードの混在したデータを、UNICODE に変換	KANJI:OP72
EBCDIC コードと KEIS83/90 コード	EBCDIC と KEIS83/90 コードの混在したデータを、UNICODE に変換	KANJI:OP82
EBCDIK コードと KEIS78 コード	EBCDIK と KEIS78 コードの混在したデータを、UNICODE に変換	KANJI:OP73
EBCDIK コードと KEIS83/90 コード	EBCDIK と KEIS83/90 コードの混在したデータを、UNICODE に変換	KANJI:OP83

コード変換はすべて、コード変換テーブルを用いて行い、UNICODEは、JIS X 0208-1990規格を想定しています。

#### 7.6.2. 英数文字フィールド変換 (CHAR:OP74, OP84, OP75, OP85)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。

コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

##### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'00FF'に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、1バイト置換文字コードの前に0x00を付加したコードに置き換えることもできます。

但し、半角カナ文字は、UNICODEでは、2バイトコードになるため指定出来ません。

##### (2) シフトコードの扱い

シフトコード (X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始) については、未定義コードと同じ扱いとなります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.6.3. 漢字フィールド変換 (KANJI:OP71, OP81)

漢字フィールド変換 (KANJI:OP71, OP81) では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード (コードの値が X'59' ≤ 第1バイト ≤ X'FE' かつ X'A1' ≤ 第2バイト ≤ X'FE' の範囲でない) については、未定義文字コードと同じ扱いとなります。

ただし、空白文字コード (X'40') は除きます。

#### (3) シフトコードの扱い

シフトコード (X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始) についても2バイトコードとして扱いますので、不当文字コードとして扱います。このため、シフトコード以降のデータは正しいコード変換が行われません。

#### (4) 空白文字コードの扱い

偶数バイトの X'40' 及び X'A1A1' は、全角の空白文字として、UNICODEの全角の空白文字コード (X'3000') に変換し、1バイト X'40' については半角の空白文字としてUNICODEの半角の空白文字コード (X'0020') に変換します。

編集オプションファイルに TRAN\_SGLSPACE オプションを指定することにより、全てUNICODEの半角の空白文字コード (X'0020') に変換することもできます。

### 7.6.4. 漢字フィールド変換 (KANJI:OP72, OP82, OP73, OP83)

漢字フィールド変換 (OP72, OP82, OP73, OP83) では、フィールドデータ中のシフトコード (X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始) により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については、漢字フィールド変換 (OP72, OP73) に準じますが、以下の点が異なります。

#### (1) シフトコードの扱い

UNICODEではシフトコードがないため、シフトコードは1バイトコード変換、2バイトコード変換の切り替えに使用しますがコード変換の対象としません。

シフトコードがデータ中にある場合のコード変換結果は、データ長が短くなりますので、左詰めにコード変換の結果を設定し、残りバイト (シフトコードのバイト数分) には X'00FF' を設定します。

なお、編集オプションファイルにフィールド充填コード半角を指定することにより、残りバイトに設定する値を、任意の1バイト文字 (フィールド充填コード半角の前に 0x00 を付加したコード) に置き換えることもできます。

但し、半角カナ文字は、UNICODEでは、2バイトコードになるため指定出来ません。

#### (2) シフトコードの仮定

フィールドデータの先頭に、シフトコードコードが含まれない場合は、1バイトコードとして変換を行います。

先頭が2バイトコードで始まる場合は、編集オプションファイルのコード変換モードに TRAN\_START\_DOUBLE を指定することにより、2バイトコード変換を仮定するよう変更できます。

## 7. メインフレーム環境からオープン環境への文字コード変換 ( m t o )

### 7.6.5. コード変換テーブルのカスタマイズ

日立メインフレームコードからUNICODEへ変換を行う場合、表 7.5.2 に示す標準提供のコード変換テーブルを使用します。

外字コード及び拡張文字セット3を使用する場合や、JIS X 0208-1990規格の文字セット以外のUNICODEにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 7.6.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
KS78UNI	EBCDIK/EBCDIC コード、KEIS78コードからUNICODEへの変換情報を格納しています。
KS90UNI	EBCDIK/EBCDIC コード、KEIS83/90コードからUNICODEへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) システムユース文字の扱い

KEISコードでは、コード領域が  $X'AD' \leq \text{第1バイト} \leq X'AF'$  かつ  $X'A1' \leq \text{第2バイト} \leq X'FE'$  の中にシステムユース文字 (システムが使用する文字として規定) として丸付き数字等があり、データとして使用している場合があります。

ご使用の場合は、外字コード等に変換するようコード変換テーブルをカスタマイズする必要があります。

##### (b) 拡張文字セット3、ユーザ定義文字領域の定義

標準提供のコード変換テーブルでは、拡張文字セット3、ユーザ定義文字領域 (外字コード) については変換の定義を行っていません。

また、UNICODEに変換する場合は、拡張文字セット3、ユーザ定義文字領域の文字は全て外字コードとして扱うこととなります。UNICODEでは、装置により使用できる外字の文字数が異なります。

Code Converter で想定しているPCの場合、外字に使用できる文字数は最大6,400文字となりますので、使用する外字を選定するとともに、使用できない文字の扱いを検討(例えば、未使用文字を表す外字を1文字決定し、使用できない文字は全て未使用文字に変換する等)する必要があります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.7. 富士通メインフレームコードからUNICODEへの変換

#### 7.7.1. 文字コード変換の概要

富士通メインフレームコードでは、1バイトコードについては、EBCDIC (英小文字) コード、EBCDIC (カナ) コード (本書ではEBCDIKと表記しています) 及びEBCDIC (ASCII) コードで表現されています。また、2バイトコードについては、JEFコードで表現されています。

Code Converter では、フィールドの内容に合わせて、表 7.6.1 に示す種類の文字コード変換を提供しています。

表 7.7.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EBCDIC コード	EBCDIC から UNICODE へ変換	CHAR:OP94
EBCDIK コード	EBCDIK から UNICODE へ変換	CHAR:OP95
JEF コード	JEF コードから UNICODE へ変換	KANJI:OP91
EBCDIC コードと JEF コード	EBCDIC と JEF コードの混在したデータを、UNICODE に変換	KANJI:OP92
EBCDIK コードと JEF コード	EBCDIK と JEF コードの混在したデータを、UNICODE に変換	KANJI:OP93

コード変換はすべて、コード変換テーブルを用いて行い、UNICODEは、JIS X 0208-1990規格を想定しています。

#### 7.7.2. 英数文字フィールド変換 (CHAR:OP94, OP95)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

##### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'00FF'に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、1バイト置換文字コードの前に0x00を付加したコードに置き換えることもできます。

但し、半角カナ文字は、UNICODEでは、2バイトコードになるため指定出来ません。

##### (2) シフトコードの扱い

シフトコード (X'28' : 12ポイント2バイトコード開始, X'38' : 9ポイント2バイトコード開始, X'29' : 1バイトコード開始) については、未定義コードと同じ扱いとなります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.7.3. 漢字フィールド変換 (KANJI:OP91)

漢字フィールド変換 (OP91) では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード (コードの値が X'41' ≤ 第1バイト ≤ X'FE' かつ X'A1' ≤ 第2バイト ≤ X'FE' の範囲でない) については、未定義文字コードと同じ扱いとなります。

ただし、空白文字コード (X'40') は除きます。

#### (3) シフトコードの扱い

シフトコード (X'28':12ポイント2バイトコード開始, X'38':9ポイント2バイトコード開始, X'29':1バイトコード開始) についても2バイトコードとして扱いますので、シフトコードとシフトコードに続くデータの2バイトでコード変換を行うため、不当文字コードとして扱います。このため、シフトコード以降のデータは、正しいコード変換が行われません。

#### (4) 空白文字コードの扱い

偶数バイトの X'40' 及び X'A1A1' は、全角の空白文字として、UNICODEの全角の空白文字コード (X'3000') に変換し、1バイト X'40' については半角の空白文字としてUNICODEの半角の空白文字コード (X'0020') に変換します。

編集オプションファイルに TRAN\_SGLSPACE オプションを指定することにより、X'40' については、全てUNICODEの半角の空白文字コード (X'0020') に変換することもできます。

### 7.7.4. 漢字フィールド変換 (KANJI:OP92, OP93)

漢字フィールド変換 (OP92, OP93) では、フィールドデータ中のシフトコード (X'28':12ポイント2バイトコード開始, X'38':9ポイント2バイトコード開始, X'29':1バイトコード開始) により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換 (OP92, OP93) では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については、漢字フィールド変換 (OP91) に準じますが、以下の点が異なります。

#### (1) シフトコードの扱い

UNICODEではシフトコードがないため、シフトコードは1バイトコード変換、2バイトコード変換の切り替えに使用しますがコード変換の対象としません。

シフトコードがデータ中にある場合のコード変換結果は、データ長が変換前より短くなりますので、左詰めにコード変換の結果を設定し、残りバイト(シフトコードのバイト数分)にはX'FF'を設定します。

なお、編集オプションファイルにフィールド充填コード半角を指定することにより、残りバイトに設定する値を、任意の1バイト文字 (フィールド充填コード半角の前に 0x00 を付加したコード) に置き換えることもできます。

但し、半角カナ文字は、UNICODEでは、2バイトコードになるため指定出来ません。

#### (2) シフトコードの仮定値

フィールドデータの先頭に、シフトコードコードが含まれない場合は、1バイトコードとして変換を行います。

先頭が2バイトコードで始まる場合は、編集オプションファイルのコード変換モードに TRAN\_START\_DOUBLE を指定することにより、2バイトコード変換を仮定するよう変更できます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.7.5. コード変換テーブルのカスタマイズ

富士通メインフレームコードからUNICODEへ変換を行う場合、通常、表 7.6.5 に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コード、J E F 拡張非漢字／漢字を使用する場合や、J I S X 0 2 0 8 - 1 9 9 0 規格の文字セット以外のUNICODEにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 7.7.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
JEFUNI	EBCDIK/EBCDIC コード、J E F コードから UNICODE への変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDICコードからUNICODEへのコード変換

標準提供のコード変換テーブルでは、UNICODEで規定されていない文字£ (X'4A'), 縦線 (X'6A') は、未定義文字の定義になっております。またー (X'5F') は ^ (X'005E':Unicode) で、     (X'A1') は ~ (X'007E':Unicode) で代替えていますので、変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) EBCDIKコードからUNICODEへのコード変換

標準提供のコード変換テーブルでは、UNICODEで規定されていない文字£ (X'4A'), 縦線 (X'6A') は未定義文字の定義になっております。またー (X'5F') は ^ (X'005E':Unicode) で、     (X'A1') は ~ (X'007E':Unicode) で代替えていますので、変更する場合は、コード変換テーブルのカスタマイズを行ってください。

また、英小文字については、本来 EBCDIK に含まれませんが、Unicode とのコード変換を行う際、情報の欠落を防ぐため、未使用の部分に割り当てています。

##### (c) J E F 拡張漢字、J E F 拡張非漢字、ユーザ定義文字領域の定義

標準提供のコード変換テーブルでは、J E F 拡張漢字、J E F 拡張非漢字、ユーザ定義文字領域 (外字コード) については変換の定義を行っていません。

また、UNICODEに変換する場合は、J E F 拡張漢字、J E F 拡張非漢字、ユーザ定義文字領域の文字は全て外字コードとして扱うこととなります。UNICODEでは、装置により使用できる外字の文字数が異なります。

Code Converter で想定しているPCの場合、外字に使用できる文字数は最大6,400文字となりますので、使用する外字を選定するとともに、使用できない文字の扱いを検討(例えば、未使用文字を表す外字を1文字決定し、使用できない文字は全て未使用文字に変換する等)する必要があります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.8. 日立メインフレームからEUCコードへの変換

#### 7.8.1. 文字コード変換の概要

日立メインフレームでは、1バイトコードについては、EBCDIC (英小文字、カナ) コード及びEBCDIK (カナ) コードで表現されています。また、2バイトコードについては、日立漢字コードで表現されています。

Code Converter では、フィールドの内容に合わせて、表 7.8.1に示す種類の文字コード変換を提供しています。

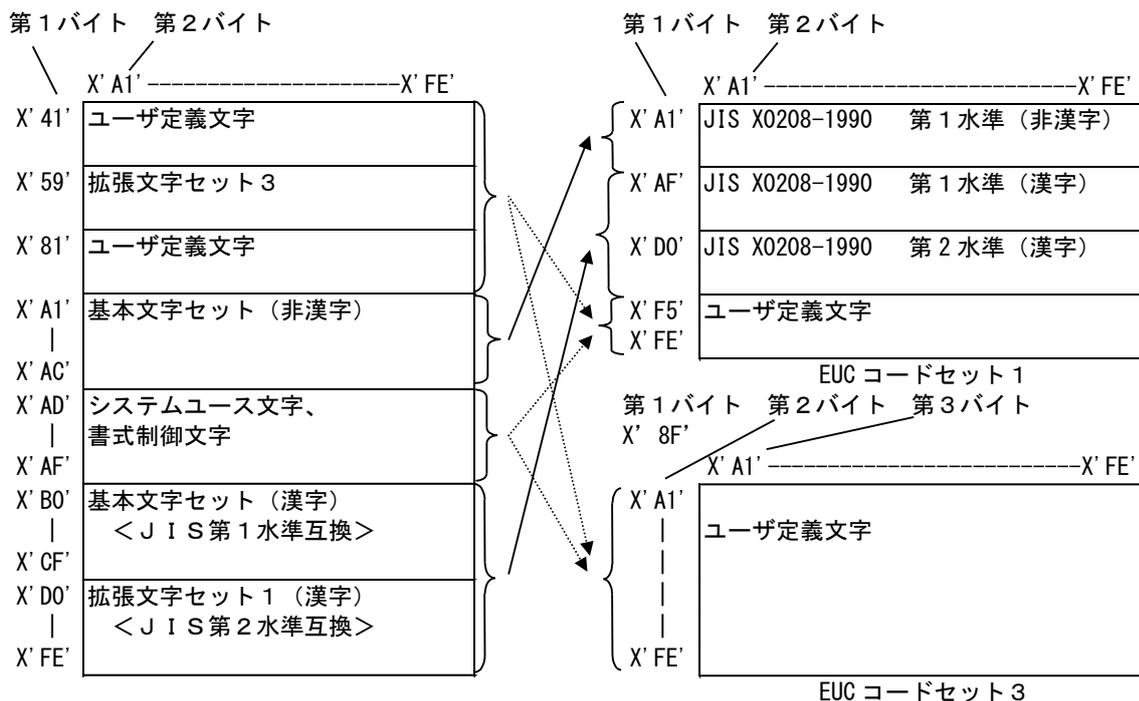
表 7.8.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EBCDIC コード	EBCDIC から EUC コードセット 0/2 へ変換	CHAR:OPD4, OPE4
EBCDIK コード	EBCDIK から EUC コードセット 0/2 へ変換	CHAR:OPD5, OPE5
KEIS 漢字コード	KEIS 漢字コードから EUC コードセット 1 へ変換	KANJI:OPD1, OPE1
EBCDIC コードと KEIS 漢字コード	EBCDIC と KEIS 漢字コードの混在したデータを、EUC コードセット 0/1/2/3 に変換	KANJI:OPD2, OPE2
EBCDIK コードと KEIS 漢字コード	EBCDIK と KEIS 漢字コードの混在したデータを、EUC コードセット 0/1/2/3 に変換	KANJI:OPD3, OPE3

コード変換はすべて、コード変換テーブルを用いて行います。KEIS漢字コードは、図 7.8.1に示すようにEUCコード (JIS X0208-1990規格:注1) のコード領域に変換します。

注1: EUCコードは、様々なUNIXワークステーションに搭載されていますが、メーカーや装置により対応している規格年度の違いや、メーカーが独自に搭載した文字があり必ずしも統一されていません。このためCode Converter では、JIS X0208-1990規格を対象としたEUCコードへの変換を標準として提供しています。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)



KEIS 漢字コードは、JIS X0208-1978 で規定された文字と日立が選定した拡張文字セット3からなる KEIS78, JIS X0208-1983/90 で規定された文字と日立が選定した拡張文字セット3からなる KEIS83/90 から構成されています。

また、全角スペースは、X'4040' 及び X'A1A1'   
 —————> : 一部の文字を除き標準で変換できます。  
 .....> : コード変換テーブルのカスタマイズが必要です。

図 7.8.1 KEIS漢字コードからEUCコードへの変換

### 7.8.2 英数文字フィールド変換 (CHAR:OPD4, OPE4, OPD5, OPE5)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、コード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの扱い

シフトコード (X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始) については、未定義コードと同じ扱いとなります。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.8.3. 漢字フィールド変換(KANJI:OPD1, OPE1)

漢字フィールド変換(OPD1, OPE1)では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード(コード変換テーブルに変換する文字コードが定義されていないコード)については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード(コードの値がX'41' ≤ 第1バイト ≤ X'FE'かつX'A1' ≤ 第2バイト ≤ X'FE'の範囲にない文字コード)については、未定義文字コードと同じ扱いとなります。

ただし、空白文字コード(X'40')は除きます。

#### (3) シフトコードの扱い

シフトコード(X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始)についても2バイトコードとして扱いますので、シフトコードとシフトコードに続くデータの2バイトでコード変換を行うため、不当文字コードとして扱います。このため、シフトコード以降のデータは正しいコード変換が行われません。

#### (4) 空白文字コードの扱い

空白文字コード(X'40')の扱いについては、通常は、偶数バイトのX'40'が連続する場合は全角の空白文字として、EUCコードセット1の全角の空白文字コード(X'A1A1')に変換し、1バイトX'40'については半角の空白文字としてEUCコードセット0の半角の空白文字コード(X'20')に変換します。

編集オプションファイルにTRAN\_SGLSPACE オプションを指定することにより、X'40'については、全てEUCコードセット0の半角の空白文字コード(X'20')に変換することもできます。

### 7.8.4. 漢字フィールド変換(KANJI:OPD2, OPE2, OPD3, OPE3)

漢字フィールド変換(OPD2, OPE2, OPD3, OPE3)では、フィールドデータ中のシフトコード(X'0A42' : 2バイトコード開始, X'0A41' : 1バイトコード開始)により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OPD2, OPE2, OPD3, OPE3)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については、漢字フィールド変換(OPD1, OPE1)に準じますが、以下の点が異なります。

#### (1) シフトコードの扱い

EUCコードでは1バイトコードと2バイトコードのモードを切り替えるシフトコードがないため、シフトコードは1バイトコード変換、2バイトコード変換の切り替えに使用しますがコード変換の対象としません。

シフトコードがデータ中にある場合のコード変換結果は、データ長が変換前と一致しない場合があります。そのためコード変換の結果がフィールド長より短くなる場合は、左詰めにコード変換の結果を設定し、更にフィールドの残りバイトにはX'FF'を設定します。また、コード変換の結果がフィールド長を超える場合は、変換可能な範囲までを変換しあふれる部分については変換しません。

なお、編集オプションファイルにフィールド充填コードを指定することにより、残りバイトに設定する値を、任意の1バイト文字に置き換えることもできます。

#### (2) シフトコードの仮定

フィールドデータの先頭に、シフトコードが含まれない場合は、1バイトコードとして変換を行います。

先頭が2バイトコードで始まる場合は、編集オプションファイルのコード変換モードにTRAN\_START\_DOUBLEを指定することにより、2バイトコード変換を仮定するよう変更できます。

## 7. メインフレーム環境からオープン環境への文字コード変換 (m t o)

### 7.8.5. コード変換テーブルのカスタマイズ

日立メインフレームコードからEUCコードへ変換を行う場合、表 7.8.2に示す標準提供のコード変換テーブルを使用します。

外字コードを使用する場合や、JIS X0208-1990規格の文字セット以外のEUCコードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 7.8.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
KS78EUC.tbl	EBCDIK/EBCDIC コード、KEIS 漢字コードから JIS X0208-1990 規格の文字セットに対応した EUC コードセット 0/1/2/3 への変換情報を格納しています。
KS90EUC.tbl	

#### (1) カスタマイズのポイント

##### (a) EBCDICコードからEUCコードセット0/2へのコード変換

標準提供のコード変換テーブルでは、EUCコードセット0では規定されていない文字¢ (X'4A'), ◡ (X'5F') および£ (X'B1') は未定義文字の定義に、また\ (X'E0') は¥ (X'5C':EUCコードセット0) に代替しておりますので、他の文字に代替える場合、コード変換テーブルのカスタマイズを行ってください。

##### (b) EBCDIKコードからEUCコードセット0/2へのコード変換

標準提供のコード変換テーブルでは、EUCコードセット0では規定されていない文字£ (X'4A') は未定義文字の定義になっております。また◡ (X'5F') については^ (X'5E':EUCコードセット0) で代替しておりますので、変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (c) KEIS漢字コードからEUCコードセット1/3へのコード変換

標準提供のコード変換テーブルでは、ユーザ定義文字領域(外字コード)と、一部のJIS X0208-1990規格に存在しない文字については変換の定義を行っていません。

また、ユーザ定義文字領域(外字コード)をEUCコードに変換する場合は、装置により使用できる外字の文字数が異なります。

Code Converter で想定しているEUCコードの場合、外字に使用できる文字数は最大9,776(8,836:コードセット3+940:コードセット1)文字となりますが、使用するEUCコードによっては、この文字数以下となる場合があります。使用できる文字数が足りない場合は、使用する外字を選定するとともに、使用できない文字の扱いを検討(例えば、未使用文字を表す外字を1文字決定し、使用できない文字は全て未使用文字に変換する等)をする必要があります。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

この章では、オープン環境からメインフレーム環境への文字コード変換 (o t m) について説明します。

#### 8.1. シフト J I S コードから I B M メインフレームコードへの変換

##### 8.1.1. 文字コード変換の概要

シフト J I S コードから I B M メインフレームコードへのコード変換では、1 バイトコードについては、J I S 8 コードを E B C D I C (英小文字、カナ) コード及び E B C D I K (カナ) コードへ、また、2 バイトコードについては、シフト J I S コードを I B M 漢字コードへ変換します。

Code Converter では、フィールドの内容に合わせて、表 8.1.1 に示す種類の文字コード変換を提供しています。

表 8.1.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
JIS8 コード	JIS8 から EBCDIC コードへ変換	CHAR:OP19
	JIS8 から EBCDIK コードへ変換	CHAR:OP1A
シフト JIS コード	シフト JIS コードから IBM 漢字コードへ変換	KANJI:OP16
JIS8 とシフト JIS コード	JIS8 とシフト JIS コードの混在したデータを、EBCDIC と IBM 漢字コードに変換	KANJI:OP17
	JIS8 とシフト JIS コードの混在したデータを、EBCDIK と IBM 漢字コードに変換	KANJI:OP18

コード変換はすべて、コード変換テーブルを用いて行います。シフト J I S コード(マイクロソフト標準キャラクタセットを標準としています)は、図 8.1.1 に示すように I B M 漢字コード領域に変換します。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

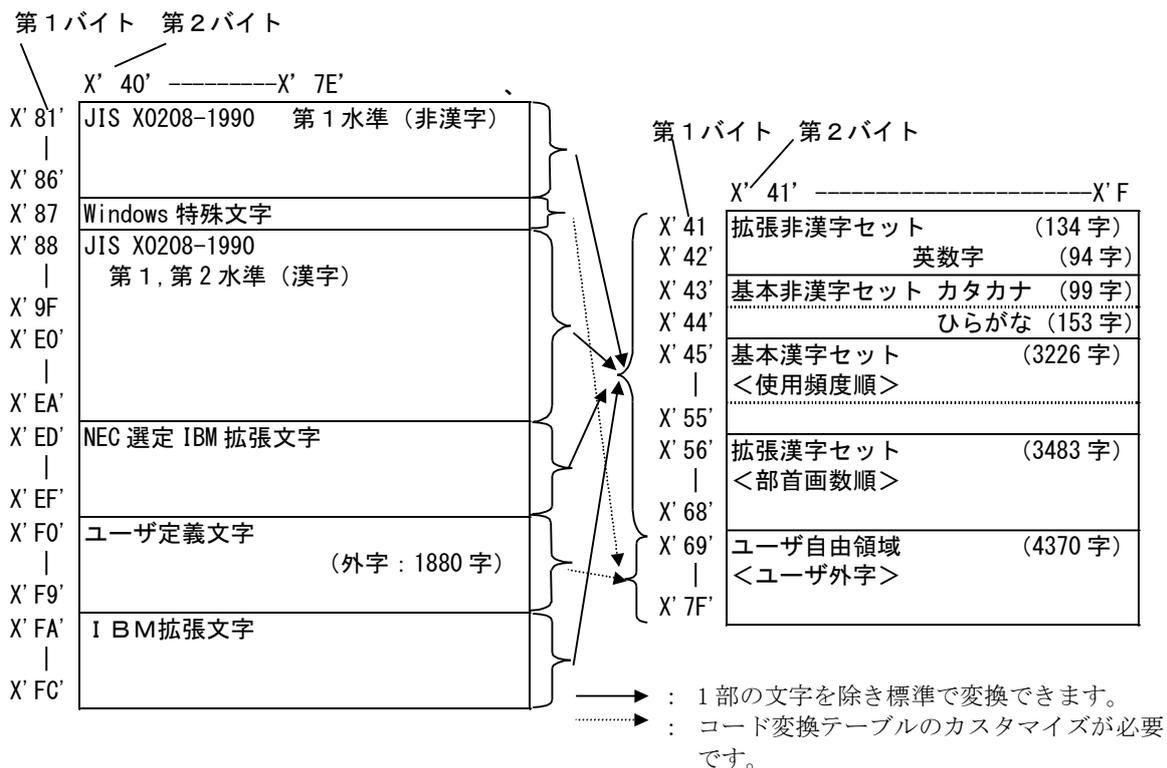


図 8.1.1 シフト J I S コードから I B M 漢字コードへの変換

### 8.1.2. 英数文字フィールド変換 (CHAR:OP19, OP1A)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X' FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.1.3. 漢字フィールド変換(KANJI:OP16)

漢字フィールド変換(OP16)では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード(コード変換テーブルに変換する文字コードが定義されていないコード)については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード(コードの値が X'81' ≤ 第1バイト ≤ X'9F' または X'E0' ≤ 第1バイト ≤ X'FC' かつ X'40' ≤ 第2バイト ≤ X'7E' または X'80' ≤ 第2バイト ≤ X'FC' の範囲でない)については、未定義文字コードと同じ扱いとなります。

ただし、空白文字(X'20')は除きます。

### 8.1.4. 漢字フィールド変換(KANJI:OP17, OP18)

漢字フィールド変換(OP17, OP18)では、変換前の文字コードの内容により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OP17, OP18)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については漢字フィールド変換(OP16)に準じますが、以下の点が異なります。

#### (1) シフトコードの挿入

IBM漢字コードではシフトコード(X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始)により1バイトコード、2バイトコードの切り替えを行なうため、変換時にシフトコードを挿入しますので、データ長が長くなります。このため、コード変換の結果がレコード長を超える場合、エラーとなります。

また、編集オプションファイルのコード変換モードに TRAN\_SHIFT\_NO を指定することにより、シフトコードを挿入しない変換も指定できます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (otm)

### 8.1.5. コード変換テーブルのカスタマイズ

シフトJISコードからIBMメインフレームコードへ変換を行う場合、通常、表 8.1.2に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コードを使用する場合や、マイクロソフト標準キャラクタセットの独自文字であるWindows特殊文字をIBM漢字コードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 8.1.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
SJISIBM	JIS8、シフトJISコードから EBCDIK/EBCDIC コード、IBM漢字コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDIKコードの定義

標準提供のEBCDIKコードへのコード変換テーブルでは、英小文字については未定義文字扱いとなります。英小文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) ユーザ定義文字領域の定義

標準提供のコード変換テーブルでは、Windows特殊文字及び外字コードについては変換の定義を行っていませんので、コード変換テーブルのカスタマイズを行ってください。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.2. シフトJISコードから日立メインフレームコードへの変換

#### 8.2.1. 文字コード変換の概要

シフトJISコードから日立メインフレームコードへのコード変換では、1バイトコードについては、JIS8コードをEBCDIC(英小文字)コード及びEBCDIK(英小文字、カナ)コードへ、また、2バイトコードについては、シフトJISコードをKEISコードへ変換します。

Code Converter では、フィールドの内容に合わせて、表 8.2.1に示す種類の文字コード変換を提供しています。

表 8.2.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
JIS8 コード	JIS8 から EBCDIC コードへ変換	CHAR:OP29, OP39
	JIS8 から EBCDIK コードへ変換	CHAR:OP2A, OP3A
シフトJISコード	シフトJISコードから KEIS78 コードへ変換	KANJI:OP26
	シフトJISコードから KEIS83/90 コードへ変換	KANJI:OP36
JIS8 とシフトJISコード	JIS8 とシフトJISコードの混在したデータを、EBCDIC と KEIS78 コードに変換	KANJI:OP27
	JIS8 とシフトJISコードの混在したデータを、EBCDIC と KEIS83/90 コードに変換	KANJI:OP37
	JIS8 とシフトJISコードの混在したデータを、EBCDIK と KEIS78 コードに変換	KANJI:OP28
	JIS8 とシフトJISコードの混在したデータを、EBCDIK と KEIS83/90 コードに変換	KANJI:OP38

コード変換はすべて、コード変換テーブルを用いて行います。シフトJISコード(マイクロソフト標準キャラクタセットを標準としています)は、図 8.2.1に示すようにKEIS漢字コード領域に変換します。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

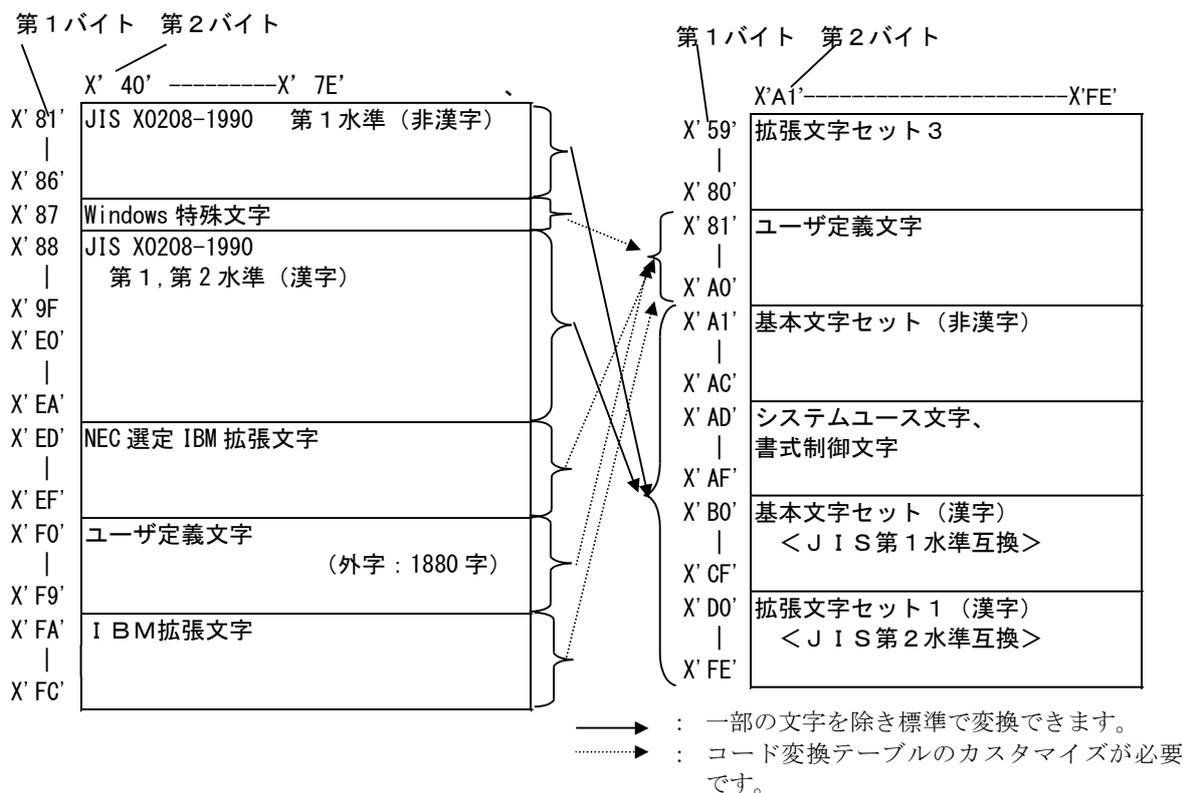


図 8.2.1 シフト J I S コードから K E I S コードへの変換

### 8.2.2. 英数文字フィールド変換 (CHAR:OP29, OP39, OP2A, OP3A)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従って行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.2.3. 漢字フィールド変換(KANJI:OP26, OP36)

漢字フィールド変換(OP26, OP36)では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード(コード変換テーブルに変換する文字コードが定義されていないコード)については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード(コードの値が X'81' ≤ 第1バイト ≤ X'9F' または X'E0' ≤ 第1バイト ≤ X'FC' かつ X'40' ≤ 第2バイト ≤ X'7E' または X'80' ≤ 第2バイト ≤ X'FC' の範囲でない)については、未定義文字コードと同じ扱いとなります。

ただし、空白文字(X'20')は除きます。

#### (3) 空白文字コードの扱い

全角の空白文字(X'8140')及び偶数バイトの半角空白文字(X'20')については、通常は、全角の空白文字として、X'A1A1'に変換します。

なお、編集オプションファイルにTRAN\_SGLSPACE オプションを指定することにより、X'4040'にすることもできます。

また、編集オプションファイルにTRAN\_MIXED\_SPACE オプションを指定することにより、X'8140'全角の空白文字は KEIS の全角の空白文字コード(X'A1A1')に変換し、1バイト X'20'については半角の空白文字として KEIS の半角の空白文字コード(X'40')に変換します。

### 8.2.4. 漢字フィールド変換(KANJI:OP27, OP37, OP28, OP38)

漢字フィールド変換(OP27, OP37, OP28, OP38)では、変換前の文字コードの内容により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OP27, OP37, OP28, OP38)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については漢字フィールド変換(OP26, OP36)に準じますが、以下の点が異なります。

#### (1) シフトコードの挿入

KEISコードではシフトコード(X'0A42':2バイトコード開始, X'0A41':1バイトコード開始)により1バイトコード、2バイトコードの切り替えを行なうため、変換時にシフトコードを挿入しますので、データ長が長くなります。このため、コード変換の結果がフィールド長を超える場合、エラーとなります。

また、編集オプションファイルのコード変換モードにTRAN\_SHIFT\_NOを指定することにより、シフトコードを挿入しない変換も指定できます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.2.5. コード変換テーブルのカスタマイズ

シフト J I S コードから日立メインフレームコードへ変換を行う場合、通常 表 8.2.2 に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コードを使用する場合や、マイクロソフト標準キャラクタセットの独自文字である W i n d o w s 特殊文字を I B M 漢字コードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 8.2.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
SJISKS78	J I S 8、シフト J I S コードから EBCDIK/EBCDIC コード、KEIS78 コードへの変換情報を格納しています。
SJISKS90	J I S 8、シフト J I S コードから EBCDIK/EBCDIC コード、KEIS83/90 コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) E B C D I K コードの定義

標準提供の E B C D I K コードへのコード変換テーブルでは、英小文字については J I S 8 コードとのコード変換を行う際、情報の欠落を防ぐために未使用の部分に割り当てています。英小文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) K E I S コードの定義

標準提供のコード変換テーブルでは、W i n d o w s 特殊文字及び外字コードについては変換の定義を行っていませんので、コード変換テーブルのカスタマイズを行ってください。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.3. シフトJISコードから富士通メインフレームコードへの変換

#### 8.3.1. 文字コード変換の概要

シフトJISコードから富士通メインフレームコードへの文字コード変換では、1バイトコードについては、JIS8コードをEBCDIC (英小文字) コード及びEBCDIK (カナ) コードへ、また、2バイトコードについては、シフトJISコードをJEFコードへ変換します。

Code Converter では、フィールドの内容に合わせて、表 8.3.1に示す種類の文字コード変換を提供しています。

表 8.3.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
JIS8 コード	JIS8 から EBCDIC (英小文字) コードへ変換	CHAR:OP49
	JIS8 から EBCDIK (カナ) コードへ変換	CHAR:OP4A
シフトJIS コード	シフトJIS コードから JEF コードへ変換	KANJI:OP46
JIS8 とシフトJIS コード	JIS8 とシフトJIS コードの混在したデータを、EBCDIC (英小文字) と JEF コードに変換	KANJI:OP47
	JIS8 とシフトJIS コードの混在したデータを、EBCDIK (カナ) と JEF コードに変換	KANJI:OP48

コード変換はすべて、コード変換テーブルを用いて行います。シフトJISコード (マイクロソフト標準キャラクタセットを標準としています) は、図 8.3.1に示すようにJEFコード領域に変換します。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

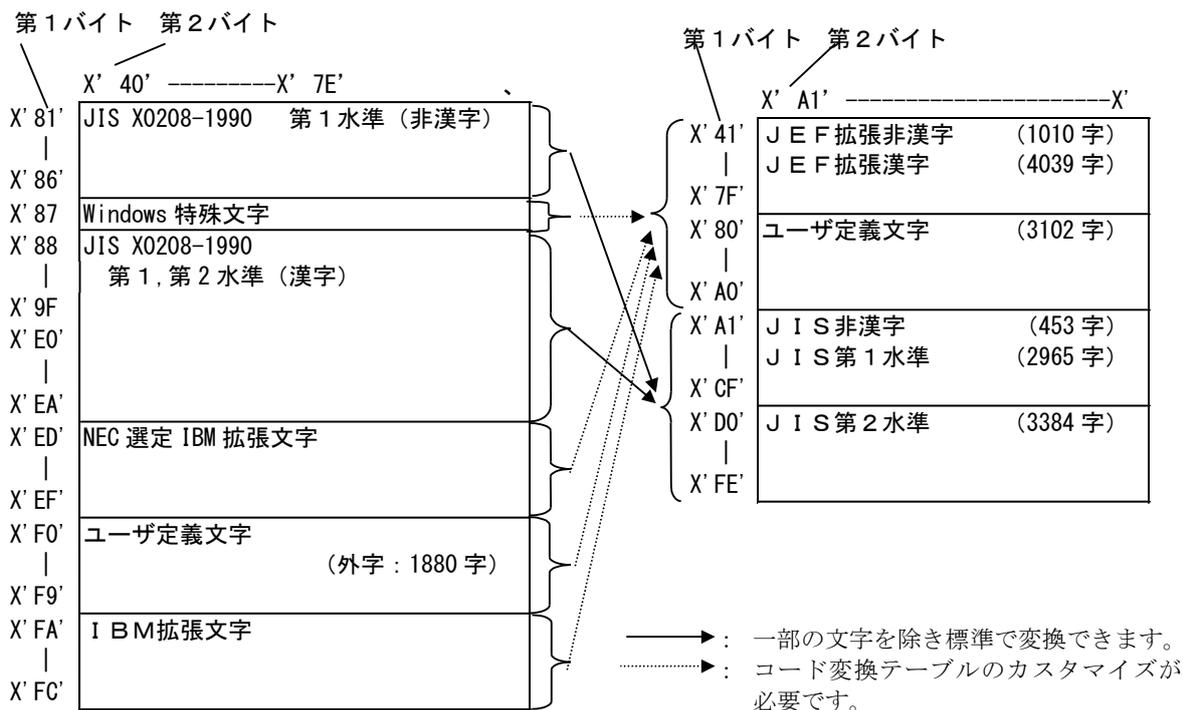


図 8.3.1 シフト J I S コードから J E F コードへの変換

### 8.3.2. 英数文字フィールド変換 (CHAR:OP49, OP4A)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X' FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.3.3. 漢字フィールド変換 (KANJI:OP46)

漢字フィールド変換 (OP46) では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FFFF' に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード (コードの値が X'81' ≤ 第1バイト ≤ X'9F' または X'E0' ≤ 第1バイト ≤ X'FC' かつ X'40' ≤ 第2バイト ≤ X'7E' または X'80' ≤ 第2バイト ≤ X'FC' の範囲でない) については、未定義文字コードと同じ扱いとなります。

ただし、空白文字 (X'20') は除きます。

#### (3) 空白文字コードの扱い

全角の空白文字 (X'8140') 及び偶数バイトの半角空白文字 (X'20') については、通常は、全角の空白文字として、X'4040' に変換します。

なお、編集オプションファイルに TRAN\_DBLSPACE オプションを指定することにより、X'A1A1' にすることもできます。

### 8.3.4. 漢字フィールド変換 (KANJI:OP47, OP48)

漢字フィールド変換 (OP47, OP48) では、変換前の文字コードの内容により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換 (OP47, OP48) では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については漢字フィールド変換 (OP46) に準じますが、以下の点が異なります。

#### (1) シフトコードの挿入

J E F コードではシフトコード (X'28' : 1 2ポイント2バイトコード開始, X'29' : 1バイトコード開始) により1バイトコード、2バイトコードの切り替えを行なうため、変換時にシフトコードを挿入しますので、データ長が長くなります。このため、コード変換の結果がレコード長を超える場合、エラーとなります。

また、編集オプションファイルのコード変換モードに TRAN\_SHIFT\_NO を指定することにより、シフトコードを挿入しない変換や、TRAN\_SHIFT\_38 を指定することにより、2バイトのシフトコードを X'38' : 9ポイント2バイトコード開始にする変換もできます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.3.5. コード変換テーブルのカスタマイズ

シフト J I S コードから富士通メインフレームコードへ変換を行う場合、通常、表 8.3.2 に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コードを使用する場合や、マイクロソフト標準キャラクタセットの独自文字である W i n d o w s 特殊文字を J E F コードに変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 8.3.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
SJISJEF	J I S 8、シフト J I S コードから EBCDIK/EBCDIC コード、JEF コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDIC コードの定義

標準提供の EBCDIC コードへのコード変換テーブルでは、カナ文字については未定義文字扱いとなります。カナ文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (c) EBCDIK コードの定義

標準提供の EBCDIK コードへのコード変換テーブルでは、英小文字については JIS8 コードとのコード変換を行う際、情報の欠落を防ぐために未使用の部分に割り当てています。英小文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) JEF コードの定義

標準提供のコード変換テーブルでは、W i n d o w s 特殊文字及び外字コードについては変換の定義を行っていませんので、コード変換テーブルのカスタマイズを行ってください。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.4. EUCコードからIBMメインフレームコードへの変換

#### 8.4.1. 文字コード変換の概要

EUCコードからIBMメインフレームコードへのコード変換では、1バイトコードについては、EUCコードセット0/2をEBCDIC (英小文字、カナ) コード及びEBCDIK (カナ) コードへ、また、2バイトコードについては、EUCコードセット1/3をIBM漢字コードへ変換します。

Code Converter では、フィールドの内容に合わせて、表 8.4.1に示す種類の文字コード変換を提供しています。

表 8.4.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
EUC コードセット 0/2	EUC コードセット 0/2 から EBCDIC コードへ変換	CHAR:OPC9
	EUC コードセット 0/2 から EBCDIK コードへ変換	CHAR:OPCA
EUC コードセット 1/3	EUC コードセット 1/3 から IBM 漢字コードへ変換	KANJI:OPC6
EUC コードセット 0/1/2/3	EUC コードセット 0/1/2/3 の混在したデータを、EBCDIC と IBM 漢字コードに変換	KANJI:OPC7
	EUC コードセット 0/1/2/3 の混在したデータを、EBCDIK と IBM 漢字コードに変換	KANJI:OPC8

コード変換はすべて、コード変換テーブルを用いて行います。EUCコード (JIS X0208-1990規格を想定)は、図 8.4.1に示すようにIBM漢字コード領域に変換します。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

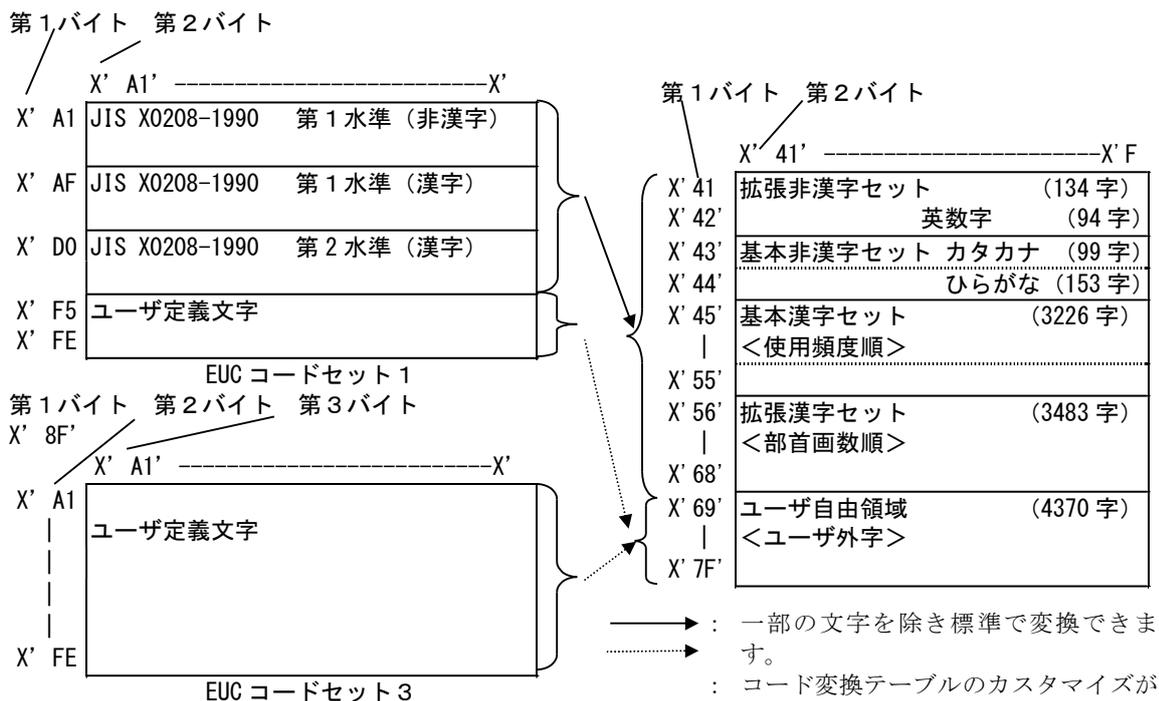


図 8.4.1 EUCコードからIBM漢字コードへの変換

### 8.4.2. 英数文字フィールド変換 (CHAR:OPC9, OPC8)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X' FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.4.3. 漢字フィールド変換 (KANJI:OPC6)

漢字フィールド変換 (OPC6) では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード (コードの値が X'A1' ≤ 第1バイト ≤ X'FE' または X'A1' ≤ 第2バイト ≤ X'FE' の範囲でない) については、未定義文字コードと同じ扱いとなります。

ただし、空白文字 (X'20') は除きます。

### 8.4.4. 漢字フィールド変換 (KANJI:OPC7, OPC8)

漢字フィールド変換 (OPC7, OPC8) では、変換前の文字コードの内容により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換 (OPC7, OPC8) では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については漢字フィールド変換 (OPC6) に準じますが、以下の点が異なります。

#### (1) シフトコードの挿入

IBM漢字コードではシフトコード (X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始) により1バイトコード、2バイトコードの切り替えを行なうため、変換時にシフトコードを挿入しますので、データ長が変換前と一致しない場合があります。そのためコード変換の結果がフィールド長より短くなる場合は、左詰めにコード変換の結果を設定し、更にフィールドの残りバイトには X'FF' を設定します。また、コード変換の結果がフィールド長を超える場合は、変換可能な範囲までを変換しあふれる部分については変換しません。

また、編集オプションファイルのコード変換モードに TRAN\_SHIFT\_NO を指定することにより、シフトコードを挿入しない変換も指定できます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (otm)

### 8.4.5. コード変換テーブルのカスタマイズ

EUCコードからIBMメインフレームコードへ変換を行う場合、通常、表 8.4.2に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コードを使用する場合や、JIS X0208-1990規格にない文字をIBM漢字コードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 8.4.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
EUCIBM	EUCコードセット0/1/2/3からEBCDIK/EBCDICコード、IBM漢字コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDIKコードの定義

標準提供のEBCDIKコードへのコード変換テーブルでは、英小文字については未定義文字扱いとなります。英小文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) ユーザ定義文字領域の定義

標準提供のコード変換テーブルでは、外字コードについては変換の定義を行っていませんので、コード変換テーブルのカスタマイズを行ってください。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.5. UNICODEからIBMメインフレームコードへの変換

#### 8.5.1. 文字コード変換の概要

UNICODEからIBMメインフレームコードへのコード変換では、英数文字コードについては、UNICODEをEBCDIC (英小文字、カナ) コード及びEBCDIK (カナ) コードへ、また、漢字コードについては、UNICODEをIBM漢字コードへ変換します。

Code Converter では、フィールドの内容に合わせて、表 8.5.1に示す種類の文字コード変換を提供しています。

表 8.5.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
UNICODE	英数文字の UNICODE データから EBCDIC コードへ変換	CHAR:OP69
	英数文字の UNICODE データから EBCDIK コードへ変換	CHAR:OP6A
UNICODE	漢字コードの UNICODE データから IBM 漢字コードへ変換	KANJI:OP66
UNICODE	英数文字と漢字コードの混在した UNICODE データを、EBCDIC と IBM 漢字コードに変換	KANJI:OP67
	英数文字と漢字コードの混在した UNICODE データを、EBCDIK と IBM 漢字コードに変換	KANJI:OP68

コード変換はすべて、コード変換テーブルを用いて行い、UNICODEは、JIS X 0208-1990規格を想定しています。

#### 8.5.2. 英数文字フィールド変換 (CHAR:OP69, OP6A)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

##### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FF'に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

#### 8.5.3. 漢字フィールド変換 (KANJI:OP66)

漢字フィールド変換(OP66)では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行いますが、以下のデータについては扱いが異なります。

##### (1) 未定義文字コード

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

##### (2) 不当文字コードの扱い

不当文字コード (0x0100~0x036F, 0x03D0~0x03FF, 0x0500~0x1FFF, 0x2070~0x20FF, 0x2400~0x24FF, 0x2700~0x2FFF, 0x3100~0x31FF, 0x3300~0x4DFF, 0xA000~0xF8FF, 0xFB00~0xFEFF) については、未定義文字コードと同じ扱いとなります。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.5.4. 漢字フィールド変換(KANJI:OP67, OP68)

漢字フィールド変換(OP67, OP68)では、変換前の文字コードの内容により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OP67, OP68)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については漢字フィールド変換(OP66)に準じますが、以下の点が異なります。

#### (1) シフトコードの挿入

IBM漢字コードではシフトコード (X'0E' : 2バイトコード開始, X'0F' : 1バイトコード開始) により1バイトコード、2バイトコードの切り替えを行なうため、変換時にシフトコードを挿入しますので、データ長が長くなります。このため、コード変換の結果がフィールド長を超える場合、あふれる部分については変換しません。

また、編集オプションファイルのコード変換モードに TRAN\_SHIFT\_NO を指定することにより、シフトコードを挿入しない変換も指定できます。

### 8.5.5. コード変換テーブルのカスタマイズ

UNICODEからIBMメインフレームコードへ変換を行う場合、通常、表 8.5.2に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コードを使用する場合や、JIS X 0208-1990規格の文字セット以外の特殊文字をIBM漢字コードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 8.5.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
UNIIBM	UNICODE から EBCDIK/EBCDIC コード、IBM漢字コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDIKコードの定義

標準提供のEBCDIKコードへのコード変換テーブルでは、英小文字については未定義文字扱いとなります。英小文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) ユーザ定義文字領域の定義

標準提供のコード変換テーブルでは、外字コードについては変換の定義を行っていませんので、コード変換テーブルのカスタマイズを行ってください。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.6. UNICODEから日立メインフレームコードへの変換

#### 8.6.1. 文字コード変換の概要

UNICODEから日立メインフレームコードへのコード変換では、英数文字コードについては、UNICODEをEBCDIC(英小文字)コード及びEBCDIK(英小文字、カナ)コードへ、また、漢字コードについては、UNICODEをKEISコードへ変換します。

Code Converter では、フィールドの内容に合わせて、表 8.6.1に示す種類の文字コード変換を提供しています。

表 8.6.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
UNICODE	英数文字の UNICODE データから EBCDIC コードへ変換	CHAR:OP79, OP89
	英数文字の UNICODE データから EBCDIK コードへ変換	CHAR:OP7A, OP8A
UNICODE	漢字コードの UNICODE データから KEIS78 コードへ変換	KANJI:OP76
	漢字コードの UNICODE データから KEIS83/90 コードへ変換	KANJI:OP86
UNICODE	英数文字と漢字コードの混在した UNICODE データを、EBCDIC と KEIS78 コードに変換	KANJI:OP77
	英数文字と漢字コードの混在した UNICODE データを、EBCDIC と KEIS83/90 コードに変換	KANJI:OP87
	英数文字と漢字コードの混在した UNICODE データを、EBCDIK と KEIS78 コードに変換	KANJI:OP78
	英数文字と漢字コードの混在した UNICODE データを、EBCDIK と KEIS83/90 コードに変換	KANJI:OP88

コード変換はすべて、コード変換テーブルを用いて行い、UNICODEは、JIS X 0208-1990規格を想定しています。

#### 8.6.2. 英数文字フィールド変換 (CHAR:OP79, OP89, OP7A, OP8A)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。

コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

##### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FF'に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.6.3. 漢字フィールド変換 (KANJI:OP76, OP86)

漢字フィールド変換(OP76, OP86)では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード(コード変換テーブルに変換する文字コードが定義されていないコード)については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード(0x0100~0x036F, 0x03D0~0x03FF, 0x0500~0x1FFF, 0x2070~0x20FF, 0x2700~0x2FFF, 0x3100~0x31FF, 0x3300~0x4DFF (KEIS78 の場合)、0x33D0~0x4DFF (KEIS83/90 の場合)、0xA000~0xF8FF, 0xFA30~0xFEFF (KEIS78 の場合)、0xFB00~0xFEFF (KEIS83/90 の場合))については、未定義文字コードと同じ扱いとなります。

#### (3) 空白文字コードの扱い

全角の空白文字(X'8140')及び偶数バイトの半角空白文字(X'20')については、通常は、全角の空白文字として、X'A1A1'に変換します。

なお、編集オプションファイルにTRAN\_SGLSPACE オプションを指定することにより、X'4040'にすることもできます。

### 8.6.4. 漢字フィールド変換 (KANJI:OP77, OP87, OP78, OP88)

漢字フィールド変換(OP77, OP87, OP78, OP88)では、変換前の文字コードの内容により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OP77, OP87, OP78, OP88)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については漢字フィールド変換(OP76, OP77)に準じますが、以下の点が異なります。

#### (1) シフトコードの挿入

KEISコードではシフトコード(X'0A42':2バイトコード開始, X'0A41':1バイトコード開始)により1バイトコード、2バイトコードの切り替えを行なうため、変換時にシフトコードを挿入しますので、データ長が長くなります。このため、コード変換の結果がフィールド長を超える場合、あふれる部分については変換しません。

また、編集オプションファイルのコード変換モードにTRAN\_SHIFT\_NOを指定することにより、シフトコードを挿入しない変換も指定できます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (otm)

### 8.6.5. コード変換テーブルのカスタマイズ

UNICODEから日立メインフレームコードへ変換を行う場合、通常表 8.6.2に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コードを使用する場合や、JIS X 0208-1990規格の文字セット以外の特殊文字をKEISコードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 8.6.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
UNIKS78	UNICODE から EBCDIK/EBCDIC コード、KEIS78 コードへの変換情報を格納しています。
UNIKS90	UNICODE から EBCDIK/EBCDIC コード、KEIS83/90 コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDIKコードの定義

標準提供のEBCDIKコードへのコード変換テーブルでは、英小文字については UNICODE とのコード変換を行う際、情報の欠落を防ぐために未使用の部分に割り当てています。英小文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) KEISコードの定義

標準提供のコード変換テーブルでは、外字コードについては変換の定義を行っていませんので、コード変換テーブルのカスタマイズを行ってください。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.7. UNICODEから富士通メインフレームコードへの変換

#### 8.7.1. 文字コード変換の概要

UNICODEから富士通メインフレームコードへの文字コード変換では、英数文字コードについては、UNICODEをEBCDIC (英小文字) コード及びEBCDIK (カナ) コードへ、また、漢字コードについては、UNICODEをJEFコードへ変換します。

Code Converter では、フィールドの内容に合わせて、表 8.7.1に示す種類の文字コード変換を提供しています。

表 8.7.1 コード変換の種類

入力フィールド内容	コード変換の内容	フィールド種別と変換オプション
UNICODE	英数文字の UNICODE データから EBCDIC コードへ変換	CHAR:OP99
	英数文字の UNICODE データから EBCDIK コードへ変換	CHAR:OP9A
UNICODE	漢字コードの UNICODE データから JEF コードへ変換	KANJI:OP96
UNICODE	英数文字と漢字コードの混在した UNICODE データを、EBCDIC と JEF コードに変換	KANJI:OP97
	英数文字と漢字コードの混在したデータを、EBCDIK と JEF コードに変換	KANJI:OP98

コード変換はすべて、コード変換テーブルを用いて行い、UNICODEは、JIS X 0208-1990規格を想定しています。

#### 8.7.2. 英数文字フィールド変換 (CHAR:OP99, OP9A)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

##### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X'FF'に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.7.3. 漢字フィールド変換(KANJI:OP96)

漢字フィールド変換(OP96)では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行います。以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード(0x0100~0x036F、0x03D0~0x03FF、0x0500~0x1FFF、0x2070~0x20FF、0x2400~0x24FF、0x2700~0x2FFF、0x3100~0x31FF、0x3300~0x4DFF、0xA000~0xF8FF、0xFB00~0xFEFF)については、X'FFFF'に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード(コードの値が X'81' ≤ 第1バイト ≤ X'9F' または X'E0' ≤ 第1バイト ≤ X'FC' かつ X'40' ≤ 第2バイト ≤ X'7E' または X'80' ≤ 第2バイト ≤ X'FC' の範囲でない)については、未定義文字コードと同じ扱いとなります。

ただし、空白文字(X'20')は除きます。

#### (3) 空白文字コードの扱い

全角の空白文字(X'8140')及び偶数バイトの半角空白文字(X'20')については、通常は、全角の空白文字として、X'4040'に変換します。

なお、編集オプションファイルにTRAN\_DBLSPACE オプションを指定することにより、X'A1A1'にすることもできます。

### 8.7.4. 漢字フィールド変換(KANJI:OP97, OP98)

漢字フィールド変換(OP97, OP98)では、変換前の文字コードの内容により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換(OP97, OP98)では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については漢字フィールド変換(OP96)に準じますが、以下の点が異なります。

#### (1) シフトコードの挿入

J E Fコードではシフトコード(X'28' : 1 2ポイント2バイトコード開始, X'29' : 1バイトコード開始)により1バイトコード、2バイトコードの切り替えを行なうため、変換時にシフトコードを挿入しますので、データ長が長くなります。このため、コード変換の結果がフィールド長を超える場合、あふれる部分については変換しません。

また、編集オプションファイルのコード変換モードにTRAN\_SHIFT\_NOを指定することにより、シフトコードを挿入しない変換や、TRAN\_SHIFT\_38を指定することにより、2バイトのシフトコードをX'38' : 9ポイント2バイトコード開始にする変換もできます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.7.5. コード変換テーブルのカスタマイズ

UNICODEから富士通メインフレームコードへ変換を行う場合、通常、表 8.7.2に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コードを使用する場合や、JIS X 0208-1990規格の文字セット以外の特殊文字をJEFコードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 8.7.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
UNIJEF	UNICODE から EBCDIK/EBCDIC コード、JEF コードへの変換情報を格納しています。

#### (1) カスタマイズのポイント

##### (a) EBCDICコードの定義

標準提供のEBCDICコードへのコード変換テーブルでは、カナ文字については未定義文字扱いとなります。カナ文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) EBCDIKコードの定義

標準提供のEBCDIKコードへのコード変換テーブルでは、英小文字についてはUNICODEとのコード変換を行う際、情報の欠落を防ぐために未使用の部分に割り当てています。英小文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (c) JEFコードの定義

標準提供のコード変換テーブルでは、外字コードについては変換の定義を行っていませんので、コード変換テーブルのカスタマイズを行ってください。



## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.8.2. 英数文字フィールド変換 (CHAR:OPD9, OPE9, OPDA, OPEA)

英数文字フィールド変換では、指定されたフィールドのデータすべてを1バイトコードとして変換します。コード変換は、付録に示すコード変換テーブルに従い行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コードの扱い

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X' FF' に変換します。

なお、編集オプションファイルに1バイト置換文字を指定することにより、未定義文字を任意の1バイト文字に置き換えることもできます。

### 8.8.3. 漢字フィールド変換 (KANJI:OPD6, OPE6)

漢字フィールド変換 (OPD6, OPE6) では、データすべてを2バイトコードとしてコード変換テーブルに従いコード変換を行いますが、以下のデータについては扱いが異なります。

#### (1) 未定義文字コード

未定義文字コード (コード変換テーブルに変換する文字コードが定義されていないコード) については、X' FFFF' に変換します。

なお、編集オプションファイルに2バイト置換文字を指定することにより、未定義文字を任意の2バイト文字に置き換えることもできます。

#### (2) 不当文字コードの扱い

不当文字コード (コードの値が X' A1' ≤ 第1バイト ≤ X' FE' または X' A1' ≤ 第2バイト ≤ X' FE' の範囲でない) については、未定義文字コードと同じ扱いとなります。

ただし、空白文字 (X' 20') は除きます。

### 8.8.4. 漢字フィールド変換 (KANJI:OPD7, OPE7, OPD8, OPE8)

漢字フィールド変換 (OPD7, OPE7, OPD8, OPE8) では、変換前の文字コードの内容により、1バイトコード変換と2バイトコード変換を切り替えてコード変換を行います。

漢字フィールド変換 (OPD7, OPE7, OPD8, OPE8) では、1バイトコードの変換については英数文字フィールド変換に、2バイトコード変換については漢字フィールド変換 (OPE6) に準じますが、以下の点が異なります。

#### (1) シフトコードの挿入

KE I S漢字コードではシフトコード (X' 0A42' : 2バイトコード開始, X' 0a41' : 1バイトコード開始) により1バイトコード、2バイトコードの切り替えを行うため、変換時にシフトコードを挿入しますので、データ長が変換前と一致しない場合があります。そのためコード変換の結果がフィールド長より短くなる場合は、左詰めにコード変換の結果を設定し、更にフィールドの残りバイトには X' FF' を設定します。また、コード変換の結果がフィールド長を超える場合は、変換可能な範囲までを変換しあふれる部分については変換しません。

また、編集オプションファイルのコード変換モードに TRAN\_SHIFT\_NO を指定することにより、シフトコードを挿入しない変換も指定できます。

## 8. オープン環境からメインフレーム環境への文字コード変換 (o t m)

### 8.8.5. コード変換テーブルのカスタマイズ

EUCコードから日立メインフレームコードへ変換を行う場合、通常、表 8.8.2に示す標準提供しているコード変換テーブルを使用しますが、お客様の定義された外字コードを使用する場合や、JIS X0208-1990規格にない文字をKEIS漢字コードにコード変換を行う場合、コード変換テーブルをカスタマイズする必要があります。

表 8.8.2 Code Converter の標準提供コード変換テーブル

ファイル名	テーブルの内容
EUCKS78.tbl	EUCコードセット0/1/2/3からEBCDIK/EBCDICコード、KEIS漢字コードへの変換情報を格納しています。
EUCKS90.tbl	

#### (1) カスタマイズのポイント

##### (a) EBCDIKコードの定義

標準提供のEBCDIKコードへのコード変換テーブルでは、英小文字についてはEUCコードとのコード変換を行う際、情報の欠落を防ぐために未使用の部分に割り当てています。英小文字の扱いを変更する場合は、コード変換テーブルのカスタマイズを行ってください。

##### (b) ユーザ定義文字領域の定義

標準提供のコード変換テーブルでは、外字コードについては変換の定義を行っていませんので、コード変換テーブルのカスタマイズを行ってください。

## 9. 数値変換機能について

### 9. 数値変換機能について

この章では、数値変換機能について説明します。

#### 9.1. メインフレーム環境からオープン環境への数値変換 (m t o)

##### 9.1.1. 数値変換の概要

メインフレーム上の数値には、パック形式データ及びゾーン形式データがあります。

Code Converter では、フィールドの内容に合わせて、表 9.1.1に示す種類の数値変換を提供しています。

表 9.1.1 数値変換の種類

入力フィールド内容	数値変換の内容	フィールド種別と変換オプション
パック形式データ	パック形式の数値データを文字形式データへ変換	NUM:OPA1
ゾーン形式データ	ゾーン形式の数値データを文字形式データへ変換	NUM:OPA2

数値変換は、出力先フィールドに対して、右詰めに変換します。

数値変換によって、出力先フィールドが不足している（変換したフィールド有効長の方が長い）場合はエラー（エラーコード：757）と判定します。

逆に、余っている（変換したフィールド有効長の方が短い）場合は、フィールド有効長より前に0もしくはスペースを埋めます。

##### 9.1.2. パック形式変換 (NUM:OPA1)

パック形式変換は、指定されたフィールドの1バイトの数値（パック形式）データを、上位4ビットと下位4ビットに分割し、それぞれ1バイトずつ、計2バイトの文字形式データに変換します。

1バイトのパック形式データにおいて、範囲外（上位/下位4ビット：X'A'~X'F'）の場合は、データ不正（エラーコード：760）と判定します。

パック形式変換は、編集オプションファイルの指定に従い変換します。

###### (1) 符号の扱い

パック形式データの最後の1バイトの下位4ビットが符号ビットです。符号ビットは以下のように判定します。

- ・符号ビットがA、C、E、Fの場合、プラス。
- ・符号ビットがB、Dの場合、マイナス。
- ・上記以外（0~9）の場合、データ不正。（エラーコード：760）

そして、編集オプションファイルに指定した「符号位置」に従い、符号を付けます。

なお、「符号位置」に「符号なし」を指定した場合、実際のデータが負の値（符号ビットがB、Dの場合）であっても、マイナス符号を付けませんのでご注意ください。

ただし、「パックデータの符号処置」に「符号なしパック変換(TRAN\_PACK\_SIGN\_NON)」を指定した場合、パックデータの符号は無条件にプラスとします。そして、符号ビットは以下のように判定します。

- ・符号ビットが0~9の場合、数値データ。
- ・上記以外（A~F）の場合、データ不正。（エラーコード：760）

- ・「符号位置が先頭」の場合



図 9.1.1 符号位置が先頭の変換概要

## 9. 数値変換機能について

- ・「符号位置が終端」の場合



図 9.1.2 符号位置が終端の変換概要

- ・「有効桁の前に符号を付ける」の場合



図 9.1.3 符号位置が有効桁の前の変換概要

- ・「符号位置なし」の場合

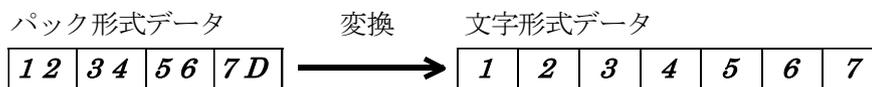


図 9.1.4 符号なしの変換概要

- ・「符号なしパックデータ」、「符号なし」の場合

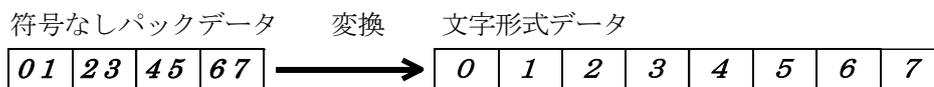


図 9.1.5 符号なしパックデータの変換概要

### (2) 0サプレスの扱い

0サプレスとは、出力先フィールド長に対して、変換したフィールド有効長より前の0をスペースに置き換えることです。0サプレスの扱いは編集オプションファイルに指定した「0サプレスの処置」に従います。0サプレスと符号位置の関連については、次のようになります。

- ・「0サプレス無し」、「符号位置が先頭」の場合

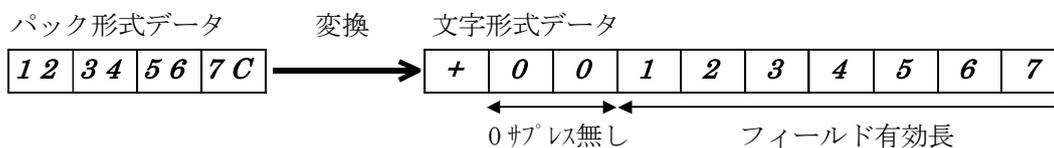


図 9.1.6 0サプレス無し、符号位置が先頭の変換概要

## 9. 数値変換機能について

- ・「0サプレス有り」、「符号位置が終端」の場合



図 9.1.7 0サプレス有り、符号位置が終端の変換概要

### (3) 小数点の扱い

編集オプションファイルに指定した「小数点桁数」に従い、小数点を付けます。小数点桁数が0の場合は、小数点無しと判定し、小数点は付けません。

- ・「小数点桁数：3」、「0サプレス有り」、「有効桁の前に符号を付ける」の場合

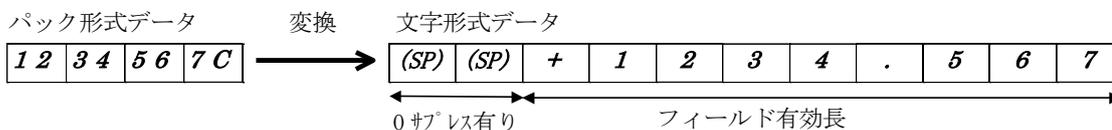


図 9.1.8 小数点桁数3、0サプレス有り、符号位置が有効桁の前の変換概要

- ・「小数点桁数：0」、「0サプレス無し」、「符号なし」の場合

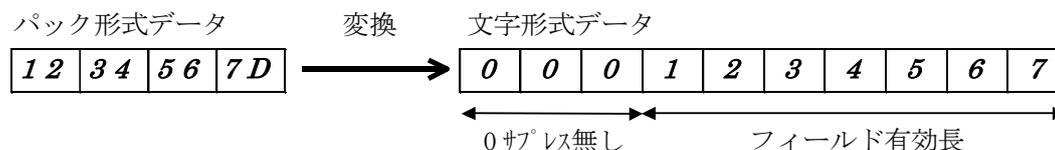


図 9.1.9 小数点桁数0、0サプレス無し、符号なしの変換概要

### (4) オール0の扱い

指定されたフィールドがオール0の場合、編集オプションファイルに指定した「オールスペースの処置」に従います。

「オール0はそのまま0に変換する (TRAN\_ALLSPACE\_NORMAL)」を指定した場合、通常の数値変換どおりに0に変換します。

「オール0はオールスペースに変換する (TRAN\_ALLSPACE\_ZERO)」を指定した場合は、図 9.1.10に示すように、オールスペース (全てX'20') に変換します。その際、編集オプションファイルに指定した「符号位置」「0サプレスの処置」「小数点桁数」ならびに入力データの符号は無視します。

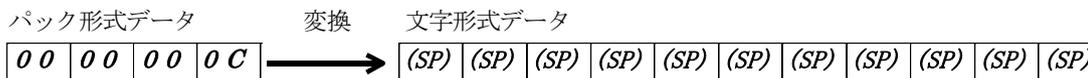


図 9.1.10 オール0をオールスペースに変換する場合の概要

## 9. 数値変換機能について

### 9.1.3. ゾーン形式変換 (NUM:OPA2)

ゾーン形式変換は、指定されたフィールドの1バイトの数値（ゾーン形式）データの低位4ビットを、1バイトの文字形式データに変換します。

1バイトのゾーン形式データにおいて、範囲外（低位4ビット：X'A'～X'F'）の場合はデータ不正（エラーコード：760）と判定します。

ゾーン形式変換は、編集オプションファイルの指定に従い変換します。

#### (1) 符号の扱い

ゾーン形式の数値データの最後の1バイトの上位4ビットが符号ビットです。符号ビットは以下のように判定します。

- ・符号ビットがA、C、E、Fの場合、プラス。
- ・符号ビットがB、Dの場合、マイナス。
- ・上記以外（0～9）の場合、データ不正。（エラーコード：760）

そして、編集オプションファイルに指定した「符号位置」に従い、符号を付けます。

なお、「符号位置」に「符号なし」を指定した場合、実際のデータが負の値（符号ビットがB、Dの場合）であっても、マイナス符号を付けませんのでご注意ください。

また、「パックデータの符号処置」に「符号なし変換 (TRAN\_PACK\_SIGN\_NON)」を指定した場合は、その指示パラメータを無効とし符号付きと判定します。

- ・「符号位置が先頭」の場合



図 9.1.11 符号位置が先頭の変換概要

- ・「符号位置が終端」の場合

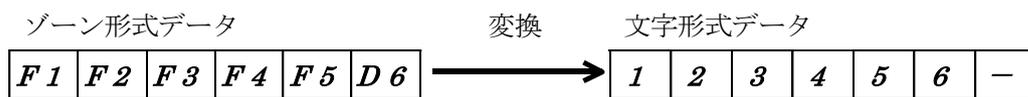


図 9.1.12 符号位置が終端の変換概要

- ・「有効桁の前に符号を付ける」の場合



図 9.1.13 符号位置が有効桁の前の変換概要

## 9. 数値変換機能について

- ・「符号なし」の場合

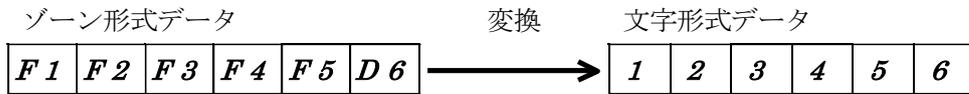


図 9.1.14 符号なしの変換概要

### (2) 0サプレスの扱い

0サプレスとは出力先フィールド長に対して、変換したフィールド有効長より前の0をスペースに置き換えることです。0サプレスの扱いは編集オプションファイルに指定した「0サプレスの処置」に従います。

0サプレスと符号位置の関連については、次のようになります。

- ・「0サプレス無し」、「符号位置が先頭」の場合

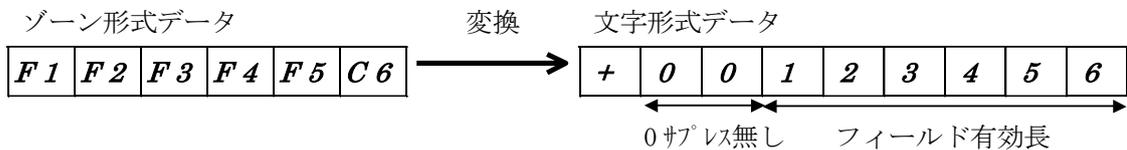


図 9.1.15 0サプレス無し、符号位置が先頭の変換概要

- ・「0サプレス有り」、「符号位置が終端」の場合

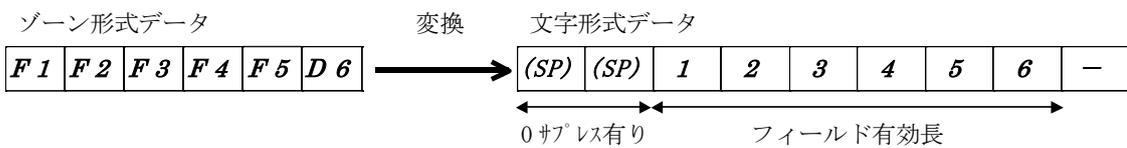


図 9.1.16 0サプレス有り、符号位置が終端の変換概要

### (3) 小数点の扱い

編集オプションファイルに指定した「小数点桁数」に従い、小数点を付けます。小数点桁数が0の場合は、小数点無しと判定し、小数点は付けません。

- ・「小数点桁数：3」、「0サプレス有り」、「有効桁の前に符号を付ける」の場合

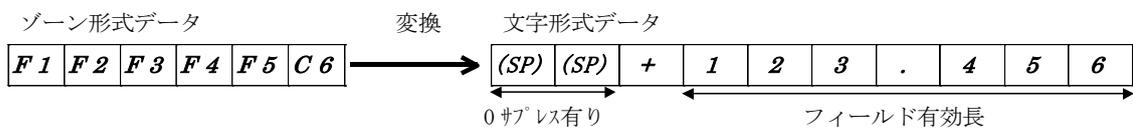


図 9.1.17 小数点桁数3、0サプレス無し、符号位置が有効桁の前の変換概要

## 9. 数値変換機能について

- ・「小数点桁数：0」、「0サプレス無し」、「符号なし」の場合

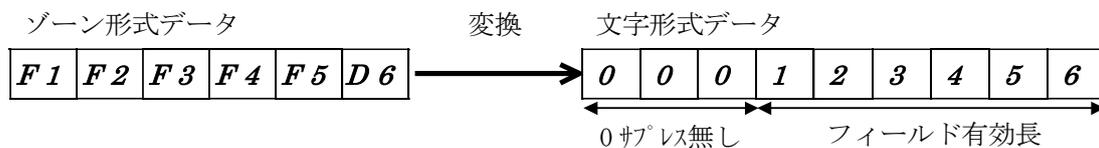


図 9.1.18 小数点桁数0、0サプレス無し、符号なしの変換概要

### (4) オール0の扱い

指定されたフィールドがオール0の場合、編集オプションファイルに指定した「オールスペースの処置」に従います。

「オール0はそのまま0に変換する (TRAN\_ALLSPACE\_NORMAL)」を指定した場合、通常の数値変換どおりに0に変換します。

「オール0はオールスペースに変換する (TRAN\_ALLSPACE\_ZERO)」を指定した場合は、図 9.1.19 に示すように、オールスペース (全てX'20') に変換します。その際、編集オプションファイルに指定した「符号位置」「0サプレスの処置」「小数点桁数」ならびに入力データの符号は無視します。

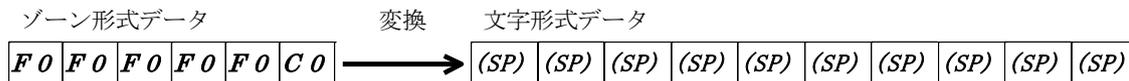


図 9.1.19 オール0をオールスペースに変換する場合の概要

## 9. 数値変換機能について

### 9.2. オープン環境からメインフレーム環境への数値変換 (o t m)

#### 9.2.1. 数値変換の概要

オープン上の数値は、文字形式データです。

Code Converter では、フィールドの内容に合わせて、表 9.2.1 に示す種類の数値変換を提供しています。

表 9.2.1 数値変換の種類

入力フィールド内容	数値変換の内容	フィールド種別と変換オプション
文字形式データ	文字形式データをパック形式の数値データへ変換	NUM:OPA6
	文字形式データをゾーン形式の数値データへ変換	NUM:OPA7

数値変換は、出力先フィールドに対して、右詰めに変換します。

数値変換によって、出力先フィールドが不足している（変換したフィールド有効長の方が長い）場合はエラー（エラーコード：757）と判定します。

逆に、余っている（変換したフィールド有効長の方が短い）場合は、フィールド有効長より前に0を埋めます。

#### 9.2.2. パック形式変換 (NUM:OPA6)

パック形式変換は、指定されたフィールドの2バイトの文字形式データについて、第1バイト目を上位4ビット、第2バイト目を下位4ビットに変換し、1バイトの数値（パック形式）データに結合します。

2バイトの文字形式データにおいて、範囲外（第1／第2バイト：X'30'～X'39'以外）の場合はデータ不正（エラーコード：760）と判定します。

パック形式変換は、編集オプションファイルの指定に従い変換します。

##### (1) 符号の扱い

編集オプションファイルに指定した「符号位置」に従い、文字形式データの符号の判定を行います。指定された位置にプラス／マイナスがない場合は、符号不正（エラーコード：758）と判定します。また「符号なし (TRAN\_SIGN\_NON)」を指定した場合は、無条件にプラスとします。

プラス／マイナスの符号は、パック形式データの最後の1バイトの下位4ビットに符号ビットとして変換します。符号ビットは、編集オプションファイルの指定により、以下のような内容に変換します。

- ・符号がプラスの場合、符号ビットは「プラス符号種別」で指定したパラメータ。
- ・符号がマイナスの場合、符号ビットは「マイナス符号種別」で指定したパラメータ。

ただし、「パックデータの符号処置」に「符号なしパック変換 (TRAN\_PACK\_SIGN\_NON)」を指定した場合、パックデータの符号は無条件にプラスとします。そして、符号ビットは以下のように判定します。

- ・符号ビットが0～9の場合、数値データ。
- ・上記以外 (A～F) の場合、データ不正。（エラーコード：760）

- ・「符号位置が先頭」の場合

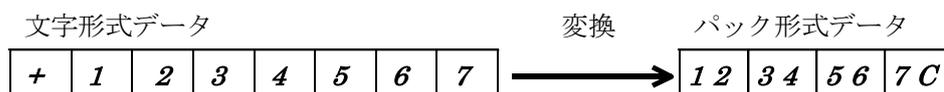


図 9.2.1 符号位置が先頭の変換概要

## 9. 数値変換機能について

- 「符号位置が終端」の場合

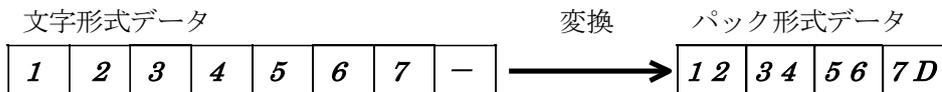


図 9.2.2 符号位置が終端の変換概要

- 「有効桁の前に符号を付ける」の場合

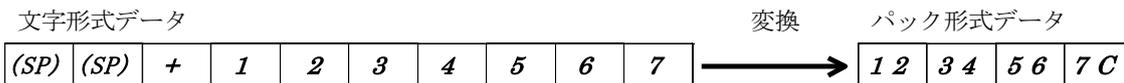


図 9.2.3 符号位置が有効桁の前の変換概要

- 「符号なし」の場合

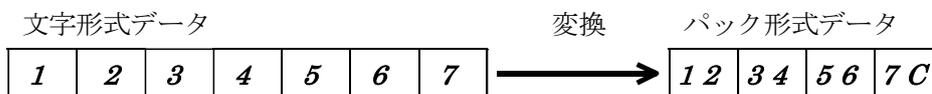


図 9.2.4 符号なしの変換概要

- 「符号なしパックデータ」、「符号なし」の場合

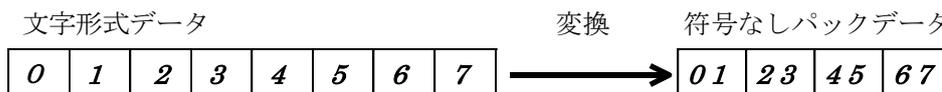


図 9.2.5 符号なしパックデータの変換概要

### (2) 0サプレスの扱い

文字形式データが0サプレスされている（フィールド有効長より前はスペースである）場合は、スペースを0に置き換える必要があるため、編集オプションファイルの「0サプレスの処置」に「0サプレスする (TRAN\_ZERO\_SUPPRESS)」を指定してください。「0サプレスしない (TRAN\_NON\_SUPPRESS)」を指定した場合は、データ不正（エラーコード：760）と判定します。

文字形式データが0サプレスされていない（フィールド有効長より前は0である）場合、編集オプションファイルの「0サプレスの処置」の指定は関係ありません（無視します）。

0サプレスと符号位置の関連については、次のようになります。

- 「0サプレス無し」、「符号位置が先頭」の場合

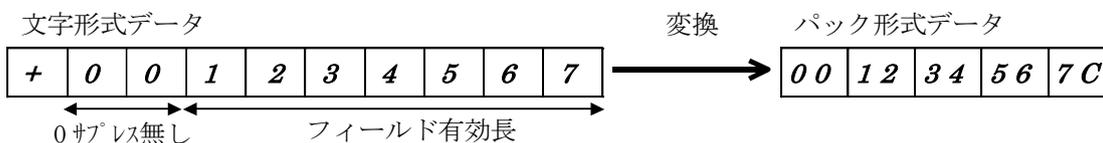


図 9.2.6 0サプレス無し、符号位置が先頭の変換概要

## 9. 数値変換機能について

- ・「0サプレス有り」、「符号位置が終端」の場合

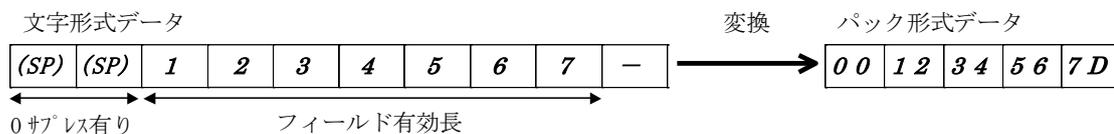


図 9.2.7 0サプレス有り、符号位置が終端の変換概要

### (3) 小数点の扱い

編集オプションファイルに指定した「小数点桁数」に従い、文字形式データの小数点の判定を行います。指定された位置に小数点がない場合は、小数点不正（エラーコード：759）と判定します。小数点桁数が0の場合は、小数点無しと判定します。

- ・「小数点桁数：3」、「0サプレス有り」、「有効桁の前に符号を付ける」の場合



図 9.2.8 小数点桁数3、0サプレス有り、符号位置が有効桁の前の変換概要

- ・「小数点桁数：0」、「0サプレス無し」、「符号なし」の場合

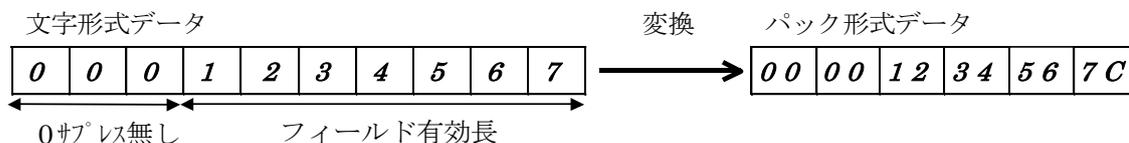


図 9.2.9 小数点桁数0、0サプレス無し、符号なしの変換概要

### (4) オールスペースの扱い

指定されたフィールドがオールスペース（全てX'20'）の場合、編集オプションファイルに指定した「オールスペースの処置」に従います。

「オールスペースはエラーとする (TRAN\_ALLSPACE\_NORMAL)」を指定した場合は、データ不正(エラーコード：760)と判定します。

「オールスペースは0に変換する (TRAN\_ALLSPACE\_ZERO)」を指定した場合は、図 9.2.10に示すように、オール0として変換します。その際、編集オプションファイルに指定した「符号位置」「0サプレスの処置」「小数点桁数」は無視します。

なお、符号は編集オプションファイルの「プラス符号種別」の指定に従います。

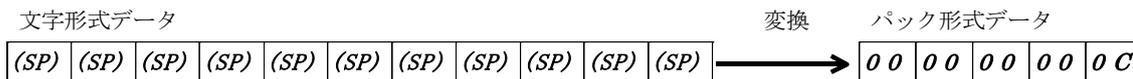


図 9.2.10 オールスペースをオール0に変換する場合の概要

### (5) オール0の扱い

指定されたフィールドがオール0の場合は、通常の数値変換どおりに0に変換します。

## 9. 数値変換機能について

### 9.2.3. ゾーン形式変換 (NUM:OPA7)

ゾーン形式変換は、指定されたフィールドの1バイトの文字形式データを下位4ビットに変換し上位4ビットにX'F'を結合して、1バイトの数値(ゾーン形式)データに変換します。

1バイトの文字形式データにおいて、範囲外(X'30'~ X'39'以外)の場合はデータ不正(エラーコード:760)と判定します。

ゾーン形式変換は、編集オプションファイルの指定に従い変換します。

#### (1) 符号の扱い

編集オプションファイルに指定した「符号位置」に従い、文字形式データの符号の判定を行います。指定された位置にプラス/マイナスがない場合は、符号不正(エラーコード:758)と判定します。また「符号なし(TRAN\_SIGN\_NON)」を指定した場合は、無条件にプラスとします。

なお、「パックデータの符号処置」に「符号なし変換(TRAN\_PACK\_SIGN\_NON)」を指定した場合は、その指示パラメータを無効とし符号付きと判定します。

プラス/マイナスの符号は、ゾーン形式データの最後の1バイトの上位4ビットに符号ビットとして変換します。符号ビットは、編集オプションファイルの指定により、以下のような内容に変換します。

- ・符号がプラスの場合、符号ビットは「プラス符号種別」で指定したパラメータ。
- ・符号がマイナスの場合、符号ビットは「マイナス符号種別」で指定したパラメータ。

#### ・「符号位置が先頭」の場合



図 9.2.11 符号位置が先頭の変換概要

#### ・「符号位置が終端」の場合

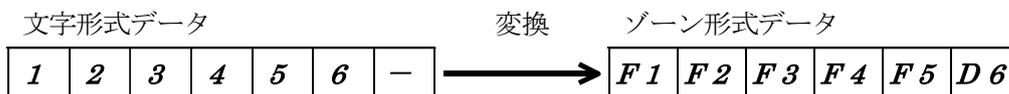


図 9.2.12 符号位置が終端の変換概要

#### ・「有効桁の前に符号を付ける」の場合

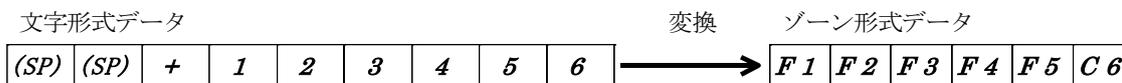


図 9.2.13 符号位置が有効桁の前の変換概要

#### ・「符号位置なし」の場合

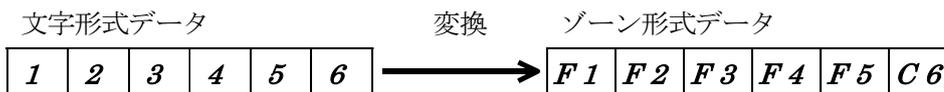


図 9.2.14 符号なしの変換概要

## 9. 数値変換機能について

### (2) 0サプレスの扱い

文字形式データが0サプレスされている（フィールド有効長より前はスペースである）場合は、スペースを0に置き換える必要があるため、編集オプションファイルの「0サプレスの処置」に「0サプレスする（TRAN\_ZERO\_SUPPRESS）」を指定してください。「0サプレスしない（TRAN\_NON\_SUPPRESS）」を指定した場合は、データ不正（エラーコード：760）と判定します。

文字形式データが0サプレスされていない（フィールド有効長より前は0である）場合、編集オプションファイルの「0サプレスの処置」の指定は関係ありません（無視します）。

0サプレスと符号位置の関連については、次のようになります。

- ・「0サプレス無し」、「符号位置が先頭」の場合

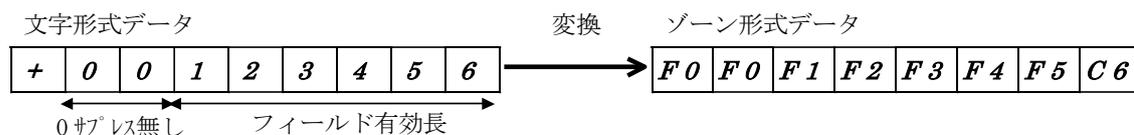


図 9.2.15 0サプレス無し、符号位置が先頭の変換概要

- ・「0サプレス有り」、「符号位置が終端」の場合

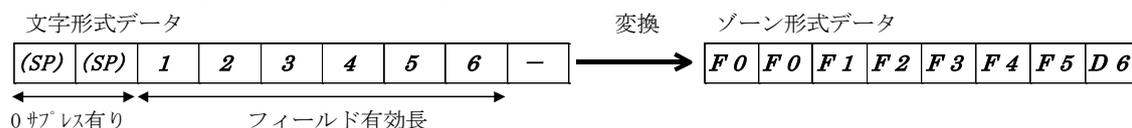


図 9.2.16 0サプレス有り、符号位置が終端の変換概要

### (3) 小数点の扱い

編集オプションファイルに指定した「小数点桁数」に従い、文字形式データの小数点の判定を行います。指定された位置に小数点がない場合は、小数点不正（エラーコード：759）と判定します。小数点桁数が0の場合は、小数点無しと判定します。

- ・「小数点桁数：3」、「0サプレス有り」、「有効桁の前に符号を付ける」の場合

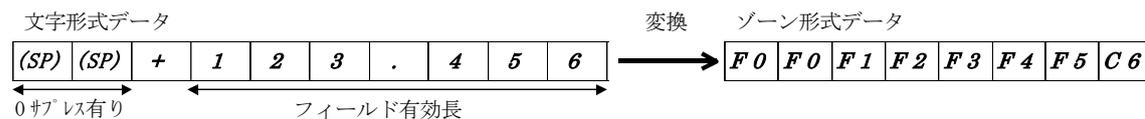


図 9.2.17 小数点桁数3、0サプレス有り、符号位置が有効桁の前の変換概要

- ・「小数点桁数：0」、「0サプレス無し」、「符号なし」の場合

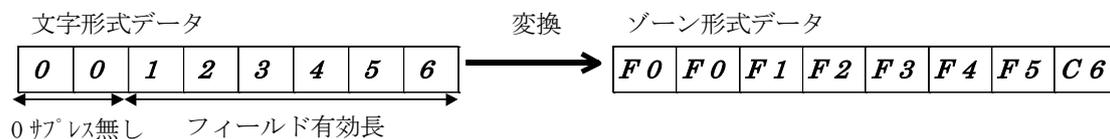


図 9.2.18 小数点桁数0、0サプレス無し、符号なしの変換概要

## 9. 数値変換機能について

### (4) オールスペースの扱い

指定されたフィールドがオールスペース（全てX'20'）の場合、編集オプションファイルに指定した「オールスペースの処置」に従います。

「オールスペースはエラーとする (TRAN\_ALLSPACE\_NORMAL)」を指定した場合は、データ不正(エラーコード：760)と判定します。

「オールスペースは0に変換する (TRAN\_ALLSPACE\_ZERO)」を指定した場合は、図 9.2.19に示すように、オール0として変換します。その際、編集オプションファイルに指定した「符号位置」「0サブレスの処置」「小数点桁数」は無視します。

なお、符号は編集オプションファイルの「プラス符号種別」の指定に従います。

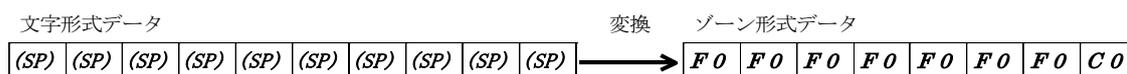


図 9.2.19 オールスペースをオール0に変換する場合の概要

### (5) オール0の扱い

指定されたフィールドがオール0の場合は、通常の数値変換どおりに0に変換します。

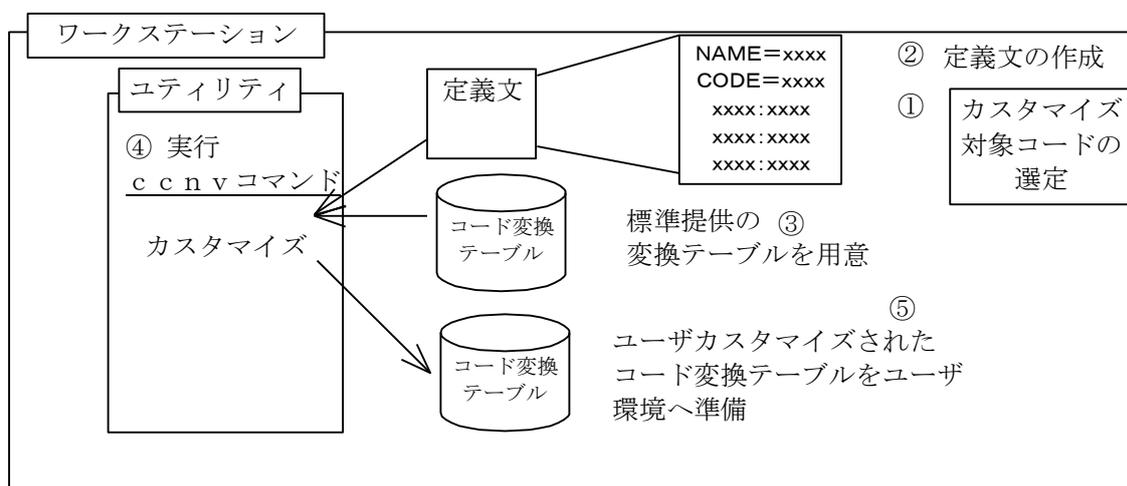
## 10. コード変換テーブルのカスタマイズ

### 10. コード変換テーブルのカスタマイズ

ここでは、コード変換テーブルを実際にカスタマイズする方法について説明します。

#### 10.1. テーブルカスタマイズの概要

ユティリティを使用する際のユーザの処理手順を図 10.1.1を例に説明します。



#### 処理手順

- ① ユーザの使用環境により、カスタマイズをする変換種別、コードを選定します。
- ② ワークステーション上のエディタ(v i 等)により、カスタマイズ方法(定義文)を記述します。
- ③ カスタマイズユティリティを使用するディレクトリに、カスタマイズの元となるコード変換テーブルを用意します。
- ④ ユティリティコマンド(c c n v)により、カスタマイズを実行します。
- ⑤ ユティリティコマンドを実行したディレクトリ(オプションで吐き出し先ディレクトリの変更可)に生成されたカスタマイズ後のコード変換テーブルをユーザの使用環境に移動します。
- ⑥ Code Converter のテーブルのディレクトリ及び、テーブル名称を変更します。

図 10.1.1 ユティリティ概要

## 10. コード変換テーブルのカスタマイズ

### 10.2. ユティリティの機能

ユティリティでは、変換テーブルをカスタマイズするために必要な、いくつかの機能を提供しています。以下にこれらの機能について説明します。

#### (1) テーブルカスタマイズ機能

ユーザが指定するカスタマイズ情報（定義文）に基づいて指定されたコード変換テーブルのカスタマイズする機能です。ただし、標準提供の変換テーブルの修正を行うことはできません。また、外字変換もこの機能を使用します。詳細は、10.6節の実行例を参照してください。

#### (2) テーブル情報出力機能

コード変換テーブルの種別、作成日時、変更履歴等のコード変換テーブル情報を標準出力に出力する機能です。

#### (3) テーブルダンプ出力機能

変換テーブルのダンプ結果をファイルへ出力する機能です。ダンプ出力される内容は、コードについてのみであり、変換テーブルが持つヘッダ情報等の内部的な情報は出力されません。また、出力形式は変換テーブル名（拡張子なし）に拡張子 “. dmp” を付加したファイル名で生成されます。

#### (4) 標準提供テーブル生成機能

標準で提供している変換テーブルの再作成を行う機能です。何らかの手違いまたは、ハードディスクの障害等により標準提供の変換テーブルを変更及び、削除してしまった場合に使用します。

### 10.3. 実行環境の設定

ユティリティでは、実行環境のロケールとしてシフト J I S / EUC を指定することができます。

EUC の場合、ユティリティが出力するメッセージに日本語を表示する方法は、実行環境のロケールに合わせて、環境変数 `CCNV_LANG` に以下の設定を行ってください。

```
CCNV_LANG=EUC
```

なお、ロケール設定の詳細 / 環境変数の設定方法については、使用する OS のマニュアルを参照してください。

## 10. コード変換テーブルのカスタマイズ

### 10.4. 定義文の文法

コード変換テーブルをカスタマイズするために、ユティリティコマンドに指定する定義文の形式を以下に説明します。

#### (1) 定義文指定規則

定義文の記述は、コメント文字以外は半角英数字で指定してください。

##### (a) 定義文の種類

表 10.4.1 にコード変換テーブルのカスタマイズに使用する定義文の種類を示します。

表 10.4.1 定義文の種類

定義文の種類	用途
NAME文	カスタマイズを行うコード変換テーブルの名称を指定します。
TYPE文	カスタマイズを行うコード変換の種類を指定します。
CODE文	カスタマイズする文字コードの種類を指定します。
コードカスタマイズ文	コード変換の対応を変更する文字コードを指定します。
コメント文	コメントを記述します。

##### (b) 定義文開始位置

文の開始位置に決まりはありませんが、1行に1文で120カラム以内に収めてください。

##### (c) 定義文の指定順序

定義文は、図 10.4.1 に順序で定義する必要があります。

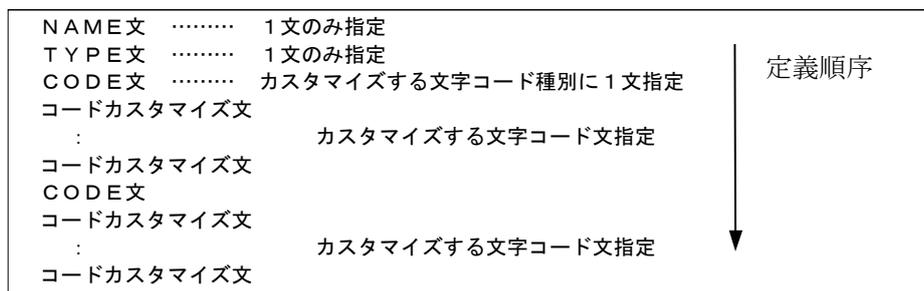


図 10.4.1 定義文の指定順序

#### (2) 定義文詳細

##### (a) NAME文

- 記述形式

NAME = 生成するコード変換テーブル名称

- パラメータの説明

生成するコード変換テーブル名称：

生成するコード変換テーブルの名称を、英字で始まる255桁以内の英数字で指定します。新規にコード変換テーブルを生成せずに、コード変換テーブルの変更のみを行う場合は、‘\*’を指定します。

## 10. コード変換テーブルのカスタマイズ

### (b) TYPE文

- 記述形式

TYPE=コード変換種別
--------------

- パラメータの説明

コード変換種別：

カスタマイズを行う、コード変換の種別を指定します。表 10.4.2に指定できる値を示します。

表 10.4.2 TYPEキーワードに指定できるパラメータの内容

パラメータ	内 容
IBMtoSJIS	IBM 漢字コードからシフト JIS コードへの変換用テーブルをカスタマイズの対象とする
SJIStoIBM	シフト JIS コードから IBM 漢字コードへの変換用テーブルをカスタマイズの対象とする
JEFtoSJIS	JEF 漢字コードからシフト JIS コードへの変換用テーブルをカスタマイズの対象とする
SJIStoJEF	シフト JIS コードから JEF 漢字コードへの変換用テーブルをカスタマイズの対象とする
KS90toSJIS	KEIS83/90 コードからシフト JIS コードへの変換用テーブルをカスタマイズの対象とする
SJIStoKS90	シフト JIS コードから KEIS83/90 コードへの変換用テーブルをカスタマイズの対象とする
KS78toSJIS	KEIS78 コードからシフト JIS コードへの変換用テーブルをカスタマイズの対象とする
SJIStoKS78	シフト JIS コードから KEIS78 コードへの変換用テーブルをカスタマイズの対象とする
IBMtoEUC	IBM 漢字コードから EUC コードへの変換用テーブルをカスタマイズの対象とする
EUCtoIBM	EUC コードから IBM 漢字コードへの変換用テーブルをカスタマイズの対象とする
KS78toEUC	KEIS78 コードから EUC コードへの変換用テーブルをカスタマイズの対象とする
EUCtoKS78	EUC コードから KEIS78 コードへの変換用テーブルをカスタマイズの対象とする
KS90toEUC	KEIS83/90 コードから EUC コードへの変換用テーブルをカスタマイズの対象とする
EUCtoKS90	EUC コードから KEIS83/90 コードへの変換用テーブルをカスタマイズの対象とする
IBMtoUNI	IBM 漢字コードから UNICODE への変換用テーブルをカスタマイズの対象とする
UNItoIBM	UNICODE から IBM 漢字コードへの変換用テーブルをカスタマイズの対象とする
JEFtoUNI	JEF 漢字コードから UNICODE への変換用テーブルをカスタマイズの対象とする
UNItoJEF	UNICODE から JEF 漢字コードへの変換用テーブルをカスタマイズの対象とする
KS90toUNI	KEIS83/90 コードから UNICODE への変換用テーブルをカスタマイズの対象とする
UNItoKS90	UNICODE から KEIS83/90 コードへの変換用テーブルをカスタマイズの対象とする
KS78toUNI	KEIS78 コードから UNICODE への変換用テーブルをカスタマイズの対象とする
UNItoKS78	UNICODE から KEIS78 コードへの変換用テーブルをカスタマイズの対象とする

※：EBCDIC/Kコードから JIS8/EUCコードへの変換を含みます。

### (c) CODE文

- 記述形式

CODE=コード種別
------------

- パラメータの説明

コード種別：

カスタマイズの対象となるコード種別を指定します。省略時は、ユティリティコマンドの-tオプションで渡された変換テーブルの変換種別ごとに仮定されます。表 10.4.3にCODEキーワードに指定できる値を示します。

## 10. コード変換テーブルのカスタマイズ

表 10.4.3 CODEキーワードに指定できるパラメータの内容

コード変換種別	パラメータ	意味
IBMtoSJIS	<u>IBM</u>	IBM 漢字コードをカスタマイズ対象とする
	EBCDIC	EBCDIC コードをカスタマイズ対象とする
	EBCDIK	EBCDIK コードをカスタマイズ対象とする
KS90toSJIS, KS78toSJIS	<u>KEIS</u>	KEIS78 KEIS83/90 コードをカスタマイズ対象とする
	EBCDIC	EBCDIC コードをカスタマイズ対象とする
	EBCDIK	EBCDIK コードをカスタマイズ対象とする
JEFtoSJIS	<u>JEF</u>	JEF コードをカスタマイズ対象とする
	EBCDIC	EBCDIC コードをカスタマイズ対象とする
	EBCDIK	EBCDIK コードをカスタマイズ対象とする
SJIStoIBM, SJIStoJEF, SJIStoKS90, SJIStoKS78	<u>SJIS</u>	シフト JIS コードをカスタマイズ対象とする
	JIS8	JIS8 コードをカスタマイズ対象とする
IBMtoEUC	<u>IBM</u>	IBM 漢字コードをカスタマイズ対象とする
	EBCDIC	EBCDIC コードをカスタマイズ対象とする
	EBCDIK	EBCDIK コードをカスタマイズ対象とする
KS78toEUC KS90toEUC	<u>KEIS</u>	KEIS78 KEIS83/90 コードをカスタマイズ対象とする
	EBCDIC	EBCDIC コードをカスタマイズ対象とする
	EBCDIK	EBCDIK コードをカスタマイズ対象とする
EUCtoIBM EUCtoKS78 EUCtoKS90	<u>EUC1</u>	EUC コードセット 1 をカスタマイズする
	EUC0	EUC コードセット 0 をカスタマイズする
	EUC2	EUC コードセット 2 をカスタマイズする
	EUC3	EUC コードセット 3 をカスタマイズする
IBMtoUNI	<u>IBM</u>	IBM 漢字コードをカスタマイズ対象とする
	EBCDIC	EBCDIC コードをカスタマイズ対象とする
	EBCDIK	EBCDIK コードをカスタマイズ対象とする
KS90toUNI, KS78toUNI	<u>KEIS</u>	KEIS78 KEIS83/90 コードをカスタマイズ対象とする
	EBCDIC	EBCDIC コードをカスタマイズ対象とする
	EBCDIK	EBCDIK コードをカスタマイズ対象とする
JEFtoUNI	<u>JEF</u>	JEF コードをカスタマイズ対象とする
	EBCDIC	EBCDIC コードをカスタマイズ対象とする
	EBCDIK	EBCDIK コードをカスタマイズ対象とする
UNItoIBM, UNItoJEF, UNItoKS90, UNItoKS78	UNI	UNICODE をカスタマイズ対象とする

## 10. コード変換テーブルのカスタマイズ

### (d) コードカスタマイズ文

- 記述形式

変換するコード：変換後コード
----------------

複数指定する場合は、カンマ(‘,’)で区切ります。

各キーワードを定義するカラムは自由ですが、各キーワードとキーワードの間はカンマ(‘,’)または、改行で必ず区切ってください。

- パラメータの説明

変換するコード：変換後コード

変換前コードは、コード変換テーブル中の変換前のコードを16進数で指定します。

変換後コードは、コード変換をすることにより得たいコードを16進数で指定します。

対象コード別に指定可能な範囲を表 10.4.4～表 10.4.9に示します。この範囲以外の値を指定するとエラーとなるので注意してください。また、1バイト変換のカスタマイズを行う際、期待するコードに0xFFを指定しないこと。指定した場合、実際にコード変換した後で得られるコードは、0xFFとならずに1バイト未定義コードとなるので注意すること。

表 10.4.4 シフトJIS⇒メインフレームコード変換での有効コード範囲

対象コード	変換前有効範囲	変換後有効範囲	
JIS8	0x00～0xFF	0x00～0xFE	
SJIS	1バイト目：0x81～0x9F, 0xE0～0xFC 2バイト目：0x40～0x7E, 0x80～0xFC	IBM漢字コード へ変換	1バイト目：0x41～0x7F 2バイト目：0x41～0xFE
		JEF漢字コード へ変換	1バイト目：0x41～0xFE 2バイト目：0xA1～0xFE
		KEISコードへ変換	1バイト目：0x41～0xFE 2バイト目：0xA1～0xFE

表 10.4.5 メインフレームコード⇒シフトJIS変換での有効コード範囲

対象コード	変換前有効範囲	変換後有効範囲
EBCDIC	0x00～0xFF *1	0x00～0xFE
EBCDIK	0x00～0xFF *1	0x00～0xFE
IBM	第1バイト目：0x41～0x7F 第2バイト目：0x41～0xFE	1バイト目：0x81～0x9F, 0xE0～0xFC 2バイト目：0x40～0x7E, 0x80～0xFC
KEIS	第1バイト目：0x41～0xFE 第2バイト目：0xA1～0xFE	
JEF	第1バイト目：0x41～0xFE 第2バイト目：0xA1～0xFE	

\*1：シフトコードは除く

## 10. コード変換テーブルのカスタマイズ

表 10.4.6 EUCコード⇒メインフレームコード変換での有効コード範囲

対象コード	変換前有効範囲	変換後有効範囲	
EUC0	X'00～7F'	X'00～FF'	
EUC1	1バイト目:X'A1～FE' 2バイト目:X'A1～FE'	IBM 漢字コードへ変換	1バイト目:X'41～7F' 2バイト目:X'41～FE'
		JEF コードへ変換	1バイト目:X'41～FE' 2バイト目:X'A1～FE'
		KEIS コードへ変換	1バイト目:X'41～FE' 2バイト目:X'A1～FE'
EUC2	2バイト目:X'A1～DF' *1	X'00～FF'	
EUC3	2バイト目:X'A1～FE' *2 3バイト目:X'A1～FE' *2	IBM 漢字コードへ変換	1バイト目:X'41～7F' 2バイト目:X'41～FE'
		JEF コードへ変換	1バイト目:X'41～FE' 2バイト目:X'A1～FE'
		KEIS コードへ変換	1バイト目:X'41～FE' 2バイト目:X'A1～FE'

\*1：1バイト目のX '8E' は指定しない

\*2：1バイト目のX '8F' は指定しない

表 10.4.7 メインフレームコード⇒EUCコード変換での有効コード範囲

対象コード	変換前有効範囲	変換後有効範囲	
EBCDIC	X'00～FF' *1	EUC コードセット 0	X'00～7F'
		EUC コードセット 2	X'8EA1～8EDF'
EBCDIK	X'00～FF' *1	EUC コードセット 0	X'00～7F'
		EUC コードセット 2	X'8EA1～8EDF'
IBM	1バイト目:X'41～7F' 2バイト目:X'41～FE'	EUC コードセット 1	1バイト目:X'A1～FE' 2バイト目:X'A1～FE'
		EUC コードセット 3	1バイト目:X'8F' 2バイト目:X'A1～FE' 3バイト目:X'A1～FE'
JEF	1バイト目:X'41～FE' 2バイト目:X'A1～FE'	EUC コードセット 1	1バイト目:X'A1～FE' 2バイト目:X'A1～FE'
		EUC コードセット 3	1バイト目:X'8F' 2バイト目:X'A1～FE' 3バイト目:X'A1～FE'
KEIS	1バイト目:X'41～FE' 2バイト目:X'A1～FE'	EUC コードセット 1	1バイト目:X'A1～FE' 2バイト目:X'A1～FE'
		EUC コードセット 3	1バイト目:X'8F' 2バイト目:X'A1～FE' 3バイト目:X'A1～FE'

\*1：シフトコードを除く

## 10. コード変換テーブルのカスタマイズ

表 10.4.8 UN I C O D E ⇒メインフレームコード変換での有効コード範囲

対象コード	変換前有効範囲	変換後有効範囲	
UNICODE	0x0000～0x007F または 0xFF60～0xFF9F	0x00～0xFE	
		IBM 漢字コード <sup>*</sup> へ変換	1 バイト目：0x41～0x7F 2 バイト目：0xA1～0xFE
	JEF 漢字コード <sup>*</sup> へ変換	1 バイト目：0x41～0xFE 2 バイト目：0xA1～0xFE	
	KEIS コード <sup>*</sup> へ変換	1 バイト目：0x41～0xFE 2 バイト目：0xA1～0xFE	

表 10.4.9 メインフレームコード ⇒UN I C O D E 変換での有効コード範囲

対象コード	変換前有効範囲	変換後有効範囲
EBCDIC	0x00～0xFF *1	0x0000～0x007F 又は 0xFF60～0xFF9F
EBCDIK	0x00～0xFF *1	0x0000～0x007F 又は 0xFF60～0xFF9F
IBM	1 バイト目：0x41～0x7F 2 バイト目：0xA1～0xFE	0x0100～0xFF5F 又は 0xFFA0～0xFFE
KEIS	1 バイト目：0x41～0xFE 2 バイト目：0xA1～0xFE	0x0100～0xFF5F 又は 0xFFA0～0xFFE
JEF	1 バイト目：0x41～0xFE 2 バイト目：0xA1～0xFE	0x0100～0xFF5F 又は 0xFFA0～0xFFE

\*1：シフトコードを除く

### (e) コメント文

コメントを記述します。

- 記述形式

# コメント
--------

- パラメータの説明  
‘#’ から改行まではコメントとなります。

## 10. コード変換テーブルのカスタマイズ

### 10.5. コマンドの文法

[Unix 系の場合]

ユーザがコード変換テーブルをカスタマイズするには、`ccnv` コマンドを使用します。以下に `ccnv` コマンドの文法を示します。

#### (1) `ccnv` コマンド

##### (a) 記述形式

<code>ccnv</code> △[-D -d△dir]△[-L -l]△定義文ファイル名
<code>ccnv</code> △[-D -d△dir]△[-P -p]△[-I -i]△-T -t△コード変換テーブル名

△: 半角1バイトのスペースを表す。

##### (b) 定義文ファイル名

コード変換テーブルのカスタマイズ情報を定義した定義文ファイル名を指定します。

ファイル名には、必ず拡張子 “.txt” を付加します。付加しなかった場合は、定義文が無い旨を示すエラーが表示されます。

##### (c) コマンドオプション

###### ●-D|-d△dir

コード変換テーブルの出力先ディレクトリを `dir` に指定します。また、ファイルを出力するこの他のオプションと組み合わせても使用できます。

###### ●-L|-l

結果リストに指定した定義文を合わせて出力します。このオプションを指定すると、エラーとなった行がエラーメッセージの真上に出力されるため、エラーの修正が容易になります。

###### ●-P|-p

生成したコード変換テーブルをデータベースのファイルとしてダンプ出力します。出力先は、コード変換テーブルと同じディレクトリにコード変換テーブルの拡張子を “.dmp” に変更したファイル名で出力されます。カスタマイズ後のテーブルを確認する時などに使用してください。

###### ●-T|-t

処理の対象となるコード変換テーブル名を指定します。(テーブルカスタマイズ実行時は省略可能で、定義文中の変換種別に応じた標準のテーブルが仮定されます。)

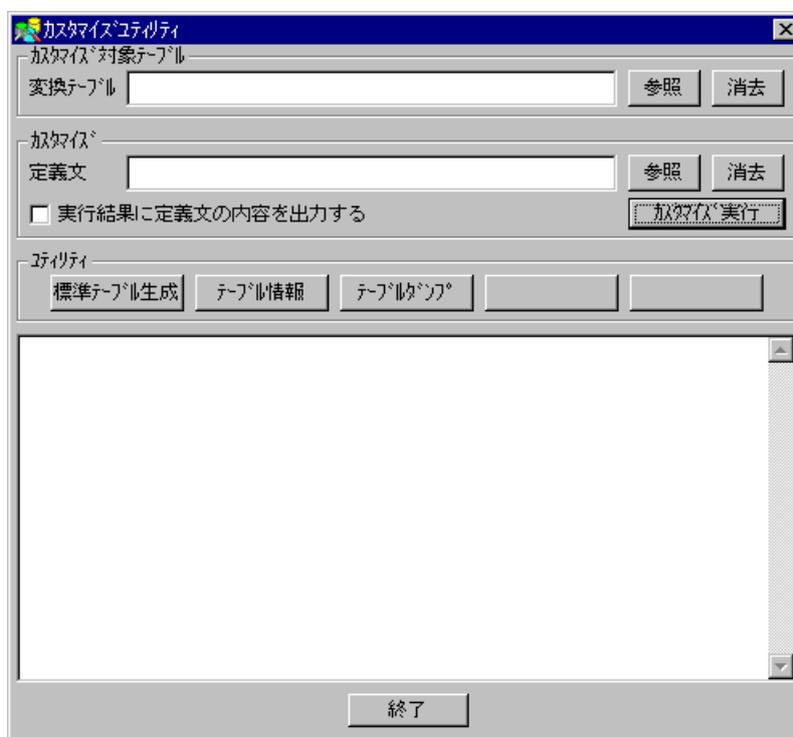
###### ●-I|-i

コード変換テーブルの生成日、変更歴等の情報を標準出力します。

## 10. コード変換テーブルのカスタマイズ

### [Windows 系の場合]

ユーザがコード変換テーブルをカスタマイズするには、カスタマイズユーティリティ(ccnv.exe)を使用します。以下にカスタマイズユーティリティの操作方法を示します。



#### (1) カスタマイズ実行

変換テーブル及び定義文を指定して[カスタマイズ実行]ボタンをクリックすると、コード変換テーブルのカスタマイズが実行されます。

#### (2) テーブル情報出力

[テーブル情報]ボタンをクリックすると、画面下部にコード変換テーブルの生成日、変更歴等の情報を出力します。

#### (3) テーブルダンプ

[テーブルダンプ]ボタンをクリックすると、ワードパッドを起動しコード変換テーブルをキャラクターベースのファイルとしてダンプ出力します。

#### (4) 標準テーブル生成

[標準テーブル生成]ボタンをクリックすると、標準で提供しているテーブルの再生成できます。標準で提供しているファイルを変更してしまうので、通常は使用しないでください。

## 10. コード変換テーブルのカスタマイズ

### 10.6. 実行例

実行例を以下に示します。

#### (1) 通常文字の変換方法をカスタマイズする

EBCDICコードのX'00'をJIS8コードのX'31'に、EBCDIKコードのX'00'をJIS8コードのX'32'に、JEF漢字コードのX'A5A2'をシフトJISコードのX'82A0'に、X'A5A4'をX'82A2'に、それぞれカスタマイズする場合。

#### ●定義文 (ファイル名 = test1.txt)

NAME=TEST1.tbl	# 出力するテーブル名称
TYPE=JEFtoSJIS	# 変換種別
CODE=EBCDIC	# カスタマイズの対象となるコード
00:31	# X'00->X'31
CODE=EBCDIK	# カスタマイズの対象となるコード
00:32	# X'00->X'32
CODE=JEF	# カスタマイズの対象となるコード
A5A2:82A0,	# X'A5A2->X'82A0
A5A4:82A2	# X'A5A4->X'82A2

#### ●実行結果

```
$ ccnv test1.txt
* テーブル情報 *   新規テーブル名 : TEST1.tbl
                   変更テーブル名 : JEF SJIS.tbl

変換種別   テーブル長   有効コード範囲   作成日時           変更日時
EBCDIC->JIS8   000256   0x00-0xFF       1998/03/13, 09:29   0000/00/00, 00:00 0
EBCDIK->JIS8   000256   0x00-0xFF       1998/03/13, 09:29   0000/00/00, 00:00 0
JEF->SJIS      036132   1バイト:0x41-FE 1998/03/13, 09:29   0000/00/00, 00:00 0
                2バイト:0xA1-FE

CCNV-102I コード変換テーブルを生成しました。
CCNV-101I ユティリティは終了しました。終了コード=0。
$
```

## 10. コード変換テーブルのカスタマイズ

### (2) 外字文字を変換する

J E F 漢字コードからシフト J I S コードへの変換で、外字変換を行う場合。メインフレーム上で登録した外字コード : X' 80A1', X' 80A2', X' 80A3' ワークステーション上で登録した外字コード : X' F040', X' F041', X' F042' に設定します。

#### ●定義文 (ファイル名 = test2.txt)

```
NAME=TEST2.tbl          # 出力するテーブル名称
TYPE=JEFtoSJIS         # 変換種別
CODE=JEF                # カスタマイズの対象となるコード
80A1:F040,              # X' 8 0 A 1 ->X' F 0 4 0
80A2:F041,              # X' 8 0 A 2 ->X' F 0 4 1
80A3:F042                # X' 8 0 A 3 ->X' F 0 4 2
```

#### ●実行結果

```
$ ccnv test2.txt
* テーブル情報 *   新規テーブル名 : TEST2.tbl
                   変更テーブル名 : JEFSJIS.tbl

変換種別   テーブル長   有効コード範囲   作成日時           変更日時
EBCDIC->JIS8  000256   0x00-0xFF        1998/03/13, 09:35  0000/00/00, 00:00 0
EBCDIK->JIS8  000256   0x00-0xFF        1998/03/13, 09:35  0000/00/00, 00:00 0
JEF->SJIS     036132   1バイト:0xA1-FE  1998/03/13, 09:35  0000/00/00, 00:00 0
                2バイト:0xA1-FE

CCNV-102I コード変換テーブルを生成しました。
CCNV-101I ユティリティは終了しました。終了コード=0。
$
```

## 10. コード変換テーブルのカスタマイズ

### (3) 変換テーブル情報を出力する

変換テーブルの情報を標準出力に出力する場合。コマンドラインから次のコマンドを入力します。

```
ccnv -i -t JEF SJIS. tbl
```

#### ●実行結果

```
* テーブル情報 * テーブル名 : JEF SJIS. tbl

変換種別      テーブル長  有効コード範囲  作成日時          変更日時
EBCDIC->JIS8  000256    0x00-0xFF      1998/02/18, 14:59  0000/00/00, 00:00 0
EBCDIK->JIS8  000256    0x00-0xFF      1998/02/18, 14:59  0000/00/00, 00:00 0
JEF->SJIS     036132    1バイト:0x41-FE 1998/02/18, 14:59  0000/00/00, 00:00 0
                2バイト:0xA1-FE

CCNV-10WI コード変換テーブル情報を出力しました。
CCNV-101I ユティリティは終了しました。終了コード=0。
```

### (4) 変換テーブルのダンプ情報をファイルへ出力する

変換テーブルのダンプ情報をファイルへ出力する場合。コマンドラインから次のコマンドを入力します。

```
ccnv -p -t JEF SJIS. tbl
```

#### ●実行結果

```
*** EBCDIC コード変換テーブル ***                               [JIS8]
[EBCDIC] +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f 0123456789abcdef
          0x00 00 01 02 03 9c 09 86 7f 97 8d 8e 0b 0c 0d 0e 0f .....
          0x10 10 11 12 13 9d 0a 08 87 18 19 92 8f 1c 1d 1e 1f .....
          0x20 80 81 82 83 84 85 17 1b 88 89 8a 8b 8c 05 06 07 .....
          .

*** EBCDIK コード変換テーブル ***                               [JIS8]
[EBCDIK] +0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +a +b +c +d +e +f 0123456789abcdef
          0x00 00 01 02 03 9c 09 86 7f 97 8d 8e 0b 0c 0d 0e 0f .....
          0x10 10 11 12 13 9d 0a 08 87 18 19 92 8f 1c 1d 1e 1f .....
          .

*** JEF コード変換テーブル ***                                   [SJIS]
[JEF]    +0    +1    +2    +3    +4    +5    +6    +7    0 1 2 3 4 5 6 7
          .
          0xb0a0 ---- 889f 88a0 88a1 88a2 88a3 88a4 88a5   亜唾娃阿哀愛挨
          0xb0a8 88a6 88a7 88a8 88a9 88aa 88ab 88ac 88ad   始逢葵茜穉患握渥
          0xb0b0 88ae 88af 88b0 e9cb 88b2 88b3 88b4 88b5   旭葦芦鱒梓压幹扱
          0xb0b8 88b6 88b7 88b8 88b9 88ba 88bb 88bc 88bd   宛姐虻飴絢綾鮎或
          .
```

## 10. コード変換テーブルのカスタマイズ

### 10.7. CCNVエラーメッセージ

ユティリティを実行した結果、標準出力へ出力されるメッセージを示します。

(a) メッセージの形式

CCNV $n_1n_2n_3$ - $s_1$	
$n_1$	メッセージのプログラム種別を示す数字 1 : ユティリティの出力メッセージ
$n_2n_3$	メッセージ通し番号
$s_1$	メッセージの対象を示す英字 I : 利用者向けメッセージ

(b) 当該メッセージの見方

出力されたメッセージについては、次の順序で説明されています。

① メッセージを次の形式で表します。

CCNV $n_1n_2n_3$ - $s_1$
日本語メッセージテキスト

- CCNV $n_1n_2n_3$ - $s_1$ は、「(a) メッセージの形式」に同じ。
- メッセージテキストは、そのメッセージの内容を表しています。また、出力する条件によって変化する部分については、各部分固有の変数で太字で表示しています。

② メッセージの説明

メッセージの持つ意味、あるいは現象の説明を示します。

③ システムの処置 (S)

メッセージを出力した後の主な処理を示します。

④ 利用者の処置 (P)

メッセージを受け取った利用者が行う処置を示します。

## 10. コード変換テーブルのカスタマイズ

### (c) メッセージ詳細

CCNV101-I

ユティリティは終了しました。終了コード=終了コード。

CCNVは終了しました。終了したときの終了コードを10進数で示します。終了コードの意味は次の通りです。

・終了コード

00：正常終了しました。

08：定義文にエラーが発生したので、その定義文を無視して処理を実行しました。

12：定義文で実行できないエラーが発生したので、処理を中止しました。

終了コードが00以外の場合は、エラーの詳細を示すエラーメッセージがこのメッセージ以前に出力されます。

(S)：処理を終了します。

CCNV102-I

コード変換テーブルを生成しました。

指定された名称で、コード変換テーブルを生成しました。

(S)：処理を続行します。(終了コード00)

CCNV103-I

エラーが発生したため、コード変換テーブルを生成できません。

このメッセージを出力する前に表示したエラーメッセージの理由によって、コード変換テーブルを生成することができません。

(S)：次の定義文の処理を行います。(終了コード12)

(P)：前のエラーメッセージの内容を解析し誤りを直した後、再実行してください。

CCNV104-I

定義文が何も指定されていません。

定義文が何も指定されていないので、ユティリティを実行することができません。

(S)：処理を終了します。(終了コード12)

(P)：定義文を指定して、再実行してください。

CCNV105-I

NAMEキーワードに指定された値が正しくないか、指定されていません。

NAMEキーワードに指定した値が正しくないか、もしくは指定されていません。

(S)：処理を終了します。(終了コード12)

(P)：次の項目を確認後、再実行してください。

- ・ テーブル名称として、英字で始まる255桁以内の英数字で指定しましたか？
- ・ NAMEオペランドを指定し忘れていませんか？

## 10. コード変換テーブルのカスタマイズ

CCNV106-I

TYPEキーワードに指定された値が正しくないか、指定されていません。

TYPEキーワードに指定した値が正しくないか、もしくは指定されていません。

- (S) : 処理を終了します。(終了コード12)
- (P) : 次の項目を確認後、再実行してください。
  - ・ TYPEキーワードにIBMtoSJIS, SJIS to IBM, JEF to SJIS又は, SJIS to JEFのいずれかを指定しましたか?
  - ・ TYPEキーワードを指定し忘れていませんか?

CCNV108-I

変更対象に指定したコード(コード)が、コード変換テーブル中に存在しません。

Code (code) to customize does not exist in the code conversion table.

変更対象に指定したコードが有効範囲外であるため、この文を無視します。

- (S) : 次の定義文を実行します。(終了コード08)
- (P) : 変更対象のコードを見直して再実行してください。

CCNV109-I

コード変換テーブルの入出力エラーが発生しました。

コード変換テーブルの入出力中にエラーが発生しました。

- (S) : 処理を終了します。(終了コード12)
- (P) : 次の項目を確認し、再実行してください。
  - ・ カスタマイズ対象のコード変換テーブルのファイル属性が読み込み可能ですか?
  - ・ カスタマイズ対象のコード変換テーブルが存在していますか?
  - ・ 生成しようとしているテーブル名称と同じ名称のファイルが、同一ディレクトリ内で書き込み禁止で存在していませんか?
  - ・ 生成しようとしているテーブルを出力するディレクトリが、書き込み禁止になっていませんか?

CCNV10A-I

コード変換テーブルを生成するために必要なメモリが不足しました。

コード変換テーブルを生成するために必要なメモリが不足した。

- (S) : 処理を終了。(終了コード12)
- (P) : 使用可能なメモリを拡大して再実行する。

CCNV10B-I

CODEキーワードにコード変換種別と排他的なパラメータが指定されています。

CODEキーワードにコード変換種別と排他的なパラメータが指定されているため、CODEキーワードを無視します。

- (S) : 次の定義文を実行します。(終了コード12)
- (P) : TYPE及び、CODEキーワードを見直して再実行してください。

## 10. コード変換テーブルのカスタマイズ

CCNV10C-I

定義文ファイルが見つかりません。

コマンドのパラメータに指定された定義文ファイルが、ユーティリティを実行したディレクトリ内にありません。

(S) : 処理を終了します。(終了コード12)

(P) : 指定した定義文ファイルが存在するか確認し、なければ定義文ファイルを用意した後再実行してください。

CCNV10D-I

定義文ファイルの入出力エラーが発生しました。

定義文ファイルの入出力中にエラーが起きました。

(S) : 処理を終了します。(終了コード12)

(P) : 指定した定義文ファイルの状態もしくは、システムの状態を確認し、再実行してください。

ユーティリティコマンドの“-T”または、“-t”パラメータで指定されたコード変換テーブルがユーティリティを実行したディレクトリ中に見つかりません。

(S) : 処理を終了します。(終了コード12)

(P) : 指定したコード変換テーブルが存在するか確認し、再実行してください。

CCNV10F-I

CODEキーワードに指定された値が正しくありません。CODE=コードを仮定します。

CODEキーワードに指定された値が正しくありません。

(S) : 次の定義文を実行します。(終了コード08)

(P) : CODEキーワードの値が変換種別に対応したものが確認し、再実行してください。

CCNV10G-I

対象に指定したコード(コード)が、不正です。

対象に指定したコードが正しくありません。

(S) : 次の定義文を実行します。(終了コード08)

(P) : 以下のような項目で対象に指定したコードの値を再確認し、再実行してください。

- ・ 対象に指定したコードはシフトコードではありませんか？

CCNV10H-I

コード変換テーブルは、生成されませんでした。

コード変換テーブルは、何らかのエラーにより生成されませんでした。

(S) : 処理を続行します。(終了コード12)

## 10. コード変換テーブルのカスタマイズ

CCNV10I-I

NAMEキーワードを複数定義することはできません。

NAMEキーワードが複数定義されている。

(S) : 最初のNAMEキーワードを有効とし、処理を続行します。(終了コード08)

(P) : NAMEキーワードを確認し、有効にしたいだけを残して再実行してください。

CCNV10J-I

TYPEキーワードを複数定義することはできません。

Cannot specify more than one TYPE statement.

TYPEキーワードが複数定義されています。

(S) : 最初のTYPEキーワードを有効とし、処理を続行します。(終了コード08)

(P) : TYPEキーワードを確認し、有効にしたいだけを残して再実行してください。

CCNV10K-I

コード変換テーブルの生成で、ファイルのオープンに失敗しました。

コード変換テーブルを生成する際に、ファイルのオープンでエラーになりました。

(S) : 処理を終了します。(終了コード12)

(P) : コード変換テーブルを吐き出すディレクトリの状態を確認し、再実行してください。

CCNV10M-I

標準コード変換テーブルのオープンに失敗しました。

標準のコード変換テーブルを作成する際に、ファイルのオープンでエラーになりました。

(S) : 処理を終了します。(終了コード12)

(P) : 標準コード変換テーブルを吐き出すディレクトリの状態を確認し、再実行してください。

CCNV10N-I

変更対象のコード(コード)が、CODEキーワードで指定したコード種別と一致しません。

変更対象に指定したコードが、CODEキーワードで指定したコードと一致しません。

(変更対象コードに2バイトコードを指定したにもかかわらずCODEキーワードにEBCDICを指定した等)

(S) : 次の定義文を実行します。(終了コード08)

(P) : CODEキーワード及び、変更対象のコードを見直し、再実行してください。

CCNV10O-I

1行に指定できる桁数を超過しました。この行を無視します。

1行中に指定した定義文が120桁を超えました。

(S) : 次の定義文を実行します。(終了コード08)

(P) : 1行中の定義文が120桁以内に収まるように修正し、再実行してください。

## 10. コード変換テーブルのカスタマイズ

CCNV10P-I

行番号行目に認識できない定義があります。その定義を無視します。

エラーメッセージに表示した行中に指定した定義文に、認識できない文字列が現れました。

- (S) : 次の定義文を実行します。(終了コード08)
- (P) : エラー行中の定義文を見直し、再実行してください。

CCNV10Q-I

コード変換テーブルのキャラクタ出力に失敗しました。

コード変換テーブルをキャラクタ出力する際に、ファイルのオープンで失敗しました。

- (S) : 処理を終了します。(終了コード12)
- (P) : キャラクタ出力するディレクトリを調査し、再実行してください。

CCNV10S-I

コード変換テーブルをキャラクタ出力しました。

コード変換テーブルのキャラクタ出力が正常に終了しました。

- (S) : 処理を終了します。(終了コード0)

CCNV10T-I

エラーが発生したため、コード変換テーブルをキャラクタ出力できません。

処理の途中で重大なエラーが発生したため、コード変換テーブルのキャラクタ出力が出来ませんでした。

- (S) : 処理を終了します。(終了コード12)
- (P) : このメッセージ以前に表示されたエラーメッセージで原因を調査し、再実行してください。

CCNV10U-I

標準提供のコード変換テーブルを生成しました。

標準提供のコード変換テーブル出力が正常に終了しました。

- (S) : 処理を終了します。(終了コード0)

CCNV10V-I

エラーが発生したため、標準提供のコード変換テーブルを生成できません。

処理の途中で重大なエラーが発生したため、標準提供のコード変換テーブルを出力することが出来ませんでした。

- (S) : 処理を終了します。(終了コード12)
- (P) : このメッセージ以前に表示されたエラーの原因を調査し、再実行してください。

CCNV10W-I

コード変換テーブル情報を出力しました。

コード変換テーブル情報の出力が正常に終了しました。

- (S) : 処理を終了します。(終了コード0)

## 10. コード変換テーブルのカスタマイズ

CCNV10Y-I

標準提供のコード変換テーブルを変更することはできません。

標準提供のコード変換テーブルは変更できません。

(S) : 処理を終了します。(終了コード12)

(P) : コード変換テーブルのカスタマイズをする際は、ユーザ用として新規にテーブルを生成するようにしてください。

CCNV10Z-I

コード変換テーブルのバージョンが不正です。

Version of the code conversion table is wrong.

コード変換テーブルが正しくありません。

(S) : 処理を終了します。(終了コード12)

(P) : コード変換テーブルが正しいかを確認して再実行してください。

CCNV110-I

コード変換テーブルが不正です。

コード変換テーブルが壊れている。または、コード変換テーブルとしてコード変換テーブル以外のファイルを指定した。

(S) : 処理を終了します。(終了コード12)

(P) : 正しいコード変換テーブルを使用する。コード変換テーブルが何らかの理由により壊れていた場合は、標準提供の変換テーブルを再生成した後実行してください。

CCNV111-I

処理の対象となるコード変換テーブルが指定されていません。

-I (-i) または -P (-p) オプションで、コード変換テーブル情報やコード変換テーブルのダンプ出力を実行しようとしたが、処理の対象となるコード変換テーブルが指定されていない。

(S) : 処理を終了します。(終了コード12)

(P) : -I (-i) または -P (-p) オプションの後に -T (-t) オプションで処理の対象となるコード変換テーブル名称を、同時に指定して実行してください。

CCNV113-I

予期せぬエラーが発生しました。

ユーティリティを実行中に、メモリ破壊等で必要な情報を参照できなくなった。

(S) : 処理を終了します。(終了コード12)

(P) : システムの環境を調査し、再実行をしてください。

CCNV114-I

変更対象のコードと変換後のコード範囲が一致しません。

EUCコードとの変換を行うテーブルをカスタマイズしようとしたが、変更後のコードに指定したコードがEUCコードの範囲に対応していない。

(S) : 処理を終了します。(終了コード12)

(P) : コード範囲をEUCコードセット1または、3に対応させて再実行してください。

## 10. コード変換テーブルのカスタマイズ

CCNV115-I

変更後のコードに指定したコード(コード) は変換先のコード体系にありません。

Converted code (**code**) does not exist in the code conversion table.

変更後のコードに指定したコードが有効範囲外であるため、この文を無視します。

(S) : 次の定義文を実行します。(終了コード08)

(P) : 変更後のコードを見直して再実行してください。

CCNV116-I

変換後のコードとして指定された値(コード) が正しくありません。

変換後のコードが2バイトであるコード体系に対して、コードで示される2バイトより大きい値が指定された。

## 付録A 編集オプションファイル

メインフレーム／サーバ間データ移行支援APP (Code Converter)

### 付録A

#### 編集オプションファイル

## 付録A 編集オプションファイル

```
##### mto #####
[KANJI:OP11]          #...変換モード指示 = KANJI:OP11 の定義ラベルヘッダ
IBMtoSJIS            #...IBM 漢字コードからシフト JIS コードに変換する(カ文字を含む)
./                   #...コード変換テーブルは ./ ディレクトリに存在する
IBMSJIS              #...IBMSJIS.tbl という変換テーブルを使用する
TRAN_DBLSPACE        #...スペース文字は全角スペースコードに変換する
20                   #...半角未定義文字は X' 20' に置換する
8140                 #...全角未定義文字は X' 8140' に置換する
20                   #...半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #...全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_DOUBLE          #...当該フィールドは 2 バイト文字(漢字)のみのフィールドである

[KANJI:OP12]         #...変換モード指示 = KANJI:OP12 の定義ラベルヘッダ
IBMtoSJIS            #...IBM 漢字コードからシフト JIS コードに変換する((英小文字を含
#                   #   む)
./                   #...コード変換テーブルは ./ ディレクトリに存在する
IBMSJIS              #...IBMSJIS.tbl という変換テーブルを使用する
TRAN_EBCDIC|TRAN_START_SINGLE|TRAN_DBLSPACE #
#...当該フィールド内の半角文字コードは EBCDIC コード(英小文字を
#   含む)である。先頭からシフトコードまでは半角に変換する。
#   スペース文字は全角スペースコードに変換する
20                   #...半角未定義文字は X' 20' に置換する
8140                 #...全角未定義文字は X' 8140' に置換する
20                   #...半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #...全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_MIX             #...当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
#                   #   である

[KANJI:OP13]         #...変換モード指示 = KANJI:OP13 の定義ラベルヘッダ
IBMtoSJIS            #...IBM 漢字コードからシフト JIS コードに変換する(英小文字を含
#                   #   まない)
./                   #...コード変換テーブルは ./ ディレクトリに存在する
IBMSJIS              #...IBMSJIS.tbl という変換テーブルを使用する
TRAN_EBCDIK|TRAN_START_SINGLE|TRAN_DBLSPACE #
#...当該フィールド内の半角文字コードは EBCDIK コード(英小文字を含
#   まない)である。先頭からシフトコードまでは半角に変換する。
#   スペース文字は全角スペースコードに変換する
20                   #...半角未定義文字は X' 20' に置換する
8140                 #...全角未定義文字は X' 8140' に置換する
20                   #...半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #...全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_MIX             #...当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
#                   #   である
```

## 付録A 編集オプションファイル

```

[KANJI:OP21]          #…変換モード指示 = KANJI:OP21 の定義ラベルヘッダ
KS78toSJIS           #…KS78 漢コード字からシフトJISコードに変換する
./                   #…コード変換テーブルは ./ テイクトリに存在する
KS78SJIS             #…KS78SJIS.tbl という変換テーブルを使用する
TRAN_DBLSPACE        #…スペース文字は全角スペースコードに変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_DOUBLE          #…当該フィールドは 2 バイト文字(漢字)のみのフィールドである

[KANJI:OP22]          #…変換モード指示 = KANJI:OP22 の定義ラベルヘッダ
KS78toSJIS           #…KS78 漢字コードからシフトJISコードに変換する
./                   #…コード変換テーブルは ./ テイクトリに存在する
KS78SJIS             #…KS78SJIS.tbl という変換テーブルを使用する
TRAN_EBCDIC|TRAN_START_SINGLE|TRAN_DBLSPACE #
#…当該フィールド内の半角文字コードは EBCDIC コード(カを含まな
# い)である。先頭からシフトコードまでは半角に変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_MIX             #…当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
# である

[KANJI:OP23]          #…変換モード指示 = KANJI:OP23 の定義ラベルヘッダ
KS78toSJIS           #…KS78 漢字コードからシフトJISコードに変換する(カ文字を含む)
./                   #…コード変換テーブルは ./ テイクトリに存在する
KS78SJIS             #…KS78SJIS.tbl という変換テーブルを使用する
TRAN_EBCDIK|TRAN_START_SINGLE|TRAN_DBLSPACE #
#…当該フィールド内の半角文字コードは EBCDIK コード(カを含む)
# である。先頭からシフトコードまでは半角に変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_MIX             #…当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
# である

```

## 付録A 編集オプションファイル

```

[KANJI:OP31]          #…変換モード指示 = KANJI:OP31 の定義ラベルヘッダ
KS90toSJIS           #…KS90 漢字コードからシフトJISコードに変換する
./                   #…コード変換テーブルは ./ テイクトリに存在する
KS90SJIS             #…KS90SJIS.tbl という変換テーブルを使用する
TRAN_DBLSPACE        #…スペース文字は全角スペースコードに変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_DOUBLE          #…当該フィルタは 2 バイト文字(漢字)のみのフィルタである

[KANJI:OP32]          #…変換モード指示 = KANJI:OP32 の定義ラベルヘッダ
KS90toSJIS           #…KS90 漢字からコードシフトJISコードに変換する(か文字を含まない)
./                   #…コード変換テーブルは ./ テイクトリに存在する
KS90SJIS             #…KS90SJIS.tbl という変換テーブルを使用する
TRAN_EBCDIC|TRAN_START_SINGLE|TRAN_DBLSPACE #
#…当該フィルタ内の半角文字コードは EBCDIC コード(かを含まない)
#   である。先頭からシフトコードまでは半角に変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_MIX             #…当該フィルタは 2 バイト文字(漢字)、1 バイト文字の混合フィルタ
#   である

[KANJI:OP33]          #…変換モード指示 = KANJI:OP33 の定義ラベルヘッダ
KS90toSJIS           #…KS90 漢字コードからシフトJISコードに変換する(か文字を含む)
./                   #…コード変換テーブルは ./ テイクトリに存在する
KS90SJIS             #…KS90SJIS.tbl という変換テーブルを使用する
TRAN_EBCDIK|TRAN_START_SINGLE|TRAN_DBLSPACE #
#…当該フィルタ内の半角文字コードは EBCDIK コード(かを含む)
#   である。先頭からシフトコードまでは半角に変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_MIX             #…当該フィルタは 2 バイト文字(漢字)、1 バイト文字の混合フィルタ
#   である

```

## 付録A 編集オプションファイル

```

[KANJI:OP41]          #…変換モード指示 = KANJI:OP41 の定義ラベルヘッダ
JEFtoSJIS            #…JEF 漢字コードからシフト JIS コードに変換する(か文字を含む)
./                   #…コード変換テーブルは ./ テレクトリに存在する
JEFSJIS              #…JEFSJIS.tbl という変換テーブルを使用する
TRAN_DBLSPACE        #…スペース文字は全角スペースコードに変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_DOUBLE          #…当該フィールドは 2 バイト文字(漢字)のみのフィールドである

[KANJI:OP42]          #…変換モード指示 = KANJI:OP42 の定義ラベルヘッダ
JEFtoSJIS            #…JEF 漢字コードからシフト JIS コードに変換する(か文字を含ま
# ない)
./                   #…コード変換テーブルは ./ テレクトリに存在する
JEFSJIS              #…JEFSJIS.tbl という変換テーブルを使用する
TRAN_EBCDIC|TRAN_START_SINGLE|TRAN_DBLSPACE #
#…当該フィールド内の半角文字コードは EBCDIC コード(かを含まな
# い)である。先頭からシフトコードまでは半角に変換する。スペ
# ス文字は全角スペースコードに変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_MIX             #…当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
# である

[KANJI:OP43]          #…変換モード指示 = KANJI:OP43 の定義ラベルヘッダ
JEFtoSJIS            #…JEF 漢字コードからシフト JIS コードに変換する(か文字を含む)
./                   #…コード変換テーブルは ./ テレクトリに存在する
JEFSJIS              #…JEFSJIS.tbl という変換テーブルを使用する
TRAN_EBCDIK|TRAN_START_SINGLE|TRAN_DBLSPACE #
#…当該フィールド内の半角文字コードは EBCDIK コード(かを含む)
# である。先頭からシフトコードまでは半角に変換する。スペース
# 文字は全角スペースコードに変換する
20                   #…半角未定義文字は X' 20' に置換する
8140                 #…全角未定義文字は X' 8140' に置換する
20                   #…半角/全角混在時、変換先領域が余れば X' 20' で充填する
8140                 #…全角のみの時、変換先領域が余れば X' 8140' で充填する
TRAN_MIX             #…当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
# である

```

## 付録A 編集オプションファイル

```
[CHAR:OP1]          #…変換モード指示 = CHAR:OP1 の定義ラベルヘッダ
EBCJEFtoJIS8       #…EBCDIC コードから ASCII コードに変換する
./                 #…コード変換テーブルは ./ テイクトリに存在する
EBCASCII           #…EBCASCII.tbl という変換テーブルを使用する
FF                 #…未定義文字は X'FF' に置換する
20                 #…変換先領域が余れば X'20' で充填する

[CHAR:OP14]         #…変換モード指示 = CHAR:OP14 の定義ラベルヘッダ
EBCIBMtoJIS8       #…EBCDIC コード (IBM 系) から JIS8 コードに変換する (英小文字を
# 含む)
./                 #…コード変換テーブルは ./ テイクトリに存在する
IBMSJIS            #…IBMSJIS.tbl という変換テーブルを使用する
20                 #…半角未定義文字は X'20' に置換する
20                 #…変換先領域が余れば X'20' で充填する

[CHAR:OP15]         #…変換モード指示 = CHAR:OP15 の定義ラベルヘッダ
EBKIBMtoJIS8       #…EBCDIK コード (IBM 系) から JIS8 コードに変換する (英小文字を
# 含まない)
./                 #…コード変換テーブルは ./ テイクトリに存在する
IBMSJIS            #…IBMSJIS.tbl という変換テーブルを使用する
20                 #…半角未定義文字は X'20' に置換する
20                 #…変換先領域が余れば X'20' で充填する

[CHAR:OP24]         #…変換モード指示 = CHAR:OP24 の定義ラベルヘッダ
EBCKS78toJIS8      #…EBCKS78 漢字コードから JIS8 コードに変換する (カナ文字を含
# まない)
./                 #…コード変換テーブルは ./ テイクトリに存在する
KS78SJIS           #…KS78SJIS.tbl という変換テーブルを使用する
20                 #…半角未定義文字は X'20' に置換する
20                 #…変換先領域が余れば X'20' で充填する

[CHAR:OP25]         #…変換モード指示 = CHAR:OP25 の定義ラベルヘッダ
EBKKS78toJIS8      #…EBKKS78 漢字からコード JIS8 コードに変換する (カナ文字を含む)
./                 #…コード変換テーブルは ./ テイクトリに存在する
KS78SJIS           #…KS78SJIS.tbl という変換テーブルを使用する
20                 #…半角未定義文字は X'20' に置換する
20                 #…変換先領域が余れば X'20' で充填する
```

## 付録A 編集オプションファイル

```

[CHAR:OP34]          #…変換モード指示 = CHAR:OP34 の定義ラベルヘッダ
EBCKS90toJIS8       #…EBCKS90 漢字コードから JIS8 コードに変換する(かな文字を含
#   まない)
./                  #…コード変換テーブルは ./ テイクトリに存在する
KS90SJIS            #…KS90SJIS.tbl という変換テーブルを使用する
20                  #…半角未定義文字は X' 20' に置換する
20                  #…変換先領域が余れば X' 20' で充填する

[CHAR:OP35]          #…変換モード指示 = CHAR:OP35 の定義ラベルヘッダ
EBKKS90toJIS8       #…EBKKS90 漢字コードから JIS8 コードに変換する(かな文字を含む)
./                  #…コード変換テーブルは ./ テイクトリに存在する
KS90SJIS            #…KS90SJIS.tbl という変換テーブルを使用する
20                  #…半角未定義文字は X' 20' に置換する
20                  #…変換先領域が余れば X' 20' で充填する

[CHAR:OP44]          #…変換モード指示 = CHAR:OP44 の定義ラベルヘッダ
EBCJEFtoJIS8        #…EBCDIC コード (JEF 系) から JIS8 コードに変換する(かな文字を含
#   まない)
./                  #…コード変換テーブルは ./ テイクトリに存在する
JEFSJIS             #…JEFSJIS.tbl という変換テーブルを使用する
20                  #…半角未定義文字は X' 20' に置換する
20                  #…変換先領域が余れば X' 20' で充填する

[CHAR:OP45]          #…変換モード指示 = CHAR:OP45 の定義ラベルヘッダ
EBKJEFtoJIS8        #…EBCDIK コード (JEF 系) から JIS8 コードに変換する(かな文字を含む)
./                  #…コード変換テーブルは ./ テイクトリに存在する
JEFSJIS             #…JEFSJIS.tbl という変換テーブルを使用する
20                  #…半角未定義文字は X' 20' に置換する
20                  #…変換先領域が余れば X' 20' で充填する

[NUM:OPA1]           #…変換モード指示 = NUM:OPA1 の定義ラベルヘッダ
PACKtoCHAR           #…パック形式の数値データから文字形式に変換する
TRAN_SIGN_F         #…符号は先頭に位置付ける
0                   #…小数点の桁数を指定する
TRAN_NON_SUPPRESS   #…先頭からの 0 はそのまま 0 にする
TRAN_SIGN_PLUS_C    #…プラス符号は C である
TRAN_SIGN_MINUS_D   #…マイナス符号は D である
TRAN_ALLSPACE_NORMAL #…入力データのオール 0 はそのまま 0 に変換する

[NUM:OPA2]           #…変換モード指示 = NUM:OPA2 の定義ラベルヘッダ
ZONEtoCHAR           #…ゾーン形式の数値データから文字形式に変換する
TRAN_SIGN_F         #…符号は先頭に位置付ける
0                   #…小数点の桁数を指定する
TRAN_NON_SUPPRESS   #…先頭からの 0 はそのまま 0 にする
TRAN_SIGN_PLUS_C    #…プラス符号は C である
TRAN_SIGN_MINUS_D   #…マイナス符号は D である
TRAN_ALLSPACE_NORMAL #…入力データのオール 0 はそのまま 0 に変換する

```

## 付録A 編集オプションファイル

```
##### otm #####
[KANJI:OP16] #...変換モード指示 = KANJI:OP16 の定義ラベルヘッダ
SJIStoIBM #...SJIS コードから IBM 漢字コードに変換する(かな文字を含む)
./ #...コード変換テーブルは ./ ディレクトリに存在する
SJISIBM #...SJISIBM.tbl という変換テーブルを使用する
TRAN_DBLSPACE #...スペース文字は全角スペースコードに変換する
40 #...半角未定義文字は X' 40' に置換する
4040 #...全角未定義文字は X' 4040' に置換する
40 #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
4040 #...全角のみの時、変換先領域が余れば X' 4040' で充填する
TRAN_DOUBLE #...当該フィールドは 2 バイト文字(漢字)のみのフィールドである

[KANJI:OP17] #...変換モード指示 = KANJI:OP17 の定義ラベルヘッダ
SJIStoIBM #...SJIS コードから IBM 漢字コードに変換する(英小文字を含む)
./ #...コード変換テーブルは ./ ディレクトリに存在する
SJISIBM #...SJISIBM.tbl という変換テーブルを使用する
TRAN_EBCDIC|TRAN_START_SINGLE|TRAN_DBLSPACE #
#...当該フィールド内の半角文字コードは EBCDIC コード(英小文字を
# 含む)である。先頭からシフトコードまでは全角に変換する。
# スペース文字は全角スペースコードに変換する
40 #...半角未定義文字は X' 40' に置換する
4040 #...全角未定義文字は X' 4040' に置換する
40 #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
4040 #...全角のみの時、変換先領域が余れば X' 4040' で充填する
TRAN_MIX #...当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
# である

[KANJI:OP18] #...変換モード指示 = KANJI:OP18 の定義ラベルヘッダ
SJIStoIBM #...SJIS コードから IBM 漢字コードに変換する(英小文字を含ま
# ない)
./ #...コード変換テーブルは ./ ディレクトリに存在する
SJISIBM #...SJISIBM.tbl という変換テーブルを使用する
TRAN_EBCDIK|TRAN_START_SINGLE|TRAN_DBLSPACE #
#...当該フィールド内の半角文字コードは EBCDIK コード(英小文字を含
# まない)である。先頭からシフトコードまでは全角に変換する。
# スペース文字は全角スペースコードに変換する
40 #...半角未定義文字は X' 40' に置換する
4040 #...全角未定義文字は X' 4040' に置換する
40 #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
4040 #...全角のみの時、変換先領域が余れば X' 4040' で充填する
TRAN_MIX #...当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
# である
```

## 付録A 編集オプションファイル

```

[KANJI:OP26]          #...変換モード指示 = KANJI:OP26 の定義ラベルヘッダ
SJIStoKS78          #...SJIS コードから KS78 漢字コードに変換する(か文字を含む)
./                  #...コード変換テーブルは ./ ディレクトリに存在する
SJISKS78            #...SJISKS78.tbl という変換テーブルを使用する
TRAN_DBLSPACE       #...スペース文字は全角スペースコードに変換する
40                  #...半角未定義文字は X' 40' に置換する
A1A1                #...全角未定義文字は X' A1A1' に置換する
40                  #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
A1A1                #...全角のみの時、変換先領域が余れば X' A1A1' で充填する
TRAN_DOUBLE         #...当該フィールドは 2 バイト文字(漢字)のみのフィールドである

[KANJI:OP27]          #...変換モード指示 = KANJI:OP27 の定義ラベルヘッダ
SJIStoKS78          #...SJIS コードから KS78 漢字コードに変換する(か文字を含まな
#   い)
./                  #...コード変換テーブルは ./ ディレクトリに存在する
SJISKS78            #...SJISKS78.tbl という変換テーブルを使用する
TRAN_EBCDIC|TRAN_START_SINGLE|TRAN_DBLSPACE #
#...当該フィールド内の半角文字コードは EBCDIC コード(かを含まな
#   い)である。先頭からシフトコードまでは全角に変換する
40                  #...半角未定義文字は X' 40' に置換する
A1A1                #...全角未定義文字は X' A1A1' に置換する
40                  #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
A1A1                #...全角のみの時、変換先領域が余れば X' A1A1' で充填する
TRAN_MIX            #...当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
#   である

[KANJI:OP28]          #...変換モード指示 = KANJI:OP28 の定義ラベルヘッダ
SJIStoKS78          #...SJIS コードから KS78 コード漢字に変換する(か文字を含む)
./                  #...コード変換テーブルは ./ ディレクトリに存在する
SJISKS78            #...SJISKS78.tbl という変換テーブルを使用する
TRAN_EBCDIK|TRAN_START_SINGLE|TRAN_DBLSPACE #
#...当該フィールド内の半角文字コードは EBCDIK コード(かを含む)
#   である。先頭からシフトコードまでは全角に変換する
40                  #...半角未定義文字は X' 40' に置換する
A1A1                #...全角未定義文字は X' A1A1' に置換する
40                  #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
A1A1                #...全角のみの時、変換先領域が余れば X' A1A1' で充填する
TRAN_MIX            #...当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
#   である

```

## 付録A 編集オプションファイル

```

[KANJI:OP36]          #…変換モード指示 = KANJI:OP36 の定義ラベルヘッダ
SJIStoKS90           #…SJIS コードから KS90 漢字コードに変換する(か文字を含む)
./                   #…コード変換テーブルは ./ ディレクトリに存在する
SJISKS90             #…SJISKS90.tbl という変換テーブルを使用する
TRAN_DBLSPACE        #…スペース文字は全角スペースコードに変換する
40                   #…半角未定義文字は X' 40' に置換する
A1A1                 #…全角未定義文字は X' A1A1' に置換する
40                   #…半角/全角混在時、変換先領域が余れば X' 40' で充填する
A1A1                 #…全角のみの時、変換先領域が余れば X' A1A1' で充填する
TRAN_DOUBLE          #…当該フィールドは 2 バイト文字(漢字)のみのフィールドである

[KANJI:OP37]          #…変換モード指示 = KANJI:OP37 の定義ラベルヘッダ
SJIStoKS90           #…SJIS コードから KS90 漢字コードに変換する(か文字を含まな
# い)
./                   #…コード変換テーブルは ./ ディレクトリに存在する
SJISKS90             #…SJISKS90.tbl という変換テーブルを使用する
TRAN_EBCDIC|TRAN_START_SINGLE|TRAN_DBLSPACE #
#…当該フィールド内の半角文字コードは EBCDIC コード(かを含まな
# い)である。先頭からシフトコードまでは全角に変換する
40                   #…半角未定義文字は X' 40' に置換する
A1A1                 #…全角未定義文字は X' A1A1' に置換する
40                   #…半角/全角混在時、変換先領域が余れば X' 40' で充填する
A1A1                 #…全角のみの時、変換先領域が余れば X' A1A1' で充填する
TRAN_MIX             #…当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
# である

[KANJI:OP38]          #…変換モード指示 = KANJI:OP38 の定義ラベルヘッダ
SJISKS90             #…SJIS コードから KS90 漢字コードに変換する(か文字を含む)
./                   #…コード変換テーブルは ./ ディレクトリに存在する
SJISKS90             #…SJISKS90.tbl という変換テーブルを使用する
TRAN_EBCDIK|TRAN_START_SINGLE|TRAN_DBLSPACE #
#…当該フィールド内の半角文字コードは EBCDIK コード(かを含む)
# である。先頭からシフトコードまでは全角に変換する
40                   #…半角未定義文字は X' 40' に置換する
A1A1                 #…全角未定義文字は X' A1A1' に置換する
40                   #…半角/全角混在時、変換先領域が余れば X' 40' で充填する
A1A1                 #…全角のみの時、変換先領域が余れば X' A1A1' で充填する
TRAN_MIX             #…当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
# である

```

## 付録A 編集オプションファイル

```

[KANJI:OP46]          #...変換モード指示 = KANJI:OP46 の定義ラベルヘッダ
SJIStoJEF            #...SJIS コードから JEF 漢字コードに変換する(かな文字を含む)
./                   #...コード変換テーブルは ./ ディレクトリに存在する
SJISJEF              #...SJISJEF.tbl という変換テーブルを使用する
TRAN_SGLSPACE        #...スペース文字は半角スペースコードに変換する
40                   #...半角未定義文字は X' 40' に置換する
4040                 #...全角未定義文字は X' 4040' に置換する
40                   #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
4040                 #...全角のみの時、変換先領域が余れば X' 4040' で充填する
TRAN_DOUBLE          #...当該フィールドは 2 バイト文字(漢字)のみのフィールドである

[KANJI:OP47]          #...変換モード指示 = KANJI:OP47 の定義ラベルヘッダ
SJIStoJEF            #...SJIS コードから JEF 漢字コードに変換する(かな文字を含まない)
./                   #...コード変換テーブルは ./ ディレクトリに存在する
SJISJEF              #...SJISJEF.tbl という変換テーブルを使用する
TRAN_EBCDIC|TRAN_START_SINGLE|TRAN_SGLSPACE|TRAN_SHIFT_28 #
#...当該フィールド内の半角文字コードは EBCDIC コード(かなを含まない)
#...である。先頭からシフトコードまでは全角に変換する。スペース
#...文字は半角スペースコードに変換する
40                   #...半角未定義文字は X' 40' に置換する
4040                 #...全角未定義文字は X' 4040' に置換する
40                   #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
4040                 #...全角のみの時、変換先領域が余れば X' 4040' で充填する
TRAN_MIX             #...当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
#...である

[KANJI:OP48]          #...変換モード指示 = KANJI:OP48 の定義ラベルヘッダ
SJIStoJEF            #...SJIS コードから JEF 漢字コードに変換する(かな文字を含む)
./                   #...コード変換テーブルは ./ ディレクトリに存在する
SJISJEF              #...SJISJEF.tbl という変換テーブルを使用する
TRAN_EBCDIK|TRAN_START_SINGLE|TRAN_SGLSPACE|TRAN_SHIFT_28 #
#...当該フィールド内の半角文字コードは EBCDIK コード(かなを含む)
#...である。先頭からシフトコードまでは全角に変換する。スペース
#...文字は半角スペースコードに変換する
40                   #...半角未定義文字は X' 40' に置換する
4040                 #...全角未定義文字は X' 4040' に置換する
40                   #...半角/全角混在時、変換先領域が余れば X' 40' で充填する
4040                 #...全角のみの時、変換先領域が余れば X' 4040' で充填する
TRAN_MIX             #...当該フィールドは 2 バイト文字(漢字)、1 バイト文字の混合フィールド
#...である

```

## 付録A 編集オプションファイル

```
[CHAR:OP6]          #…変換モード指示 = CHAR:OP6 の定義ラベルヘッダ
JIS8toEBCJEF        #…ASCII コードから EBCDIC コードに変換する
./                  #…コード変換テーブルは ./ テイクトリに存在する
ASCIIIEBC           #…ASCIIIEBC.tbl という変換テーブルを使用する
FF                  #…半角未定義文字は X' FF' に置換する
40                  #…半角/全角混在時、変換先領域が余れば X' 40' で充填する

[CHAR:OP19]         #…変換モード指示 = CHAR:OP19 の定義ラベルヘッダ
JIS8toEBCIBM        #…JIS8 コードから EBCIBM 漢字コードに変換する(英小文字を含
# む)
./                  #…コード変換テーブルは ./ テイクトリに存在する
SJISIBM             #…SJISIBM.tbl という変換テーブルを使用する
40                  #…半角未定義文字は X' 40' に置換する
40                  #…変換先領域が余れば X' 40' で充填する

[CHAR:OP1A]         #…変換モード指示 = CHAR:OP1A の定義ラベルヘッダ
JIS8toEBKIBM        #…JIS8 コードから EBKIBM 漢字コードに変換する(英小文字を含
# まない)
./                  #…コード変換テーブルは ./ テイクトリに存在する
SJISIBM             #…SJISIBM.tbl という変換テーブルを使用する
40                  #…半角未定義文字は X' 40' に置換する
40                  #…変換先領域が余れば X' 40' で充填する

[CHAR:OP29]         #…変換モード指示 = CHAR:OP29 の定義ラベルヘッダ
JIS8toEBCKS78       #…JIS8 コードから EBCKS78 漢字コードに変換する(カナ文字を含
# まない)
./                  #…コード変換テーブルは ./ テイクトリに存在する
SJISKS78            #…SJISKS78.tbl という変換テーブルを使用する
40                  #…半角未定義文字は X' 40' に置換する
40                  #…変換先領域が余れば X' 40' で充填する

[CHAR:OP2A]         #…変換モード指示 = CHAR:OP2A の定義ラベルヘッダ
JIS8toEBKKS78       #…JIS8 コードから EBKKS78 漢字コードに変換する(カナ文字を含
# む)
./                  #…コード変換テーブルは ./ テイクトリに存在する
SJISKS78            #…SJISKS78.tbl という変換テーブルを使用する
40                  #…半角未定義文字は X' 40' に置換する
40                  #…変換先領域が余れば X' 40' で充填する

[CHAR:OP39]         #…変換モード指示 = CHAR:OP39 の定義ラベルヘッダ
JIS8toEBCKS90       #…JIS8 コードから EBCKS90 漢字コードに変換する(カナ文字を含
# まない)
./                  #…コード変換テーブルは ./ テイクトリに存在する
SJISKS90            #…SJISKS90.tbl という変換テーブルを使用する
40                  #…半角未定義文字は X' 40' に置換する
40                  #…変換先領域が余れば X' 40' で充填する
```

## 付録A 編集オプションファイル

```

[CHAR:OP3A]          #…変換モード指示 = CHAR:OP3A の定義ラベルヘッダ
JIS8toEBKKS90      #…JIS8 コードから EBKKS90 漢字コードに変換する(かな文字を含む)
./                  #…コード変換テーブルは ./ テイクトリに存在する
SJISKS90           #…SJISKS90.tbl という変換テーブルを使用する
40                 #…半角未定義文字は X' 40' に置換する
40                 #…変換先領域が余れば X' 40' で充填する

[CHAR:OP49]         #…変換モード指示 = CHAR:OP49 の定義ラベルヘッダ
JIS8toEBCJEF       #…JIS8 コードから EBCJEF 漢字コードに変換する(かな文字を含ま
# ない)
./                  #…コード変換テーブルは ./ テイクトリに存在する
SJISJEF            #…SJISJEF.tbl という変換テーブルを使用する
40                 #…半角未定義文字は X' 40' に置換する
40                 #…変換先領域が余れば X' 40' で充填する

[CHAR:OP4A]         #…変換モード指示 = CHAR:OP4A の定義ラベルヘッダ
JIS8toEBKJEF       #…JIS8 コードから EBKJEF 漢字コードに変換する(かな文字を含ま
# ない)
./                  #…コード変換テーブルは ./ テイクトリに存在する
SJISJEF            #…SJISJEF.tbl という変換テーブルを使用する
40                 #…半角未定義文字は X' 40' に置換する
40                 #…変換先領域が余れば X' 40' で充填する

[NUM:OPA6]          #…変換モード指示 = NUM:OPA6 の定義ラベルヘッダ
CHARtoPACK         #…文字形式データをパック形式の数値データに変換する
TRAN_SIGN_F        #…入力データの符号が先頭にある
0                  #…入力データの小数点の桁数を指定する
TRAN_NON_SUPPRESS  #…先頭からの0はそのまま0にする
TRAN_SIGN_PLUS_C   #…プラス符号をCにする
TRAN_SIGN_MINUS_D  #…マイナス符号をDにする
TRAN_ALLSPACE_NORMAL #…入力データのオールスペースはエラーとする

[NUM:OPA7]          #…変換モード指示 = NUM:OPA7 の定義ラベルヘッダ
CHARtoZONE         #…文字形式データをゾーン形式の数値データに変換する
TRAN_SIGN_F        #…入力データの符号が先頭にある
0                  #…入力データの小数点の桁数を指定する
TRAN_NON_SUPPRESS  #…先頭からの0はそのまま0にする
TRAN_SIGN_PLUS_C   #…プラス符号をCにする
TRAN_SIGN_MINUS_D  #…マイナス符号をDにする
TRAN_ALLSPACE_NORMAL #…入力データのオールスペースはエラーとする

[]                  #…終了記述子

```