
Micro Focus Enterprise Developer チュートリアル

メインフレーム COBOL, PL/I 開発： クラシック データファイル ツールの活用

1. 目的

メインフレームの COBOL, PL/I アプリケーション開発と運用においては、EBCDIC 文字コードのデータや開発言語特有のデータ型で格納されたデータが使用されています。オープン環境へアプリケーションの移行を検討する際、そのまま EBCDIC 文字コードのデータを使用するか、または変換作業を行い ANSI 文字コードのデータを使用するかは移行設計時に検討される事柄となりますが、変換作業を行い ANSI 文字コードのデータを利用する場合はどのようにデータ変換を行うのか、また、開発言語特有のデータ型は保持できるのかなどが一般的に懸念される事項です。

他オープン系システムとのインターフェイスを考慮して ANSI 文字コード運用へ移行する事例が大多数ですが、実行環境製品の Enterprise Server では EBCDIC 文字コードのデータを使用できる運用モードを、開発環境製品の Enterprise Developer では EBCDIC 文字コードのデータを使用した開発が可能な機能を提供しています。このため開発環境製品にはこれら懸念点の解決をサポートするクラシック データファイル ツールがデフォルトで搭載されています。本チュートリアルでは、このクラシック データファイル ツールの使用方法の習得を目的としています。

当ツールで可能なことは以下の通りです。

- 1) データとレコードレイアウトを関連付けた項目ごとのメンテナンス
- 2) ANSI 文字コードデータのメンテナンス
- 3) EBCDIC 文字コードデータのメンテナンス
- 4) COMP-3 などの開発言語特有のデータ型を数値でメンテナンス
- 5) G 項目など、ダブルバイト文字のメンテナンス
注意) MIX 項目に EBCDIC 文字コードの漢字データがある場合は前後にシフトコードが入ります。
- 6) 16 進数を使用したメンテナンス
- 7) EBCDIC 文字コードデータから ANSI 文字コードデータへの変換
- 8) ANSI 文字コードデータから EBCDIC 文字コードデータへの変換
など

2. 前提

- 本チュートリアルで使用したマシン OS : Windows 10 Enterprise
- 本チュートリアルで使用した製品 : Micro Focus Enterprise Developer 6.0

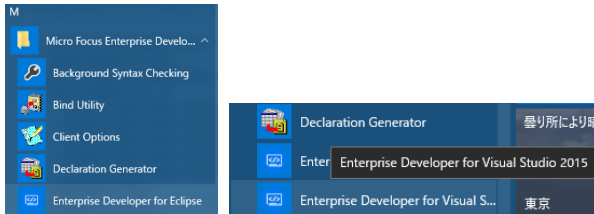
3. チュートリアル手順の概要

1. デバッグ用ファイルの準備
2. クラシック データファイル ツールの起動
3. レコードレイアウトの作成
4. 条件付きレコードレイアウトの作成
5. データのメンテナンス
6. データの文字コード変換
7. 半角カタカナの扱い

3.1 デバッグ用ファイルの準備

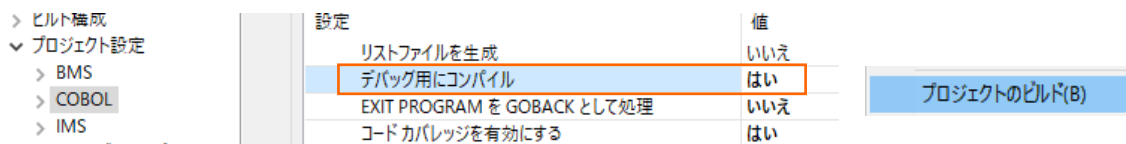
クラシックデータファイルツールはデータとレコードレイアウトを関連付けて、項目ごとのメンテナンスが可能な機能を提供しています。そのためにレコードレイアウトが必要になり、このレコードレイアウトはソースコンパイル時に生成されるデバッグ用ファイルを利用することになります。まずはこのデバッグ用ファイルを生成します。

- 1) Enterprise Developer for Eclipse または for Visual Studio を起動します。

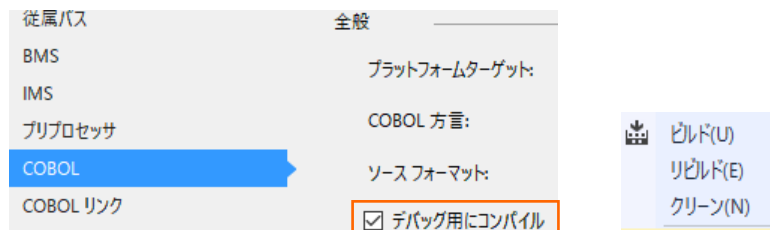


- 2) プロジェクトのコンパイル指定で [デバッグ用にコンパイル] をチェックして、プロジェクトのビルドを行います。

[Eclipse]

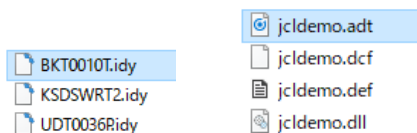


[Visual Studio]



- 3) 指定した出力フォルダに、COBOL は XXXXX.idy が PL/I は XXXXX.adt (XXXXX=プログラム名) ファイルが生成されます。このファイルはデバッグ時に使用するものですが、ソースに書かれた構造体を利用するためにレコードレイアウトを作成する際にも必要になります。

例)



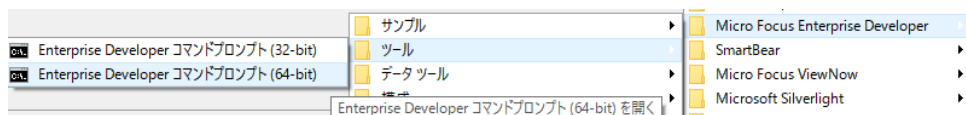
3.2 クラシック データファイル ツールの起動

1) クラシック データファイル ツールを起動します。この際、環境変数 MFICODESET の値によりデータ表示や文字コード変換結果が異なりますので、目的に沿った設定を行ってください。当チュートリアルは 939 を使用しています。

① 半角カタカナを含むデータをメンテナンスする場合

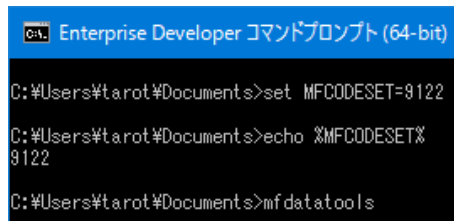
環境変数に MFICODESET=9122 を指定します。英小文字データは使用できないため英大文字に置き換えられます。

Enterprise Developer コマンドプロンプトを起動して環境変数を指定後、クラシック データファイル ツールをコマンドで起動します。



環境変数コマンド) set MFICODESET=9122

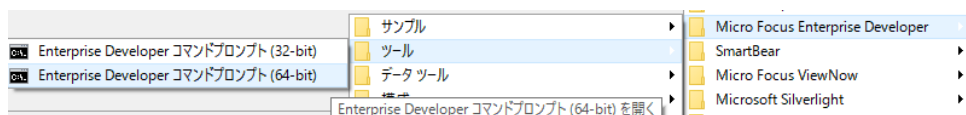
ツール起動コマンド) mfdatatools



② 英大小文字を含むデータをメンテナンスする場合

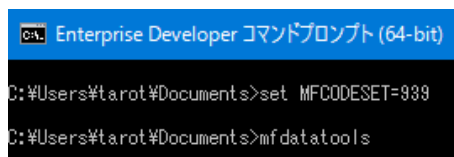
環境変数に MFICODESET=939 を指定します。半角カタカナデータは使用しないことが基本ですが、ANSI データに半角カタカナが含まれるデータ変換時は EBCDIC 文字コードの空き番号を使用しています。

Enterprise Developer コマンドプロンプトを起動して環境変数を指定後、クラシック データファイル ツールをコマンドから起動します。



環境変数コマンド) set MFICODESET=939

ツール起動コマンド) mfdatatools

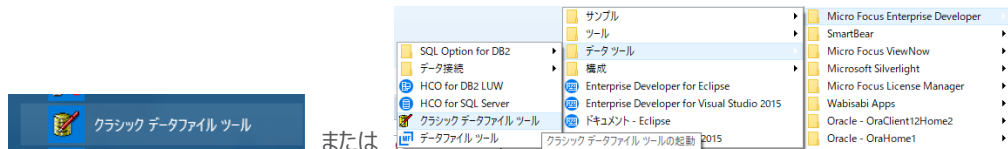


③ 半角カタカナを含むデータをメンテナンスする場合（デフォルト）

環境変数に何も指定しない、もしくは MFCODESET=930 を指定します。英小文字データは使用しないことが基本ですが、ANSI データに英小文字が含まれるデータ変換時は EBCDIC 文字コードの空き番号を使用しています。

Enterprise Developer コマンドプロンプトを起動して環境変数を指定後、クラシック データファイル ツールをコマンドから起動、もしくはメニューから起動します。

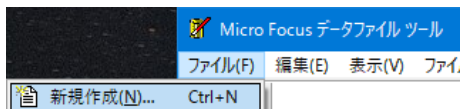
【環境変数を設定しない場合：メニューから起動】



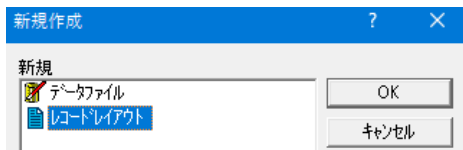
3.3 レコードレイアウトの作成

起動されたクラシック データファイル ツール ウィンドウで操作を行います。

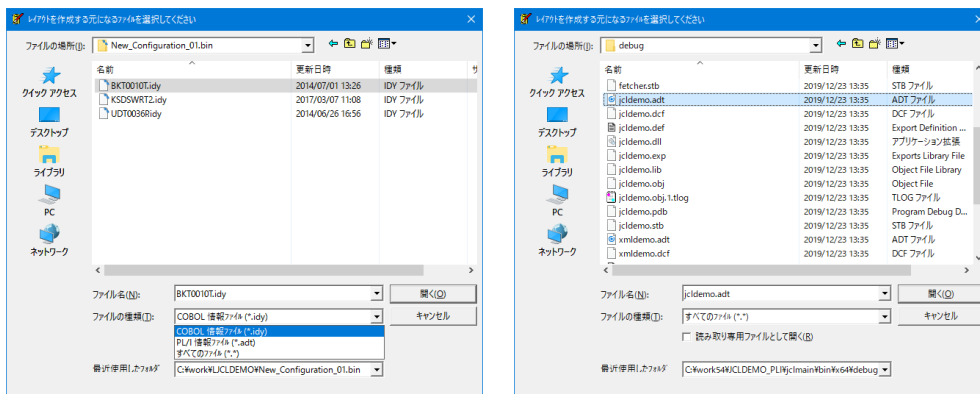
- 1) [ファイル] プロダクションメニュー > [新規作成] を選択します。



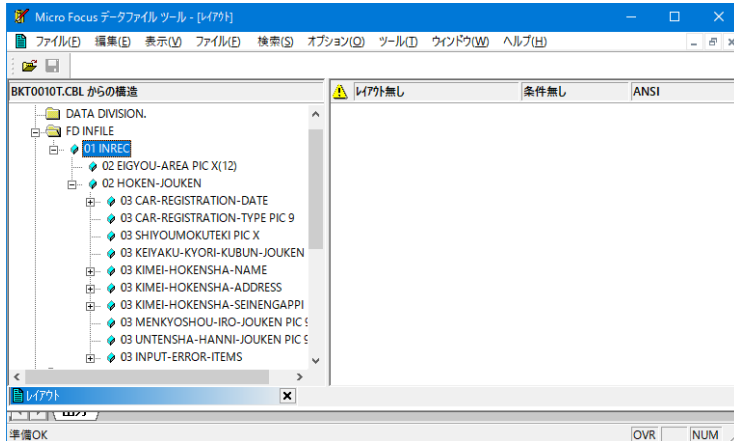
- 2) [レコードレイアウト] 選択して [OK] ボタンをクリックします。



- 3) [ファイルの種類] へ COBOL の場合は [COBOL 情報ファイル(*.idy)] を、PL/I の場合は [すべてのファイル(*.*)] を選択し、[ファイル] へ前項で生成された .idy または .adt ファイルを選択して [開く] ボタンをクリックします。



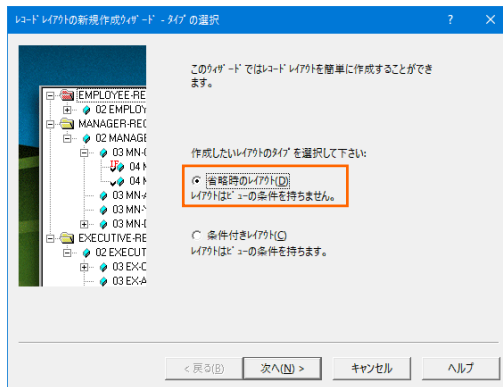
4) レイアウト ウィンドウの左側ツリービューに使用されている構造体が表示されます。



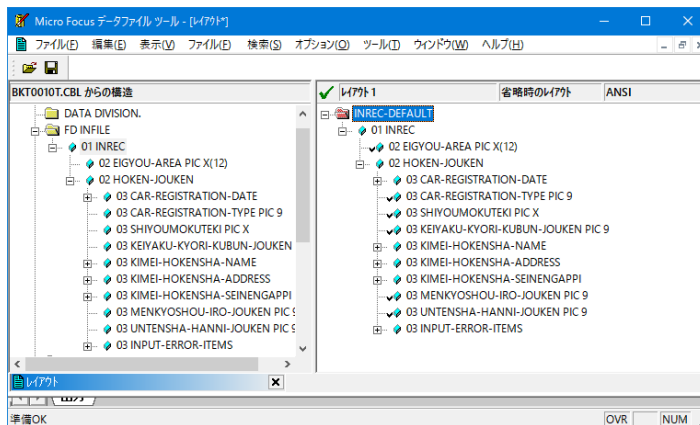
5) 左側ツリービューからレコードレイアウトとして利用したい構造体を右側フレームヘドラッグ&ドロップすると [レコードレイアウトの新規作成ウィザード] ウィンドウが表示されます。

① [タイプの選択] には [省略時のレイアウト] を選択して [次へ] ボタンをクリックします。

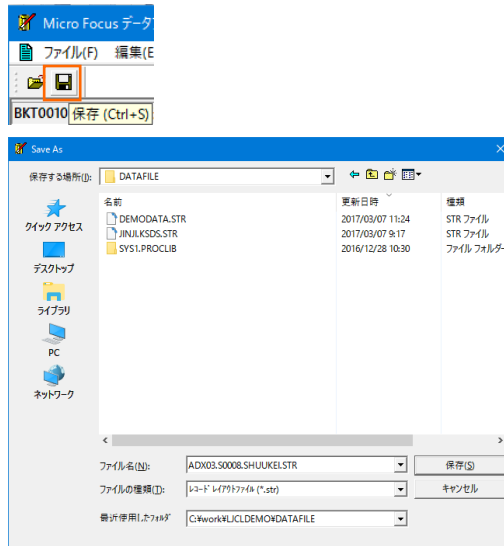
補足) [条件付きレイアウト] を選択すると構造体に含まれるキーを判断して複数レイアウトへ振り分けが可能になります。



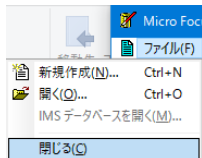
② 次のウィンドウで [完了] ボタンをクリックすると、右側フレームに選択した構造体が表示されます。



- 6) 保存アイコンをクリックして、レコードレイアウトを適用したいデータファイルが存在するパスへ、データファイルと同名を指定して [保存] ボタンをクリックします。レコードレイアウトの拡張子は .str です。



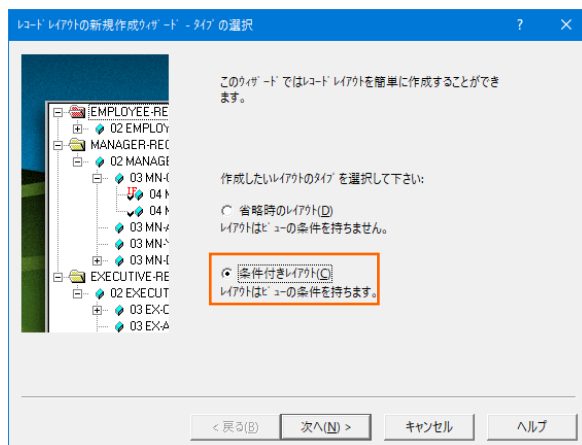
- 7) [ファイル] プロダウメニュー > [閉じる] を選択してレコードレイアウトウィンドウを終了します。



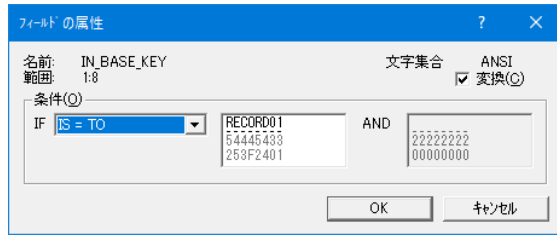
3.4 条件付きレコードレイアウトの作成

ある項目の値を条件としたマルチレイアウトを定義することができます。

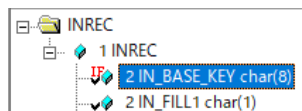
- 1) 前項と同様に [タイプの選択] まで進み [条件付きレイアウト] を選択して [次へ] ボタンをクリック後 [完了] ボタンをクリックします。



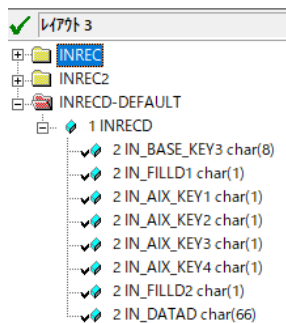
- 2) 右側フレームに展開されたレイアウト内の条件項目を右クリック後 [属性] を選択し、条件と条件値を設定して [OK] ボタンをクリックします。



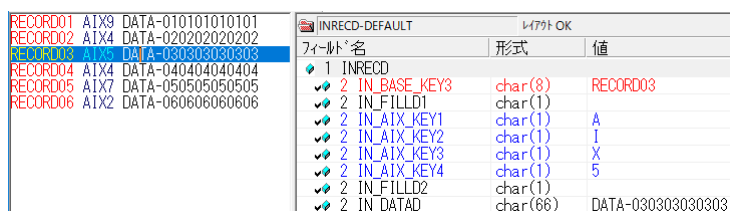
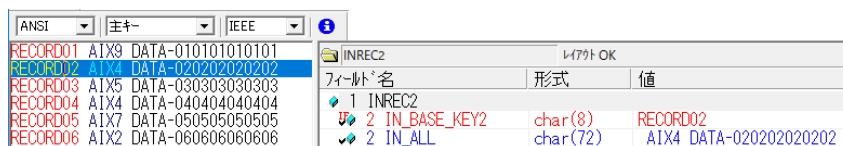
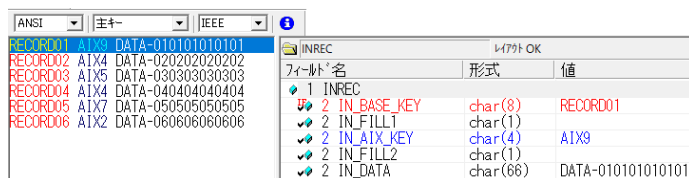
条件項目に [IF] と表示されます。



- 3) 異なるレイアウトを左側ツリービューから右側フレームヘドラッグ & ドロップして上記と同様に条件を設定していきます。
- 4) 前項と同様に [省略時のレイアウト] を選択するとデフォルトレイアウトとして設定され、下記のように表示されます。



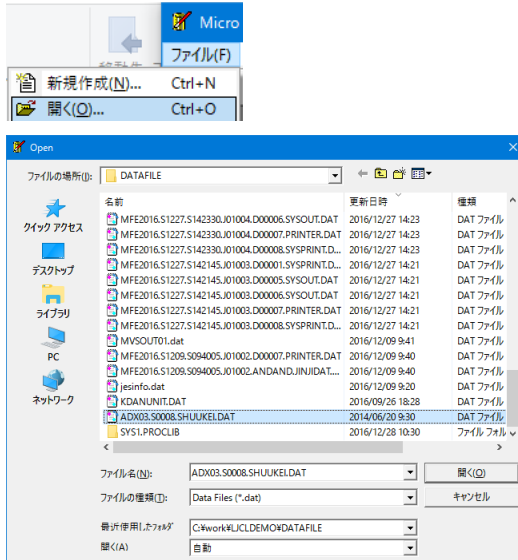
- 5) データファイルを開くと条件に沿ったレイアウトが適用されてデータが表示されます。



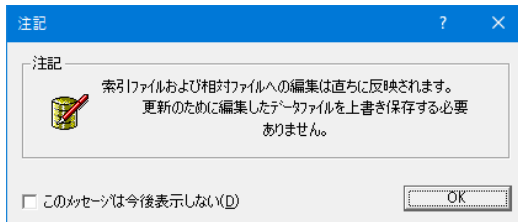
3.5 データのメンテナンス

データファイルを開いてメンテナンスを行います。

- 1) [ファイル] プロダウメニュー > [開く] を選択後、対象データファイルを指定して [開く] ボタンをクリックします。

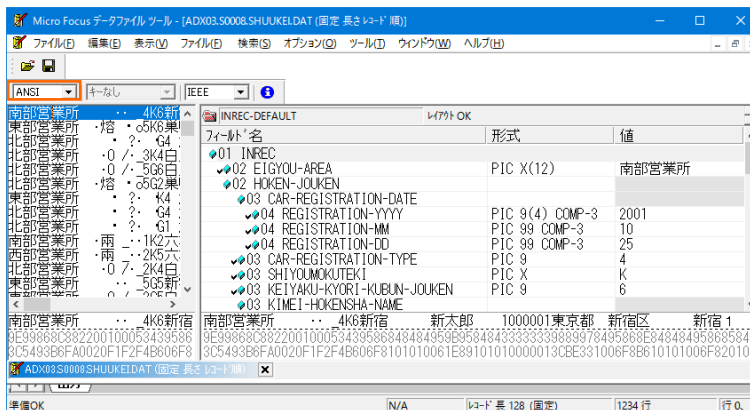


- 2) 入力値が直ちに反映される旨の注意喚起メッセージが表示されますので [OK] ボタンをクリックします。

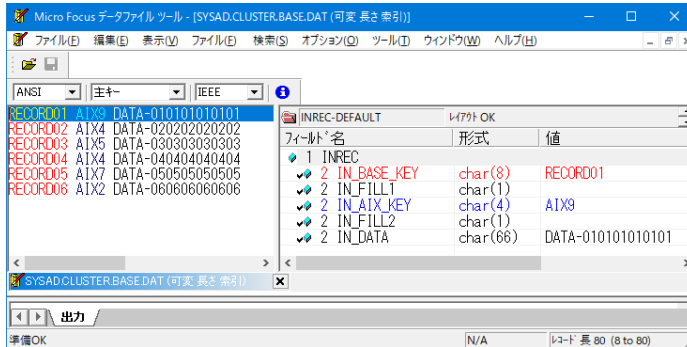


- 3) データファイル名とレコードレイアウト名が同名の場合は、データファイルと関連付けてレコードレイアウトが表示されます。このデータは ANSI 文字コードのため [ANSI] の選択により可視化しています。

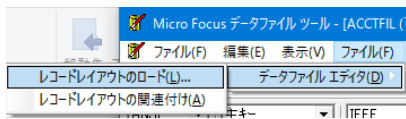
[COBOL ANSI 文字コードデータ]



【PL/I ANSI 文字コードデータ】



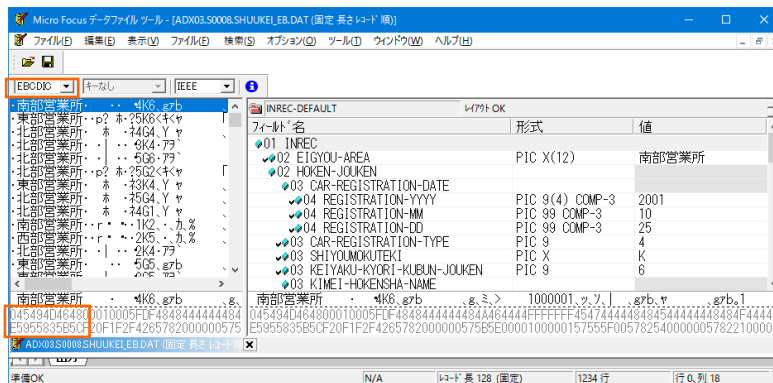
レコードレイアウトがデータファイル名と異なる場合は [ファイル] プルダウンメニュー > [データファイル エディタ] > [レコードレイアウトのロード] を選択して、レコードレイアウトを適用してください。



補足)

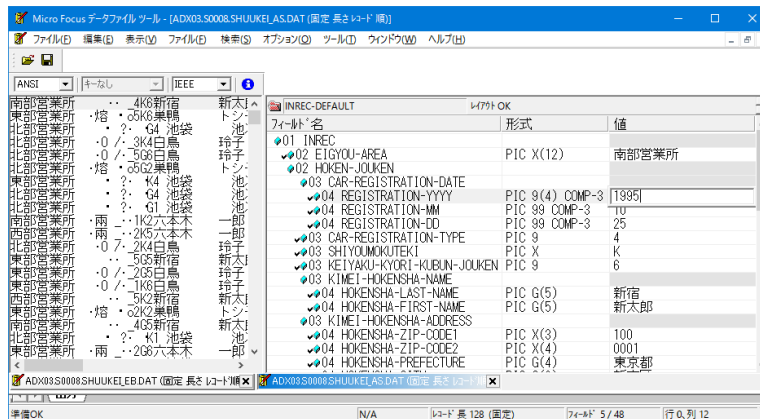
EBCDIC 文字コードデータの場合は [EBCDIC] へ切り替えることで可視化されます。次の例は EBCDIC 文字コードデータのため、MIX 項目の漢字前後にシフトコードが確認できます。

【EBCDIC 文字コードデータ】

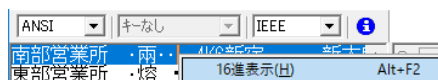


- 4) 先頭データをクリック後、右フレーム内で項目を更新してみます。

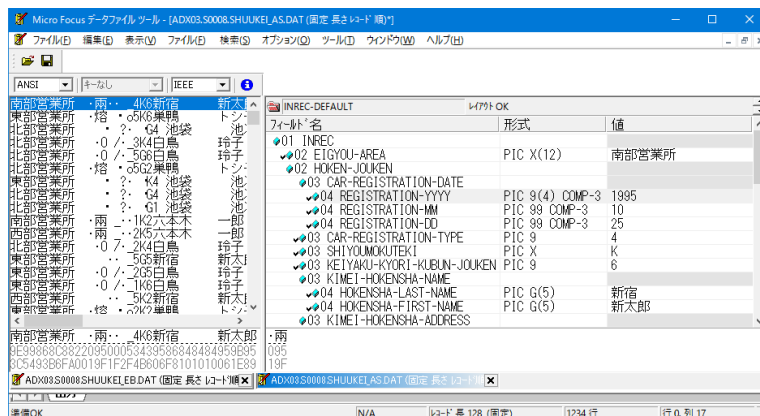
REGISTRATION-YYYY 項目を 2001 から 1995 へ変更してみます。



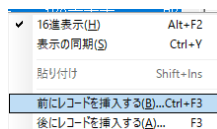
- 5) 変更後、変更レコードを選択した状態で右クリックし、[16進表示] を選択してヘキサコードを確認してみます。



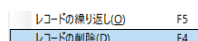
ウィンドウ下にヘキサコードが表示され、ANSI 文字コードが適用されていることがわかります。直接ヘキサコードを入力することも可能です。また、右フレームには選択されている項目のヘキサコードのみが表示されています。



- 6) 新規データを作成する際には、右クリック後 [前にレコードを挿入する] もしくは [後にレコードを挿入する] を選択します。索引順ファイルの場合は [索引レコードの挿入] が選択肢として表示されます。



- 7) 既存データを削除する際には、右クリック後 [レコードの削除] を選択します。



- 8) EBCDIC 文字コードデータに関しても同様に追加、更新、削除が可能です。ヘキサコード表示を行うと EBCDIC 文字コードが格納されていることがわかります。

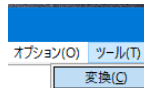


データ追加時の適用コードは、現在どちらの文字コードで表示されているかに依存します。[EBCDIC] が選択されている場合は新規レコードのデータ文字コードは EBCDIC になります。

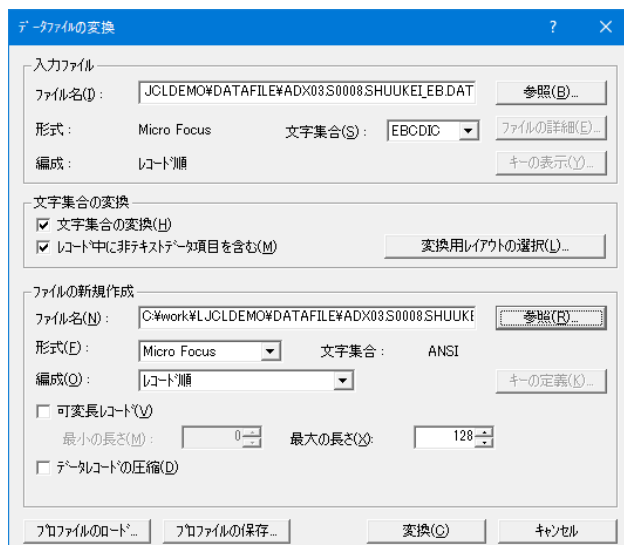
3.6 データのコード変換

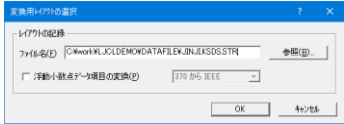
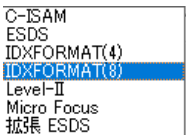
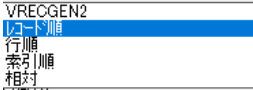
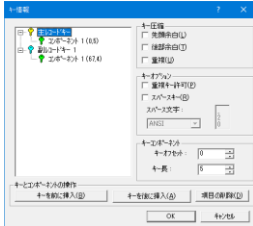
EBCDIC 文字コードデータから ANSI 文字コードデータへ、またその逆コードへの変換を行います。変換対象データファイルを編集集中は実行できませんので、その場合はファイルを閉じてください。

- 1) [ツール] プロダウンメニュー > [変換] を選択して [データファイルの変換] ウィンドウを表示します。



- 2) まずは EBCDIC 文字コードデータを ANSI 文字コードデータへ変換するよう指定します。



項目名	説明
入力ファイル：ファイル名	EBCDIC 文字コードデータのフルパスを指定します。
入力ファイル：文字集合	EBCDIC を指定します。
文字集合の変換：文字集合の変換	変換を行うためチェックをオンにします。
文字集合の変換：レコード中に非テキストデータ項目を含む	COMP-3 などの非テキスト項目を含むレコードの場合はチェックをオンにします。
文字集合の変換：変換用レイアウトの選択	非テキスト項目を含むレコードの場合はボタンをクリックして適用するレコードレイアウトを指定します。 
ファイルの新規作成：ファイル名	コード変換後のファイル名を指定します。
ファイルの新規作成：形式	ファイル形式を指定します。 
ファイルの新規作成：文字集合	変換後の文字コードです。ANSI が表示されます。
ファイルの新規作成：編成	レコード編成を指定します。 
ファイルの新規作成：キーの定義	索引順などキーを保持する場合はボタンをクリックして指定します。 
ファイルの新規作成：可変長レコード	可変長レコードの場合はチェックをオンにして最小と最大のレコード長を指定します。
ファイルの新規作成：データレコードの圧縮	レコードを圧縮する場合はチェックをオンにします。

- 3) 変換に指定した前項の入力値を保存するため [プロファイルの保存] ボタンをクリックします。保存したプロファイルは次回の変換時やコマンドベースで変換を実行する際に利用できます。拡張子は .pro です。



注意) [プロファイルのロード] ボタンで適用後には、変換設定内容が正当であるかを必ず確認してください。

- 4) 保存したプロファイルをエクスプローラーからテキストベースのエディタで表示し、内容を確認してみてください。

```

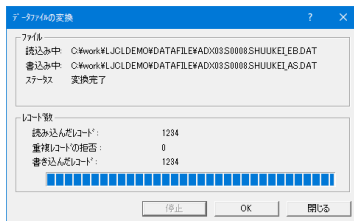
DataTools Profile Version V10.2011.01
-----
Batch-Process: Convert-File=
*****
  (Type) (Single)
  * Get Items: Convert-File None
*****
Start-File: C:\Work\JLJCLDEMOW\DATAFILE\ADX03S0008\SHUUKI_EB.DAT
Edit-Mode: Quick=
  * Get Items: Quick/Full
-----
Input-File: C:\Work\JLJCLDEMOW\DATAFILE\ADX03S0008\SHUUKI_EB.DAT
Format:
  Micro-Focus
  * Get Items: Micro-Focus DDFORMAT(4)
  * Bitwise D-124#
  * LEV=11 E20#
  * DDFORMAT(8) Extended-E300
  * Visio#
-----
Organization: Sequential
  * Get Items: Sequential Line-Sequential
  * Indexed Mainframe-Report-AM3
  * Quickio Mainframe-Report-quick
  * VREGEN PIC-Print
  * Block=VRe#
-----
Record-Format: Fixed=
  * Get Items: Fixed/Variable
-----
Character-Set: EBCDIC=
  * Get Items: ASCII/EBCDIC
-----
Floating-Point: IEEE=
  * Get Items: IEEE/F70
-----
Compression: Off=
  * Get Items: On/Off
-----
Min-Rec-Length: 0=
Max-Rec-Length: 128=
  
```

- 5) [データファイルの変換] ウィンドウへ戻り、[変換] ボタンをクリックして変換を実行します。



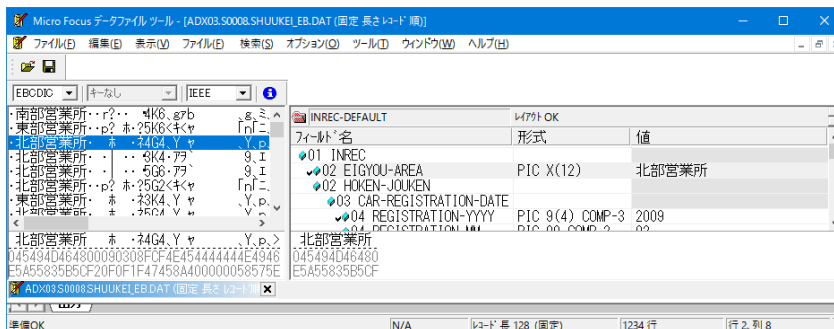
注意) 変換対象ファイルが編集集中は実行できません。

- 6) [データファイルの変換] ウィンドウへ戻り、[変換] ボタンをクリックして変換を実行します。終了時には対象レコード件数が表示されますので、確認後 [OK] ボタンをクリックします。

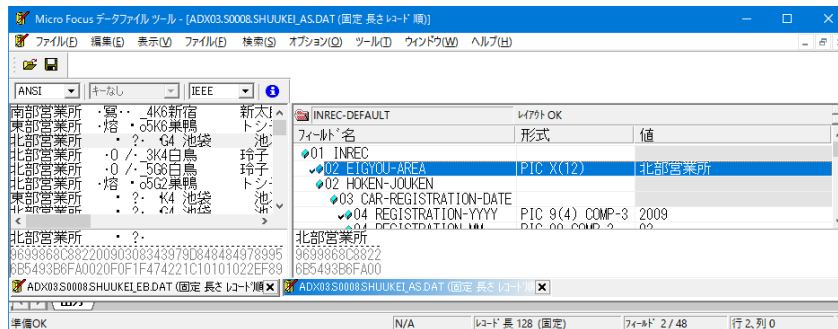


- 7) [データファイルの変換] ウィンドウを閉じて、[ファイル] プロダウンメニュー > [開く] を選択後、変換後のファイルを指定して [開く] ボタンをクリックします。変換前のファイルも同時に開きます。

[変換前：EBCDIC 文字コードデータ]



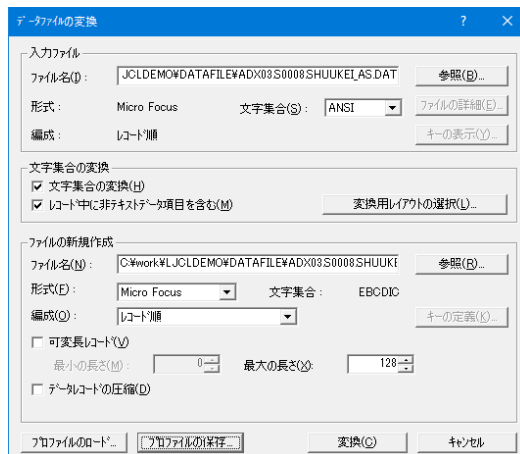
【変換後：ANSI 文字コードデータ】



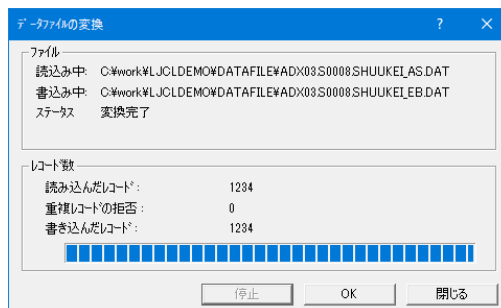
漢字の前後にあるシフトコードが削除されて最後に 2 バイト追加されています。

注意) MIX 項目が連続する構造体の場合、変換時のシフトコードの扱いにより桁ずれが起こる可能性があります。

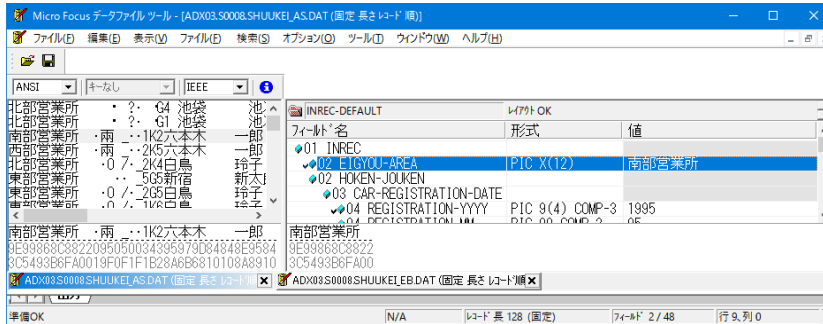
- 8) 次に ANSI 文字コードデータを EBCDIC 文字コードデータへ変換するよう指定します。



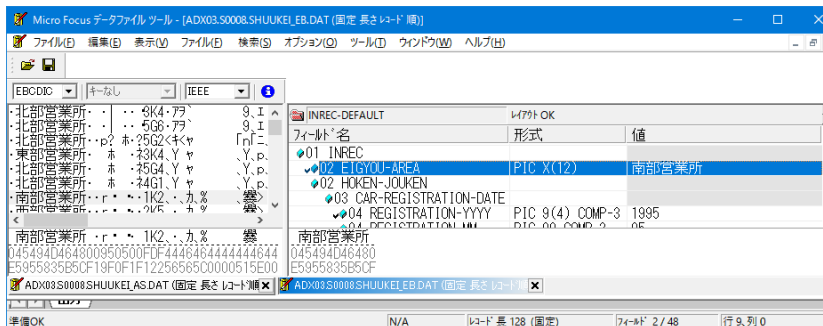
前項と同様に変換します。



【変換前：ANSI 文字コードデータ】



【変換後：EBCDIC 文字コードデータ】



漢字の前後にシフトコードが挿入されています。

変換時に作成したプロファイルはコマンドを利用した変換にも使用可能ですので、DFCONV コマンドを使用した複数ファイルを一括変換する際などにご活用ください。

3.7 データファイルの再文字コード変換について

EBCDIC 文字コードデータから ANSI 文字コードデータへ変換を行い、同じデータファイルを ANSI 文字コードデータから EBCDIC 文字コードデータへ再度変換した場合、元のヘキサコードへ戻らない場合があります。

例えば MFCODESET=9122 を指定したマッピング表を下記コマンドで出力し確認すると、X'FF' は EBCDIC から ANSI へ変換時に X'20' となりますが、ANSI から EBCDIC への変換時には X'40' となり、元の値には戻りません。

コマンド) codecomp /p mapfile.a2e mapfile.e2a

【EBCDIC→ANSI】

F6	36
F7	37
F8	38
F9	39
FA	20
FB	20
FC	20
FD	20
FE	20
FF	20

【ANSI→EBCDIC】

1E	1E
1F	1F
20	40
21	5A
22	7F
23	7B
24	E0

独自のマッピング表を使用したい場合には、上記コマンドで出力したマッピング表を更新し、独自の MFCODESET 番号を設定して使用してください。詳細に関しては製品マニュアルをご参照ください。