

Enterprise Developer チュートリアル

メインフレーム COBOL, PL/I 開発：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。変換を行う際の一般的な懸念事項として、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、開発言語特有のデータ型は保持できるのかなどが挙げられます。

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多く見受けられますが、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ご提供しています。このため開発環境製品にはこれらの懸念点をサポートするデータファイル 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) JES 機能を持つ Enterprise Server インスタンスのカタログファイルと連携したメンテナンス
- 2) 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
- 3) ANSI 文字コードデータのメンテナンス
- 4) EBCDIC 文字コードデータのメンテナンス
- 5) 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
- 6) 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。

- 7) 16 進数を使用したメンテナンス
- 8) EBCDIC 文字コードデータから ANSI 文字コードデータへの変換
- 9) 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 11 Pro
- 本チュートリアルで使用した製品：Enterprise Developer 9.0

3. チュートリアル手順の概要

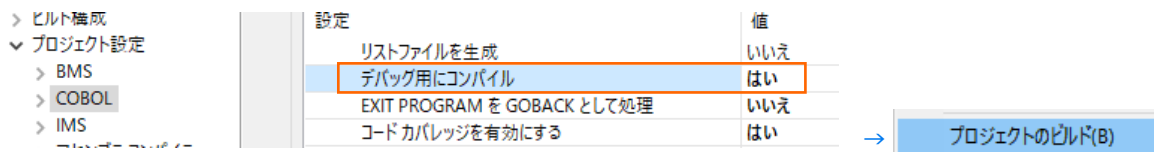
1. デバッグ用ファイルの準備
2. データファイル ツールの起動
3. レコードレイアウトの作成
4. 条件付きレコードレイアウトの作成
5. データのメンテナンス
6. JES カタログファイルとの連携
7. MFDBFH との連携
8. データの文字コード変換
9. 半角カタカナの扱い

3.1 デバッグ用ファイルの準備

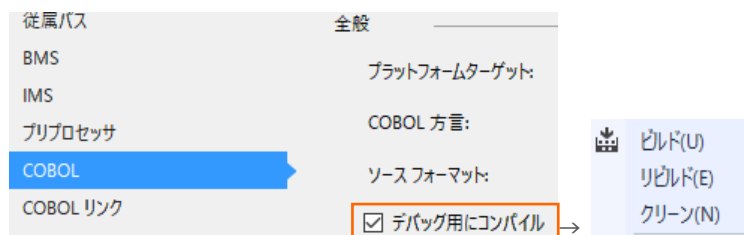
データファイルツールは、データとレコードレイアウトを関連付けて、項目ごとのメンテナンスが可能な機能をご提供しています。このレコードレイアウトの作成には、コンパイル時に生成されるデバッグ用ファイルが必要になります。まずはこのデバッグ用ファイルを生成します。

- 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] にチェックして、プロジェクトのビルドを行います。

[Eclipse]

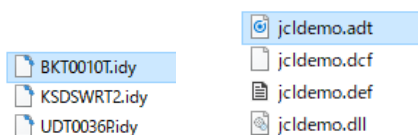


[Visual Studio]



- 3) 指定した出力フォルダに、COBOL は XXXXX.idy、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レコードレイアウトに利用するために必要となります。

例)



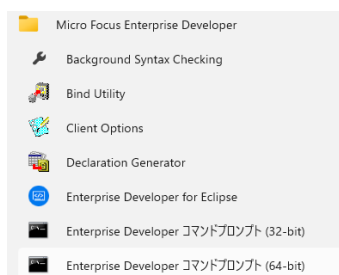
3.2 データファイル ツールの起動

データファイル ツールを起動します。この際、環境変数 MFCODESET の値によりデータ表示や文字コード変換の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。本チュートリアルでは 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 MFCODESET=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、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 MFCODESET=9122

ツール起動コマンド) mdatatools2

```
Enterprise Developer コマンドプロンプト (64-bit)
C:\Users\tarot\Documents> set MFCODESET=9122
C:\Users\tarot\Documents> mdatatools2
```

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 MFCODESET=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、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 MFCODESET=939

ツール起動コマンド) mdatatools2

```
Enterprise Developer コマンドプロンプト (64-bit)
C:\Users\tarot\Documents> set MFCODESET=939
C:\Users\tarot\Documents> mdatatools2
```

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、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。

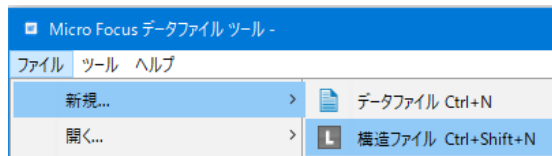
【環境変数を設定しない場合:メニューから起動】



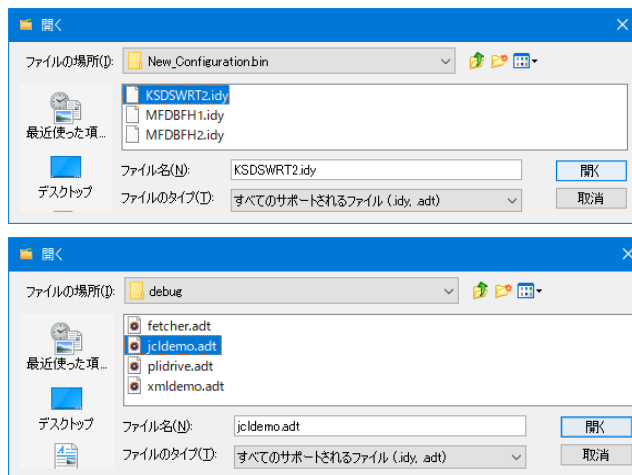
3.3 レコードレイアウトの作成

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。

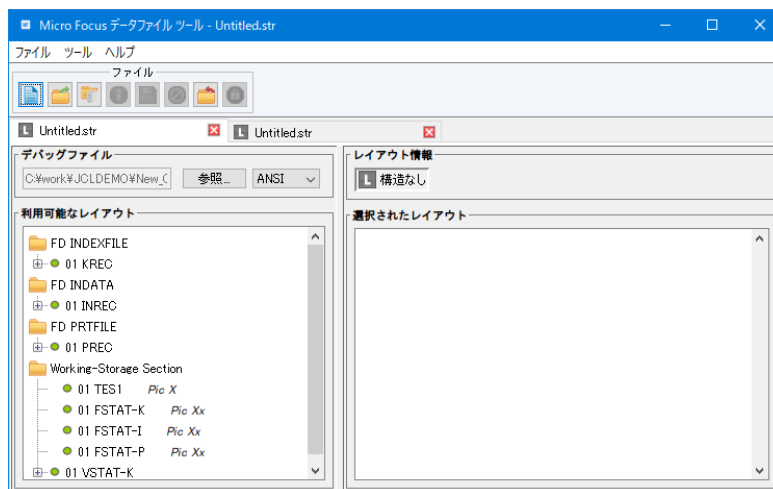
- 1) [ファイル] プルダウンメニュー > [新規] > [構造ファイル] を選択します。



- 2) COBOL の場合は .idy ファイルを、PL/I の場合は .adt ファイルを選択し、[開く] ボタンをクリックします。



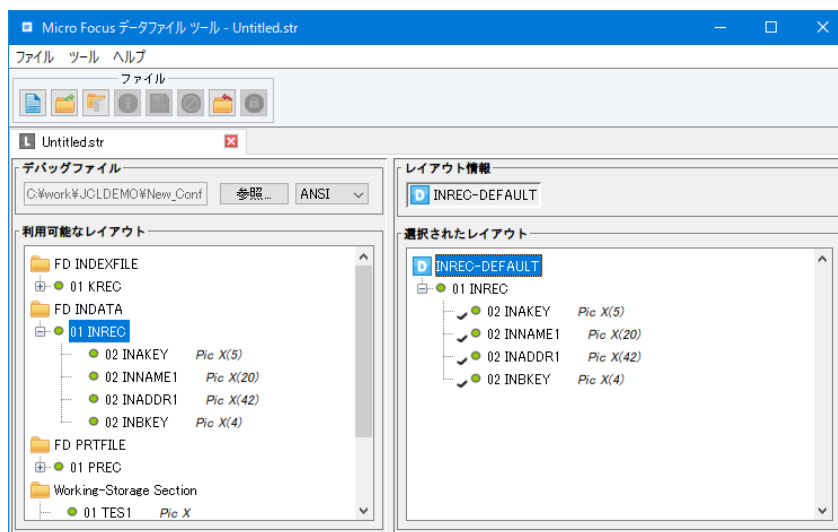
- 3) レイアウト ウィンドウの左側ツリービューに使用されている構造体が表示されます。



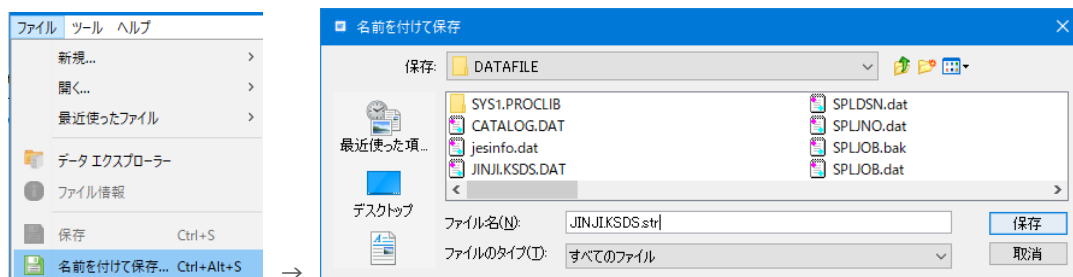
- 4) 左側ツリービューからレコードレイアウトとして利用したい構造体を右クリックし、[デフォルトレイアウトの作成] もしくは [条件付きレイアウトの作成] を選択します。[条件付きレイアウト] を選択すると構造体に含まれるキーを判断して複数レイアウトへ振り分けが可能になります。ここでは [デフォルトレイアウトの作成] を選択します。



- 5) 右側フレームに選択した構造体が表示されます。



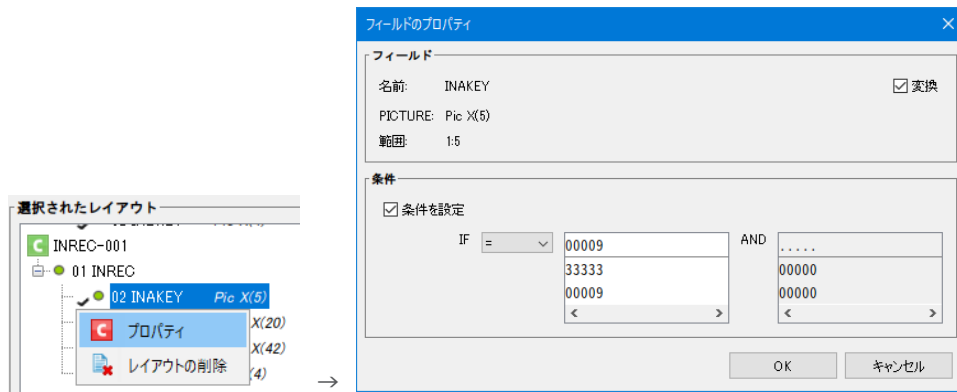
- 6) [ファイル] プルダウンメニューから [名前を付けて保存] を選択し、レコードレイアウトを適用したいデータファイルが存在するパスへ、データファイルと同名を指定して [保存] ボタンをクリックします。レコードレイアウトの拡張子は .str です。



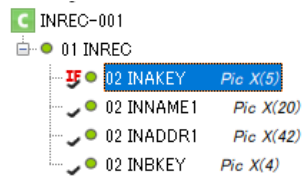
3.4 条件付きレコードレイアウトの作成

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) 前項と同様に使用する構造体を右側フレームにクリックし、[条件付きレイアウト] を選択します。
- 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 [プロパティ] 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。

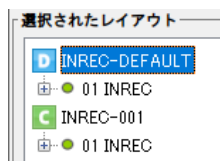


条件項目に [IF] と表示されます。



- 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ます。

- [選択されたレイアウト] には次のように表示されます。[ファイル] プルダウンメニューから [保存] を選択して保存します。

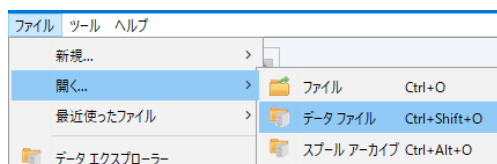


- この設定により、条件に沿ったレイアウトが適用されてデータが表示されます。
[ファイル] プルダウンメニュー > [閉じる] を選択して構造ファイルウィンドウを終了します。

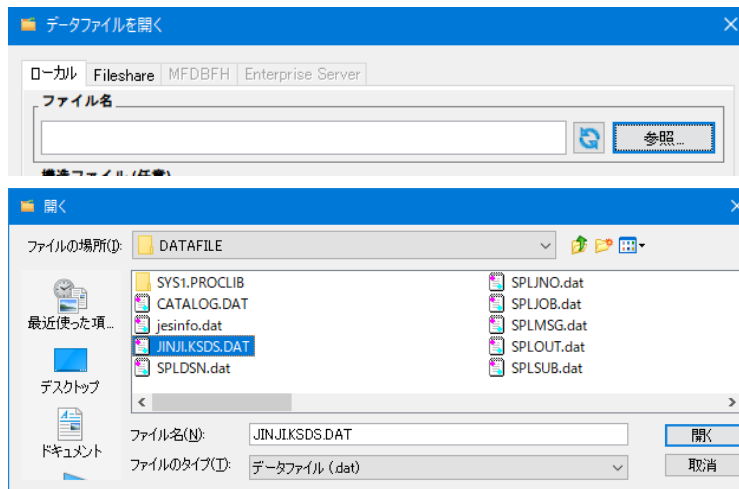
3.5 データのメンテナンス

データファイルを開いてメンテナンスを行います。

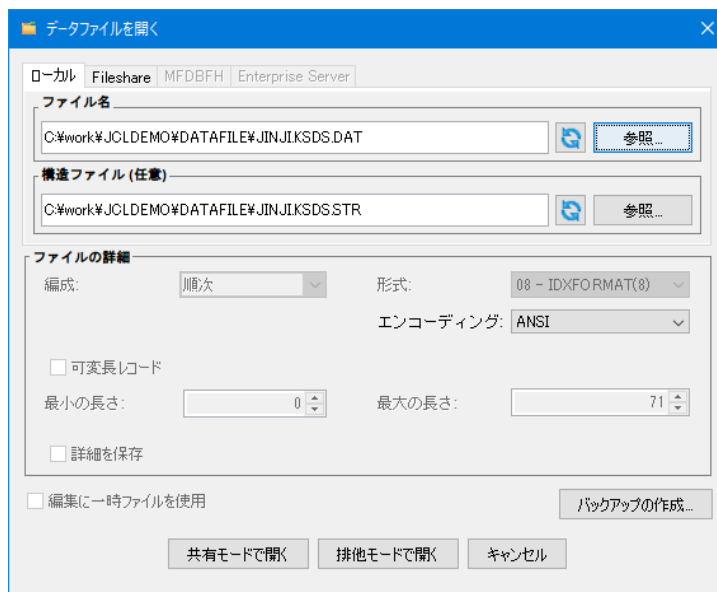
- [ファイル] プルダウンメニュー > [開く] > [データファイル] を選択します。



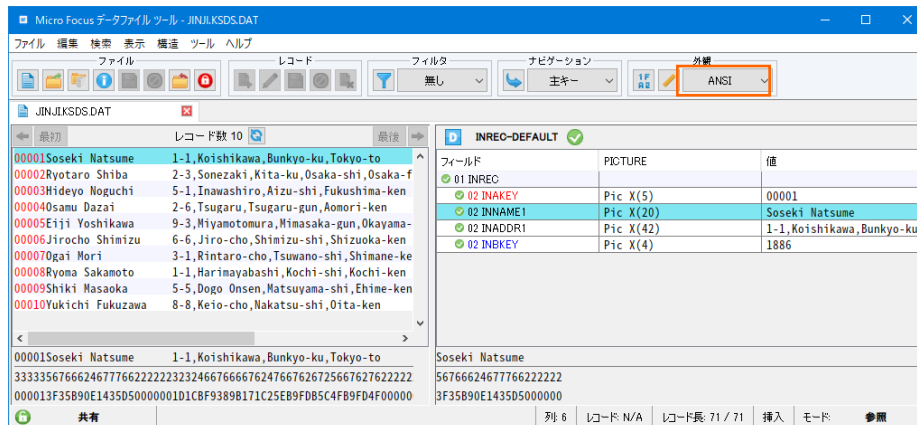
- [データファイルを開く] ウィンドウの [ファイル名] の [参照] ボタンからメンテナンスを行うファイルを選択し、[開く] ボタンをクリックします。



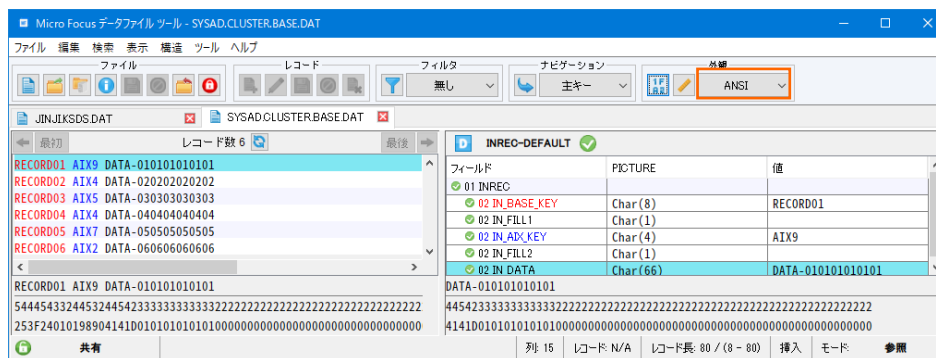
- 3) データファイル名と構造ファイル名が同名の場合は自動的に構造ファイルが適用されてデータが表示されます。異なる名前の構造ファイルを適用させる場合には [参照] ボタンから選択してください。
- また、データファイルを開く際に [共有モード] または [排他モード] を選択できます。ここでは [共有モードで開く] を選択します。



- 4) 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]



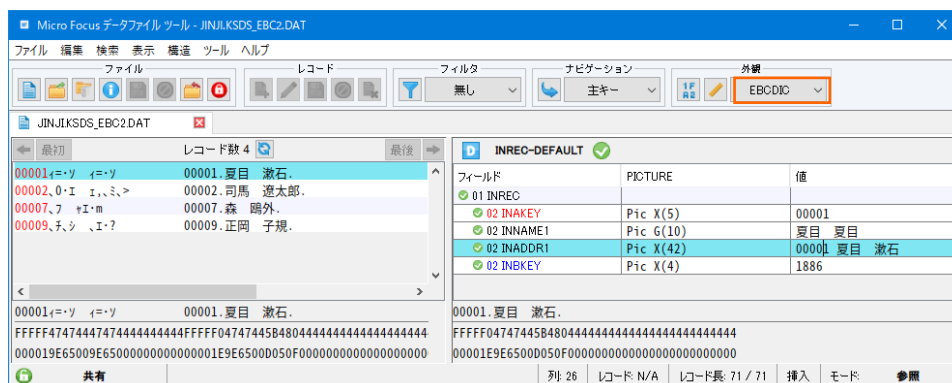
[PL/I ANSI 文字コードデータ]



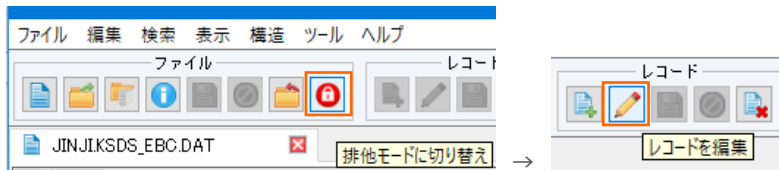
補足)

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。

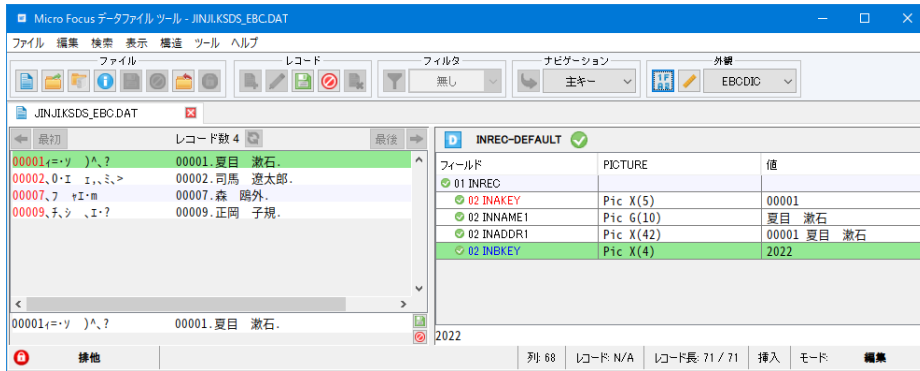
[COBOL EBCDIC 文字コードデータ]



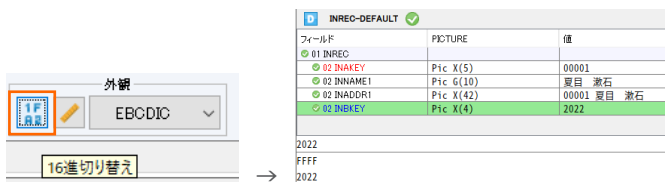
- 5) データを修正するために [排他モード] へ切り替え、[レコードを編集] アイコンをクリックします。



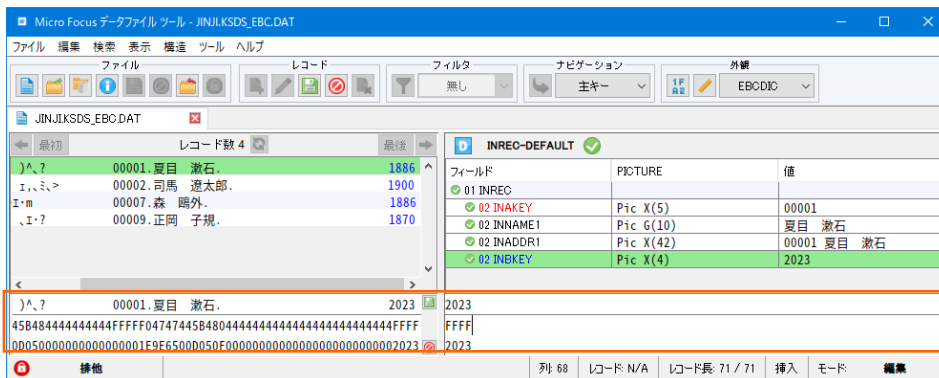
- 6) 選択行がグリーンになり変更可能になります。右フレーム内で [INBKEY] 項目を 2022 へ変更します。



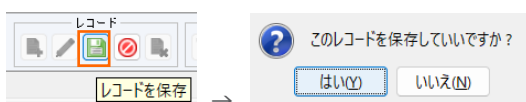
- 7) 変更後、変更レコードを選択した状態で [16 進切り替え] アイコンをクリックして値を確認します。



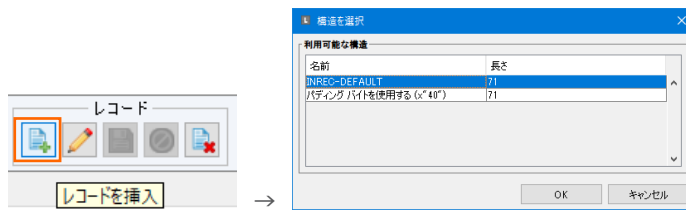
このケースでは EBCDIC 文字コードが適用されていることがわかります。ウィンドウ下部から値を入力することも、直接 16 進数を入力することも可能です。また、右フレームには選択されている項目のみが表示されます。



変更後は [レコードを保存] アイコンをクリックし、確認ウィンドウで [はい] をクリックします。

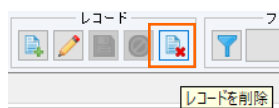


- 8) 新規データを作成する際には、[レコードを挿入] アイコンをクリックし、適用する構造を指定します。左側フレームで右クリックにより挿入することも可能です。



データ挿入時の適用コードは、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。

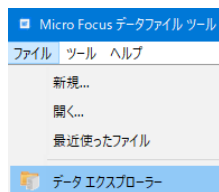
- 9) 既存データを削除する際には、[レコードを削除] アイコンをクリックします。左側フレームで右クリックにより削除することも可能です。



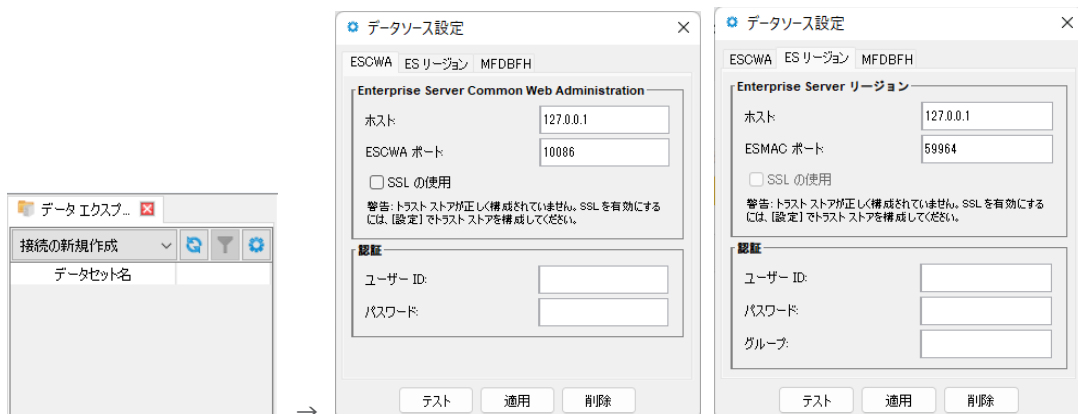
3.6 JES カタログファイルとの連携

データファイルツールは ESCWA と連携して JES 機能を持つ Enterprise Server インスタンスのカタログファイルと連携してデータをメンテナンスすることができます。各インスタンスと個々に連携することも可能です。

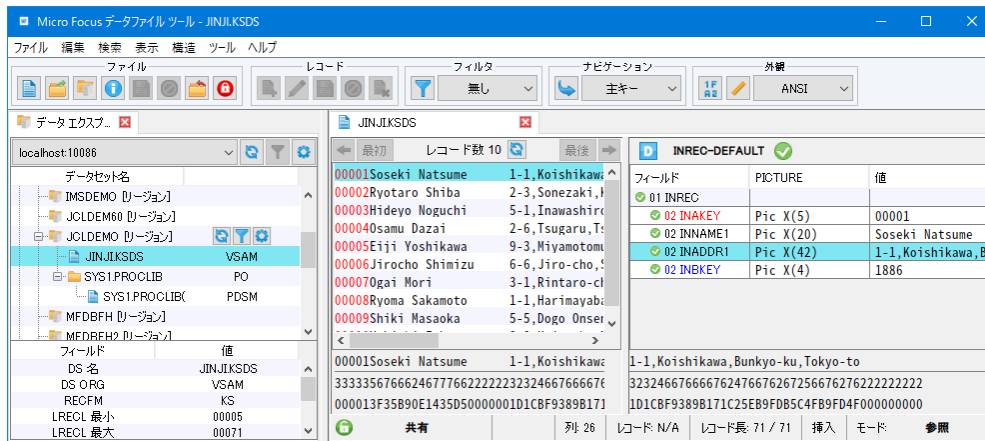
- 1) [ファイル] プルダウンメニューから [データエクスプローラー] を選択します。



- 2) [データエクスプローラー] タブで [接続の新規作成] を選択すると [データソース設定] ウィンドウが表示されます。[ESCWA] タブでは対象マシンの IP アドレスを [ホスト] へ、デフォルトポート番号である 10086 を [ESCWA ポート] へ設定します。[ES リージョン] タブでは、接続したいインスタンスが存在する IP アドレスを [ホスト] へ、対象インスタンスが持つ Web Services and J2EE リスナーのポート番号を [ESMAC ポート] へ設定し、[適用] ボタンをクリックします。ユーザー認証は必要に応じて入力してください。



- 3) ESCWA 接続の例では、[データエクスプローラー] タブに全てのインスタンスが表示されて一括管理ができます。また、開始されている各インスタンスのカatalogファイルを通じて、ファイルの情報を参照しながらデータをメンテナンスすることもできます。



3.7 MFDBFH との連携

データファイルツールは使用するファイルをデータベースに格納できる MFDBFH と連携して KSDS フォーマットの VSAM ファイルのデータをメンテナンスすることができます。

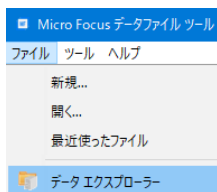
- 1) Enterprise Developer コマンドプロンプトから MFDBFH 構成ファイルを環境変数で指定後、データファイルツールを起動します。

環境変数コマンド例) set MFDBFH_CONFIG=C:¥MFDBFH80¥MFDBFH.cfg

ツール起動コマンド) mfdatatools2

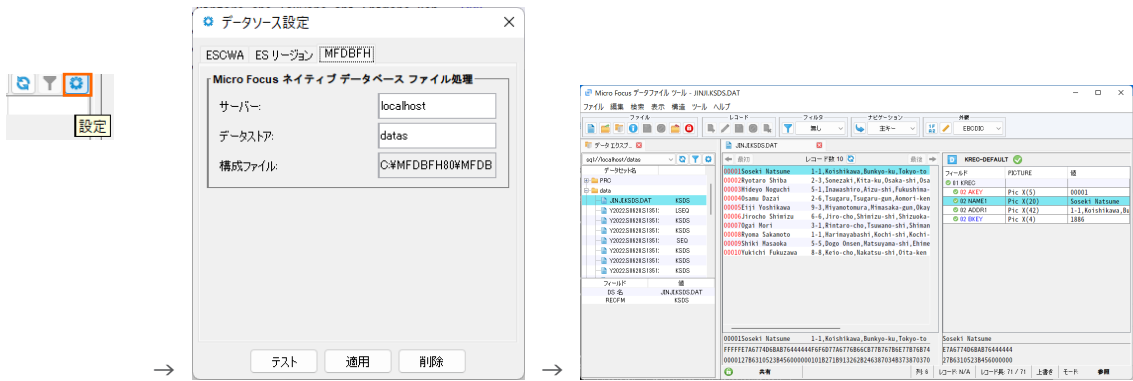
```
C:¥work¥JCLDEMO>set MFDBFH_CONFIG=C:¥MFDBFH80¥MFDBFH.cfg
C:¥work¥JCLDEMO>mfdatatools2
```

- 2) [ファイル] プルダウンメニューから [データエクスプローラー] を選択します。



- 3) 既に接続が存在する場合は [データエクスプローラー] タブの [設定] アイコンをクリックして [データソース設定] ウィンドウを表示します。[MFDBFH] タブを選択し、対象マシンの IP アドレスを [サーバー] へ、対象データストア名を [データストア] へ入力し、[適用] ボタンをクリックします。

接続後は KSDS フォーマットの VSAM ファイルのデータをメンテナンスすることができます。



3.8 データファイルのコード変換

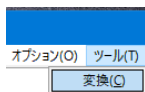
EBCDIC 文字コードデータファイルから ANSI 文字コードデータファイルへ、またその逆コードへの変換を行うことができます。データファイルの変換はクラシックデータツール、もしくは DFCONV コマンドから実行します。

- 1) データファイルツールと同様に、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もしくはメニューから起動します。

ツール起動コマンド) mfdatatools もしくは

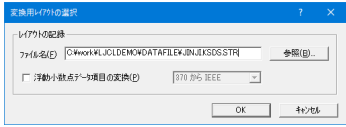
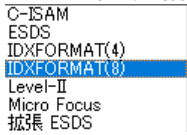
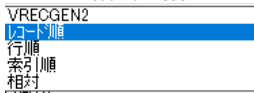
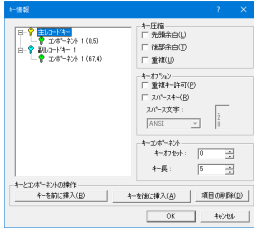


- 2) [ツール] プルダウンメニュー > [変換] を選択して [データファイルの変換] ウィンドウを表示します。



- 3) 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。

- 4) 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] ボタンで適用後は、変換設定内容が正しいか必ず確認してください。

- 5) 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: V10.001.01
Batch-Process: Convert-File
*****
* Options: Convert-File
* Options: None
Start-File:
Edit-Mode:
*****
Input-File:
Format:
*****
* Options: Micro-Focus
* Options: IDFORMAT(4)
* Options: Reverse
* Options: EBCDIC
* Options: LEVEL=11
* Options: IDFORMAT(8)
* Options: Extended-EBCDIC
* Options: Vision
*****
Organization: Sequential
*****
* Options: Sequential Line-Sequential
* Options: Inlined
* Options: Mainframe-Report-ASCII
* Options: Relative
* Options: Mainframe-Report-mach
* Options: RESLEN
* Options: PO-Print
* Options: Blocked-View
*****
Record-Format: Fixed
*****
* Options: Fixed/Variable
*****
Character-Set: EBCDIC
*****
* Options: ASCII/EBCDIC
*****
Final-Line-Point: IEEE
*****
* Options: IEEE/SFD
*****
Compression: Off
*****
* Options: On/Off
*****
Min-Rec-Length: 0
Max-Rec-Length: 128

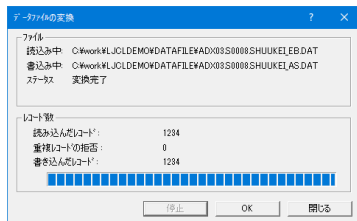
```

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。



注意) 変換対象ファイルが編集中は実行できません。

7) [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。



8) [データファイルの変換] ウィンドウを閉じて、[ファイル] プルダウンメニュー > [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】



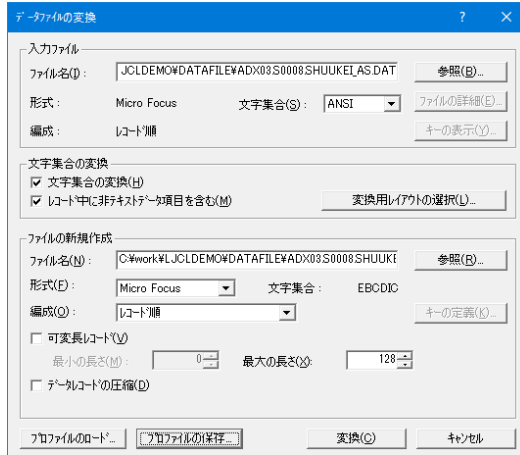
【変換後:ANSI 文字コードデータ】



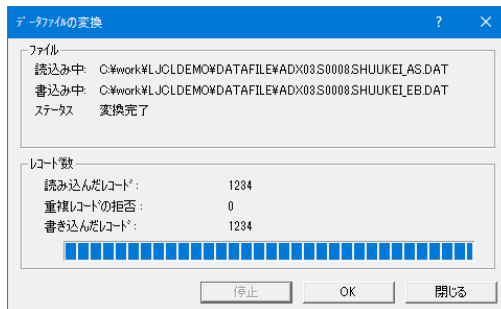
漢字の前後にあるシフトコードが削除されて最後部に 2 バイト追加されています。

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。

9) 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。



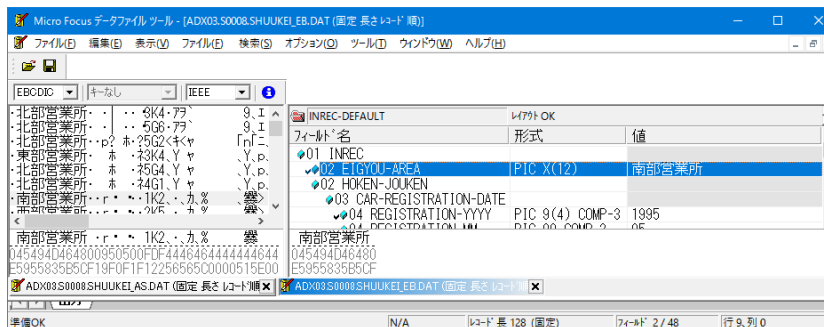
前項と同様に変換します。



【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。

3.9 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 16 進数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	28	←
FE	28	←
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の [リファレンス] > [コードセットの変換] > [Codecomp ユーティリティ] > [文字セット変換をカスタマイズするには] をご参照ください。

4. 免責事項

本チュートリアル of 例題ソースコードは機能説明を目的としたサンプルであり、無謬性を保証するものではありません。例題ソースコードは弊社に断りなくご利用いただけますが、本チュートリアルに関わる全てを対象として、二次的著作物に引用する場合は著作権法の精神に基づき適切な扱いを行ってください。

本チュートリアルで学習した技術の詳細については製品マニュアルをご参照ください。