Micro Focus Visual COBOL チュートリアル

ファイルハンドラーを使ったデータファイルアクセス

1. 目的

メインフレームやオフコンからオープンプラットフォームに COBOL 資産を移行した時に必要になる処理は、やはりデータファイルアクセスではないでしょうか。Visual COBOL を使用しないでこれらのデータファイルにアクセスする場合、OS が提供するローレベルな API を使ってカーネルディスクドライバーを経由しファイルにアクセスします。ただし、この API が提供するのは低水準のバイトストリームファイルアクセスであり、COBOL アプリケーションから発行されるレコード単位による I/O 処理には対応できません。Visual COBOL は、 COBOL アプリケーションと OS の API の間にファイルハンドラーというインターフェースを介して COBOL アプリからの各種 I/O のハンドリングを行います。これにより、オープンプラットフォームにおいても、これまで同様、レコード単位によるデータファイルへのアクセスが可能になります。データファイルの種類には順編成ファイル、索引編成ファイルなどの種類が存在します。

順編成ファイルは、バッチ処理で扱うトランザクションファイルの操作に優れています。多くのバッチ処理は、夜間などのコンピュータ資源を占有できる環境で実行されるため、排他制御などのオーバーヘッドを受けることなく効率的に実行できる点にメリットがあります。一方で、夜間のバッチ処理は翌朝の業務開始時刻までには確実に処理が完了していることが必須であり、オンライン処理のレスポンスタイムと同様に、あるいは、それ以上に厳しい性能要求にさらされる処理となっています。たとえ 0.01 秒しかかからない処理でも 100 万回繰り返すことで 3 時間以上になりますので、大量バッチ処理を定時までに完了させるためには慎重なプログラミングが必要となります。トランザクションファイル 1 件の読み込みでも最も高速な手段が要求されるため、これを COBOL の順編成ファイルとして扱うことは最適な選択となります。
索引編成ファイルは、キー値で指定されたファイル内の固有のレコードにランダムにアクセスできる機能を持つファイル編成です。索引編成ファイルはマスターファイルとして使用されます。たとえば従業員マスターファイルは全従業員に関する様々な情報を保持していますが、従業員番号や従業員名などでランダムアクセスできなければなりません。
索引編成ファイルの概念は SQL を使い慣れた方には理解しやすいものです。実際、歴史的には索引編成ファイルは、データベースの概念に先立って登場しており、その後複数の索引編成ファイルの有機的な統合を実現するためにデータベース管理システムが考案された経緯があります。

このチュートリアルでは、ファイルハンドラーを使用したファイルのアクセス方法、ソートユーティリティの概要、rebuild ユーティリティの使用方法を学びます。

2. 前提条件

本チュートリアルは、下記の環境を前提に作成されています。

- 開発クライアント ソフトウェア

OS	Windows Server 2019 Standard Edition（64bit）
COBOL 製品	Visual COBOL 8.0 for Eclipse

- チュートリアル用プログラム

  下記のリンクから事前にチュートリアル用のプログラムやデータファイルをダウンロードして、任意のフォルダに解凍しておいてください。

  プログラムおよびデータファイルのダウンロード

内容

3. チュートリアル手順の概要

3.1. データファイルの準備

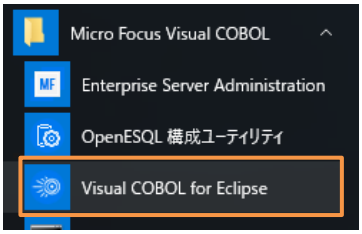
1) チュートリアル用データファイルの用意

① ダウンロードしたファイルを任意の場所に解凍します。work フォルダ、MFSORT フォルダ、REBUILD フォルダが入っていることを確認します。

3.2. Visual COBOL for Eclipse の起動とプロジェクト作成の準備

1) Visual COBOL for Eclipse の起動とプロジェクトの作成

① スタートメニューより［Micro Focus Visual COBOL］>［Visual COBOL for Eclipse］を選択します。

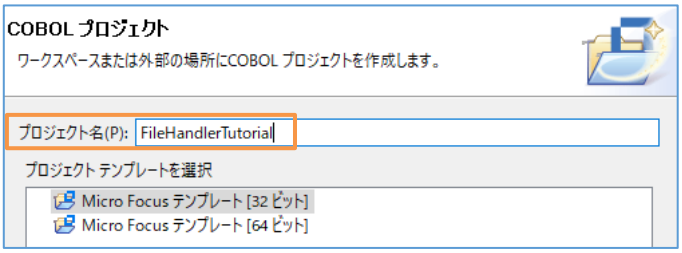


② ワークスペースの選択画面は任意のワークスペースを指定して、[起動(L)] ボタンをクリックします。

3.3. ネイティブ COBOL プログラムの作成と動作確認

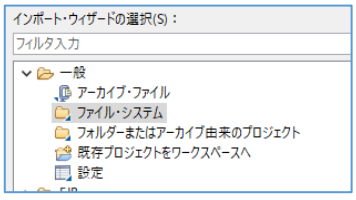
1) ネイティブ COBOL プロジェクトの作成とプログラムのインポート

① Eclipse メニューより、[ファイル(F)] > [新規(N)] > [COBOL プロジェクト] を選択し、プロジェクト名に“FileHandlerTutorial”を指定して、[終了(F)] ボタンをクリックします。



② COBOL エクスプローラーにて作成した「FileHandlerTutorial」プロジェクトを右クリックし、コンテクストメニューから [インポート(I)] > [インポート(I)] を選択します。

③ [一般] > [ファイル・システム] を選択し、[次へ(N)] ボタンをクリックします。



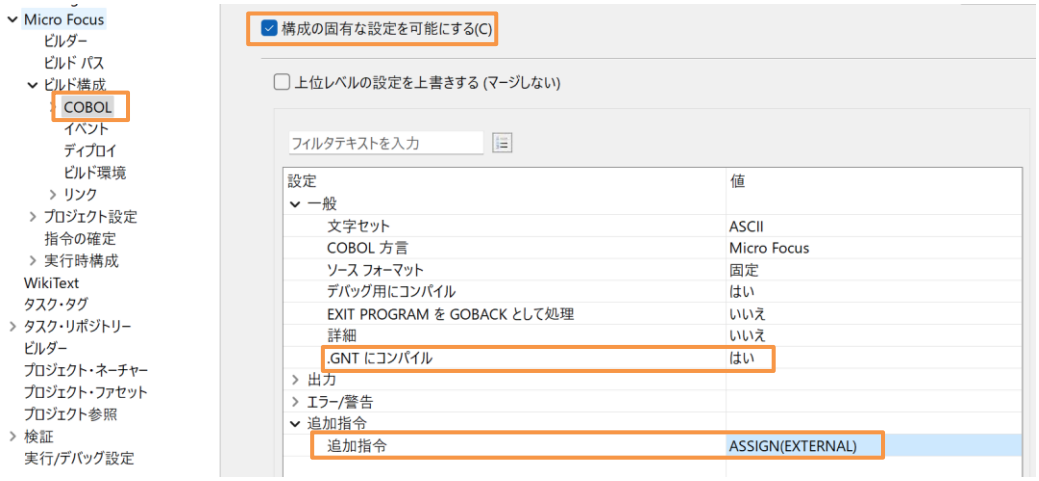
④ [参照(R)…] をクリックし、エクスプローラから解凍したフォルダ配下の work フォルダを指定します。「FileDemo1.cbl」にチェックを入れて［終了(F)］ボタンをクリックします。

2) プロジェクトプロパティの指定

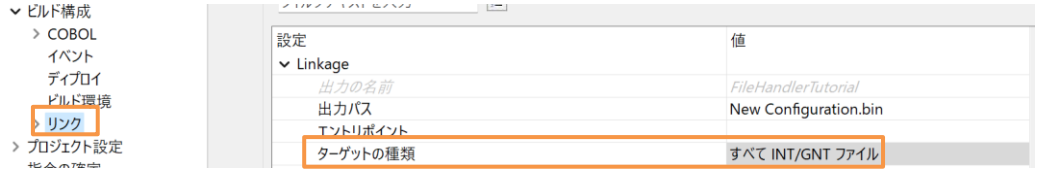
① プロジェクトの構成を変更します。COBOL エクスプローラーにて作成した「FileHandlerTutorial」プロジェクトを右

クリックし、コンテクストメニューから［プロパティ(R)］を選択します。

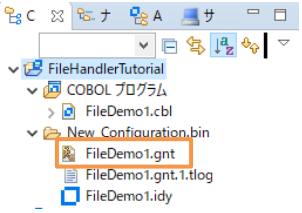
② プロパティ設定ダイアログが表示されます。[Micro Focus] > [ビルド構成] > [COBOL］をクリックし、[構成の固有な設定を可能にする(C)］にチェックを入れたうえで、[.GNT にコンパイル］を「はい」、[追加指令]に、「ASSIGN(EXTERNAL)」を入力します。



③ 次に［ビルド構成] > [COBOL] > [リンク］をクリックし、［ターゲットの種類］から「すべて INT/GNT ファイル」を選択し、[適用して閉じる］をクリックします。



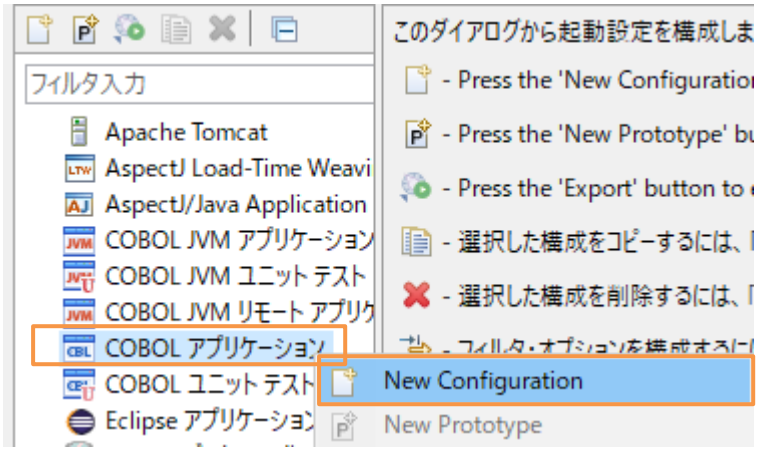
④ 自動的にビルドが行われます。実行ファイルが作成されていることを確認します。



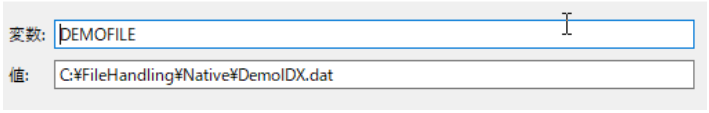
3) プログラムの実行

① 「FileDemo1.gnt」上で右クリックし、コンテクストメニューから［実行(R)] > [実行の構成(N)...］を選択します。

② ［COBOL アプリケーション］を選択し、マウスの右クリックにてコンテクストメニューを開き、[New Configuration］をクリックします。



③ [名前] に “FILEDEMO” を入力し、[環境] タブを選択し、 [追加] ボタンをクリックします。[変数] に ”DEMOFILE”、値には ″C:¥FileHandling¥Native¥DemoIDX.dat” を指定して、[OK] ボタンをクリックします。※子のフォルダは任意です。



④ エクスプローラーより「C:¥FileHandling¥Native」ディレクトリを作成します。

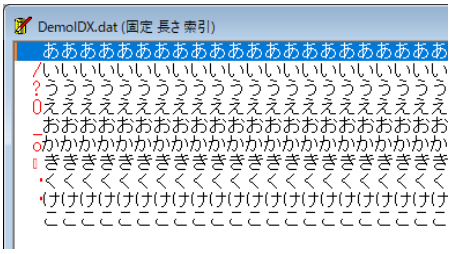
⑤ [実行] ボタンをクリックすると、DOS プロンプトが表示されプログラムが実行され、ファイル作成が行われます。「C:¥FileHandling¥Native¥DemoIDX.dat」が作成されているか確認します。

4) 実行結果の確認

① Windows のスタートメニューから [Micro Focus Visual COBOL] > [クラシックデータファイルツール] を選択します。

② [ファイル(F)] メニュー から [開く(O)] を選択し、「C:¥FileHandling¥Native¥DemoIDX.dat」をオープンします。

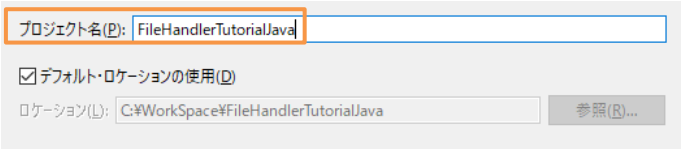
③ 以下のようにファイルの中身が表示されます。



3.4. JVM COBOL プログラムの作成と動作確認

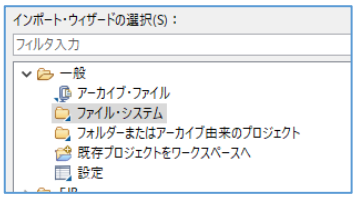
5) JVM COBOL プロジェクトの作成とプログラムのインポート

① Eclipse メニューより、[ファイル(F)] > [新規(N)] > [COBOL JVM プロジェクト] を選択し、プロジェクト名に “FileHandlerTutorialJava” を指定して、[終了(F)] ボタンをクリックします。



② COBOL エクスプローラーにて作成した「FileHandlerTutorialJava」プロジェクト配下の src フォルダを右クリックし、コンテクストメニューから［インポート(I)］>［インポート(I)］を選択します。

③ ［一般］>［ファイル・システム］を選択し、［次へ(N)］ボタンをクリックします。



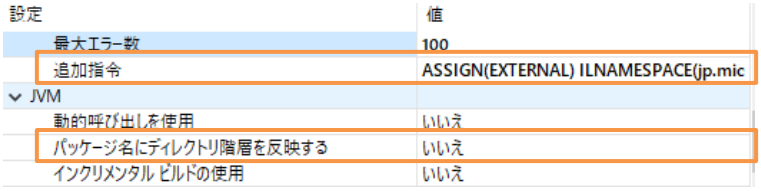
④ ［参照(R)…］をクリックし、エクスプローラから解凍したフォルダ配下の work フォルダを指定します。「FileDemo1.cbl」にチェックを入れて［終了(F)］ボタンをクリックします。

6) プロジェクトプロパティの指定

① プロジェクトの構成を変更します。COBOL エクスプローラーにて作成した「FileHandlerTutorialJava」プロジェクトを右クリックし、コンテクストメニューから［プロパティ(R)］を選択します。

② プロパティ設定ダイアログが表示されます。［Micro Focus］>［ビルド構成］をクリックし、［追加指令］に、「ASSIGN(EXTERNAL) ILNAMESPACE(jp.microfocus.demo)」を入力します。

③ 下にスクロールし、［JVM］>［パッケージ名にディレクトリ階層を反映する］を「いいえ」に変更します。



④ ［適用して閉じる］ボタンをクリックすると、自動的にビルドが行われます。

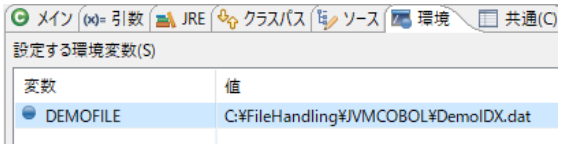
7) プログラムの実行

① 「FileHandlerTutorialJava」プロジェクト上で右クリックし、コンテクストメニューから[実行] >［実行の構成(N)…］を選択します。

② 左側の［COBOL JVM アプリケーション］上で右クリックし、コンテクストメニューから［New Configuration］を選択します。名前は、”FileDemoJava” と入力します。

③ ［メイン］タブの［メイン・クラス(M):］横にある［検索(S)］ボタンをクリックし、「FILEDEMO1 – jp.microfocus.demo」を選択し、［OK］ボタンをクリックします。

④ ［環境］タブを選択し、［新規(E)］ボタンをクリックします。［変数(N)］に ”DEMOFILE”、［値(V)］には ”C:¥FileHandling¥JVMCOBOL¥DemoIDX.dat” を入力し、［OK］ボタンをクリックします。



⑤ エクスプローラーより「C:¥FileHandling¥JVMCOBOL」ディレクトリを作成します。

⑥ [実行(R)] ボタンをクリックすると、プログラムが実行され、ファイル作成が行われます。

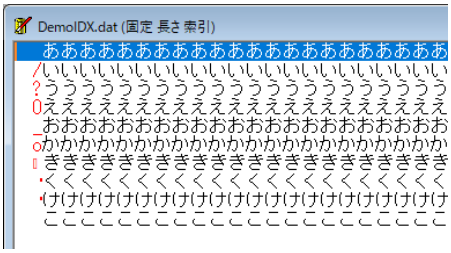
「C:¥FileHandling¥JVMCOBOL¥DemoIDX.dat」が作成されているか確認します。

8) 実行結果の確認

① Windows メニューから［Micro Focus Visual COBOL］> [クラシックデータファイルツール］を選択します。

② [ファイル(F)] メニュー から［開く(O)］を選択し、「C:¥FileHandling¥JVMCOBOL¥DemoIDX.dat」をオープンします。

③ 以下のようにファイルの中身が表示されます。



3.5. クラシックデータファイルツール

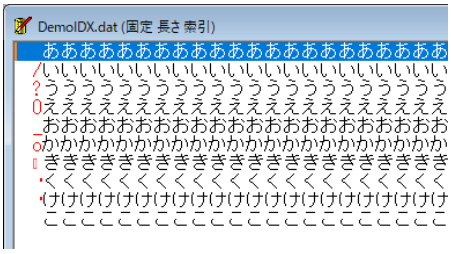
データファイルツールは COBOL ファイル編成の変換、ファイルのブラウズ、レコードの編集等の機能を持った GUI ツールです。

1) クラシックデータファイルツールの起動とファイルの読み込み

① 前の章で使用したファイルをオープンしていない場合は再度オープンします。

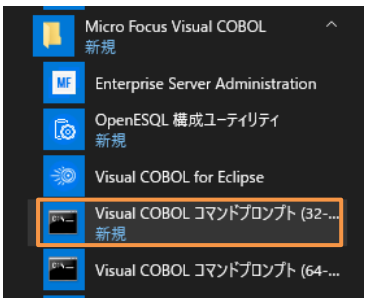
② Windows のスタートメニューから[Micro Focus Visual COBOL] > [クラシックデータファイルツール］を選択します。

③ [ファイル(F)] メニュー から［開く(O)］を選択し、「C:¥FileHandling¥JVMCOBOL¥DemoIDX.dat」をオープンします。



2) デバッグ情報ファイルからレコードレイアウトを作成

① Windows のスタートメニューから［Micro Focus Visual COBOL］> [Visual COBOL コマンドプロンプト(32-bit)］を選択します。



② 「Data File Structure Command Line ユーティリティ」を使用してデバッグ情報ファイルからレコードレイアウトを作成します。下記の通りタイプしてレコードレイアウトを生成します。レイアウトファイルは、指定した idy ファイルと同じフォルダに出力されます。

dfstrcl “<FileHandlerTutorial ワークスペースフォルダへのパス>¥New_Configuration.bin¥FileDemo1.idy” /d F-REC

<FileHandlerTutorial ワークスペースフォルダへのパス> この部分は、各自の環境に合わせて変更してください。

例えば、FileHandlerTutorial ワークスペースフォルダが “C:¥WorkSpace¥FileHandlerTutorial” の場合は、以下のようになります。

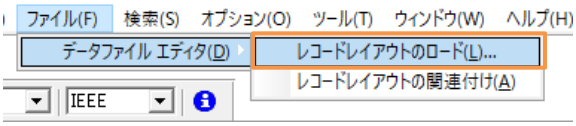
dfstrcl "C:¥WorkSpace¥FileHandlerTutorial¥ New_Configuration.bin¥FileDemo1.idy" /d F-REC

補足）
Eclipse IDE 上で FileHandlerTutorial ワークスペースへのフォルダパスを、以下の手順で確認できます。

① FileHandlerTutorial プロジェクトを選択し、マウスの右クリックにて、[プロパティ(R)] をクリックします。
② 画面左側より [リソース] を選択すると、右側の [ロケーション(L)] にワークスペースへのフォルダパスが表示されます。

3) レコードレイアウトを適用し可視化できないデータを確認

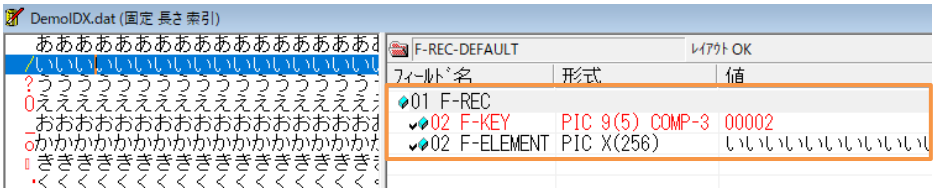
① [ファイル(F)] メニュー ＞ [データファイルエディタ(D)] ＞ [レコードレイアウトのロード(L)...]を選択します。



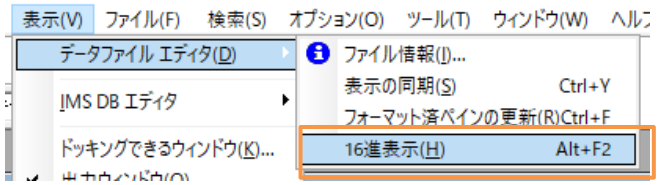
② 前のステップで作成した「FileDemo1.str」ファイルを指定します。

注意）
このファイルは、前手順にて dfstrcl コマンドで指定した idy ファイルと同じフォルダにあります。

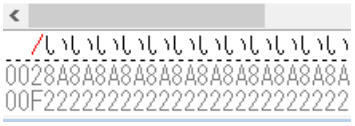
③ ファイルの中身がレコード単位で確認できるようになります。



④ [表示(V)] メニュー ＞ [データファイルエディタ(D)] ＞ [16 進表示(H)] を選択します。



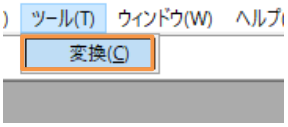
⑤ データの内容が 16 進でも表示されます。



4) 他のファイル編成に変換

① データファイルを開いているウィンドウを一旦閉じます。

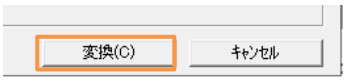
② [ツール(T)] メニュー > [変換(C)] を選択します。



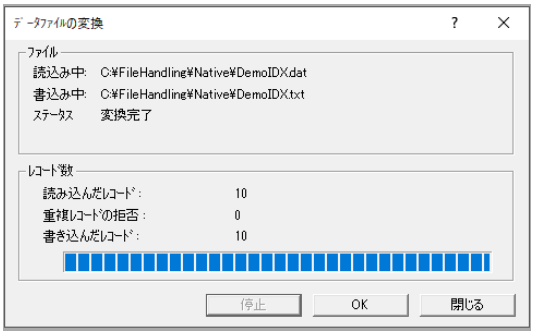
③ [データファイルの変換] ウィンドウが表示されます。[入力ファイル] セクションの [参照(B)...] ボタンをクリックし、エクスプローラより「C:¥FileHandling¥Native¥DemoIDX.dat」をオープンします。

④ [ファイルの新規作成] セクションの [参照(R)...] ボタンをクリックし、エクスプローラより「C:¥FileHandling¥Native¥DemoIDX.txt」を指定します。次に [編成(O)] を「行順」に変更します。

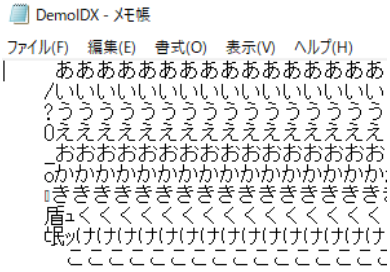
⑤ [変換(C)] ボタンをクリックします。



⑥ 「データファイルの変換」ウ ィンドウが表示され、変換処理が完了しますので [閉じる] ボタンをクリックします。



⑦ メモ帳で作成したファイルを開きます。各レコードが改行で区切られた行順ファイルになっていることを確認します。



3.6. ソートユーティリティ MFSOFT の確認

ソートユーティリティ MFSORT は、IBM メインフレームの DFSORT とコマンド互換のあるソート・マージユーティリティです。

1) ソート用ファイルの用意

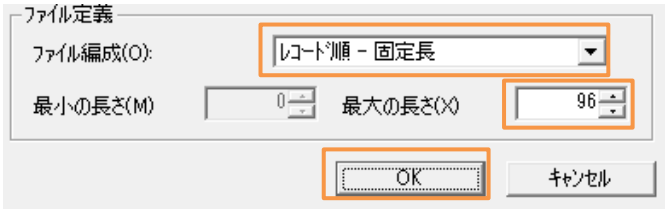
① 解凍したフォルダに MFSORT フォルダが含まれていることを確認します。

② MFSOFT フォルダを C ドライブ配下に配置します。例：C:¥MFSORT

2) クラシックデータファイルツールの起動と読み込み

① クラシックデータファイルツールを起動して MFSORT フォルダ配下の「airports.dat」を開きます。

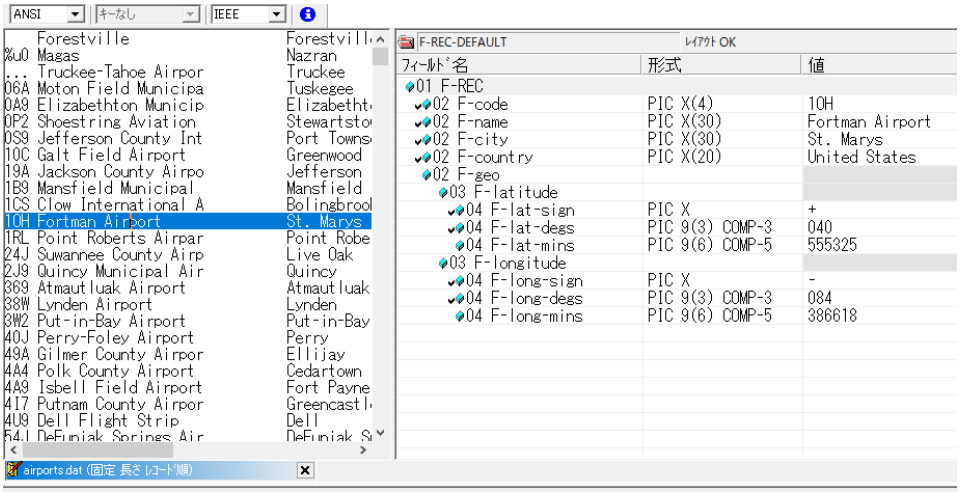
② [ファイル編成(O)] に “レコード順 – 固定長” が選択されていることを確認し、[最大の長さ(X)] に “96” を入力したうえで、[OK] をクリックします。



以下のようなダイアログが表示された場合は、[OK] をクリックします。

③ [ファイル(F)] > [データツールエディタ(D)] > [レコードレイアウトのロード(L)…] を選択し、MFSORT フォルダ配下の「AIRPORTSEQ.STR」ファイルを指定して、[開く(O)] をクリックします。

④ ファイルの中身がレコード単位で確認できるようになります。



3) ソート順の変更

① このファイルは「F-code」と呼ばれる空港コードの昇順でソートされています。これを「F-name」という空港名でソートしてみます。

② ソート用のコマンドファイルを用意します。１行目はインプットするデータファイルの指定、２行目はアウトプットするデータファイルの指定、３行目がソートするフィールド名の指定です。ここでは「F-name」を昇順で指定するので 5 バイト目から 30 バイト、昇順（ascending）に並べ替えを指定しています。

```
use airports.dat record (f, 96)
give sorted.dat ORG SQ
sort fields (5, 30 CH, a)
```

補足）
上記ファイルは、MFSORT フォルダ配下内の airport.srt ファイルとして保存されています。

⑧ Windows のスタートメニューから［Micro Focus Visual COBOL］>［Visual COBOL コマンドプロンプト(32-bit)］を選択します。

⑨ CD C:¥MFSORT」を実行し、C:¥MFSORT フォルダまで移動します。

⑩ mfsort コマンドを実行します。コマンドラインに「mfsort take airport.srt」とタイプしてリターンキーを押します。

注意）
クラシックデータファイルツールなどで airports.dat を開いている場合、上記コマンドを実行する前に閉じてください。

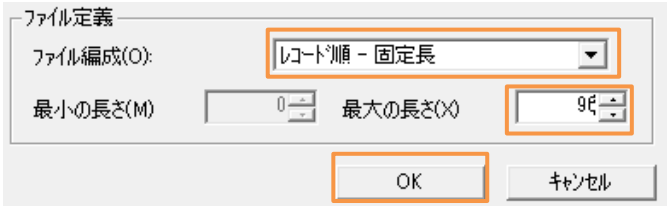
4) 実行結果の確認

① 「SYSOUT」というファイルに実行結果ログが入っているのでメモ帳を起動して内容を確認します。

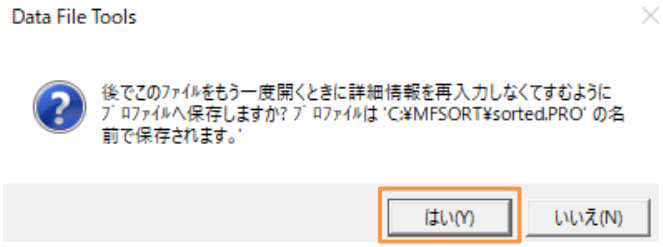
```
|          Micro Focus MFJSORT ユーティリティ 3.0.00

           take airport.srt
           use airports.dat record (f, 96)
           give sorted.dat ORG SQ
           sort fields (5, 30 CH, a)
SORT204I: ********************** ソート結果 **********
SORT205I: INPUT  ファイル 'airports.dat'
                 入力レコード                5402 件
                 使用レコード                5402 件
SORT206I: OUTPUT ファイル 'sorted.dat'
                 使用レコード                5402 件
                 出力レコード                5402 件
SORT399I: Micro Focus MFJSORT ユーティリティ終了
```

② クラシックデータファイルツールを起動して「sorted.dat」を開きます。

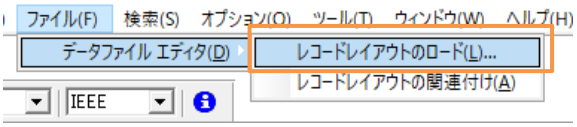
さきほど同様、[ファイル編成(O)] に “レコード順 – 固定長” を選択、[最大の長さ(X)] に “96” を指定して [OK] をクリックします。



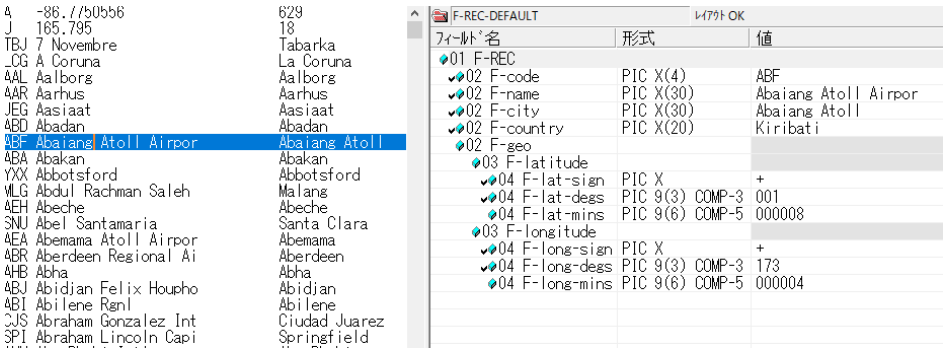
以下のダイアログが表示された場合は、[はい(Y)] をクリックします。



③ [ファイル(F)]メニュー > [データファイルエディタ(D)] > [レコードレイアウトのロード(L)…] を選択し、MFSORT フォルダ配下の「AIRPORTSEQ.STR」ファイルを指定します。



④ 空港名でソートされていることを確認します。



3.7. ファイル管理ユーティリティ REBUILD の確認

ファイル管理ユーティリティ REBUILD は、COBOL のファイル編成の変換、索引ファイルの索引再編成、破損した索引ファイルのリビルド機能を提供するコマンドラインのユーティリティです。本チュートリアルでは索引ファイルの再編成を行います。

1) ファイルの用意

① 解凍したフォルダに REBUILD が含まれていることを確認します。

2) rebuild コマンドによるインデックスの再編

① Windows のスタートメニューから [Micro Focus Visual COBOL] > [Visual COBOL コマンドプロンプト(32-bit)] を選択します。

② コマンドプロンプト上で、先ほど確認した REBUILD フォルダ配下まで移動します。

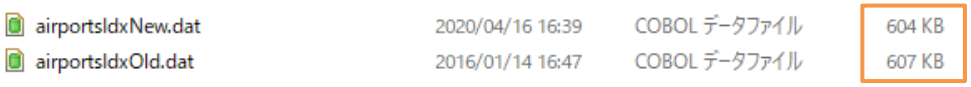
③ rebuild コマンドを実行します。コマンドラインに「rebuild airportsIdxOld.dat,airportsIdxNew.dat」とタイプしてリターンキーを押します。

3) 実行結果の確認

① 実行が終わるとメッセージが表示されます。

再構成が完了しました - レコード読み込み = 5383

再編成前のファイル airportsIdxOld.dat はレコードの追加、更新、削除を繰り返していたため、ファイルサイズが大きくなってしまっていました。rebuild コマンドにて、再編成が行われ、不要領域が解放されファイルサイズが縮小されたことが確認できます。



WHAT'S NEXT

- 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。

免責事項