Visual COBOL チュートリアル

COBOL 開発：ステップバイステップチュートリアル

Eclipse

1 目的

Visual COBOL for Eclipse は、高機能なオープンソースの IDE（統合開発環境）として広く普及する Eclipse 上で COBOL のアプリケーション開発を行うための製品です。 COBOL プログラマーが既存の COBOL 資産を Windows、UNIX/Linux といったオープン環境で活用するだけでなく、COBOL プログラミング経験のないプログラマーが初めて COBOL アプリケーション開発を行う場合に最適な製品です。

このドキュメントは、Visual COBOL for Eclipse を学ぶためのステップバイステップのチュートリアルです。

2 前提

- 本チュートリアルで使用したマシン OS ： Windows 11
- Visual COBOL 11.0 Patch Update 01 for Eclipse がすでにインストールされていること
  インストール手順は以下の FAQ サイトを参照してください。
- https://support.amc.rocketsoftware.co.jp/SupportInf/amc_faqpublic.aspx?VC01002
- プログラミングの基礎知識を有していること
- Windows の基本操作を理解していること

下記のリンクから事前にチュートリアル用のサンプル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。

サンプルプログラムのダウンロード

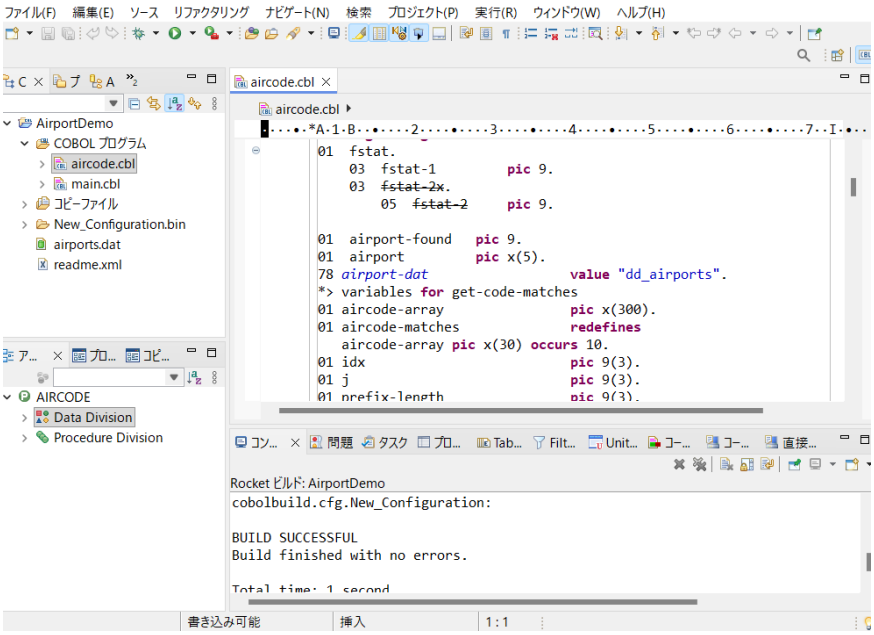
内容

3 チュートリアル

3.1 Eclipse の IDE に慣れてみよう

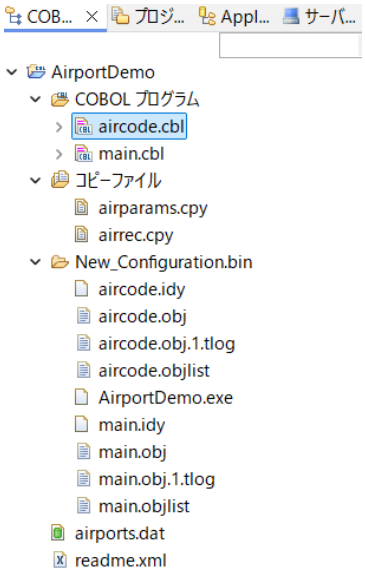
1) Windows のスタートメニューから [Rocket Visual COBOL] > [Visual COBOL for Eclipse] を選択します。

2) Eclipse を起動して任意の COBOL プロジェクトを作成すると以下のような画面が表示されます。

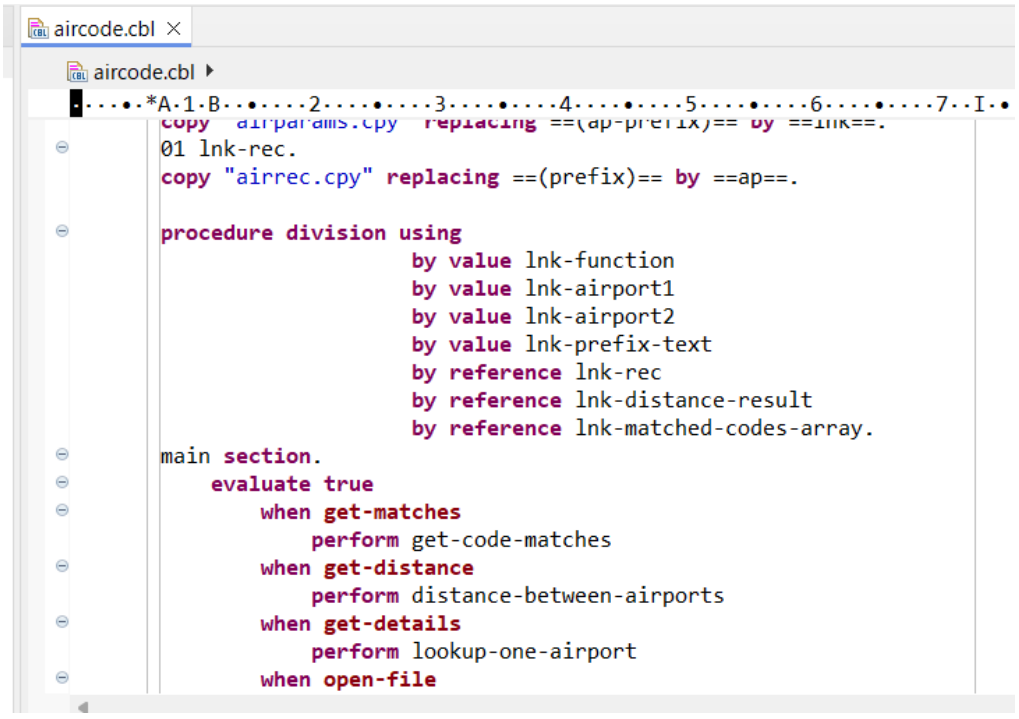


Eclipse の IDE は、ビュー、エディター、ツールバー、メニューバー、ステータスバー、パースペクティブから構成されるワークベンチというウィンドウ内で作業します。パースペクティブは、行いたい作業によって切り替えて利用するワークベンチのレイアウト（表示するビュー、メニュー、ツールバーの種類や場所）のことを指します。パースペクティブを切り替えることによって目的の作業に応じた構成要素を揃えることができ、各要素の配置もカスタマイズできます。

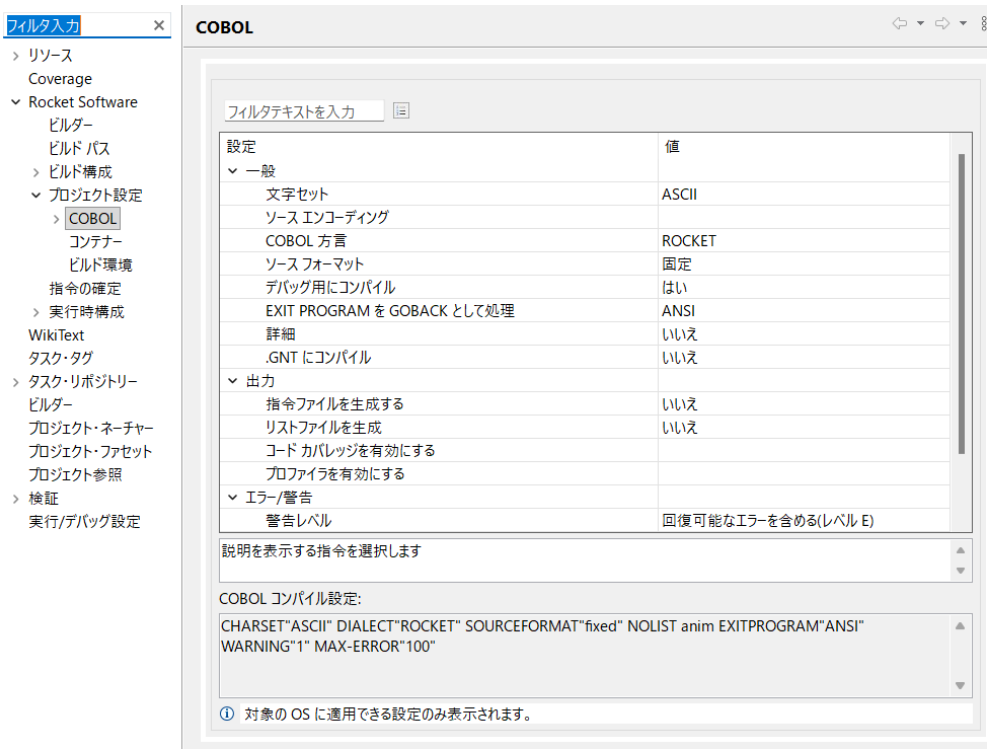
Eclipse のワークスペースとプロジェクトには、アプリケーションの作成に必要なビルドパス、データ接続、フォルダ、およびファイルを表す項目等が含まれています。 ワークスペースには複数のプロジェクトを含めることができ、プロジェクトには、通常、複数の項目が含まれます。COBOL エクスプローラーには、ワークスペースに紐づけられたプロジェクト、それらのプロジェクト内の項目が階層状に表示されます。COBOL エクスプローラー上で目的の要素を検索し、編集するファイルを開く、プロジェクトに新規ファイルを追加する、プロジェクトおよび項目のプロパティを表示するなどの操作を実行できます。パースペクティブによっては異なる名称のビューが同等の用途のために用意されています。例えば Java パースペクティブにはパッケージエクスプローラーという Java 開発時に必要な要素をフィルター表示するビューが紐つけられています。



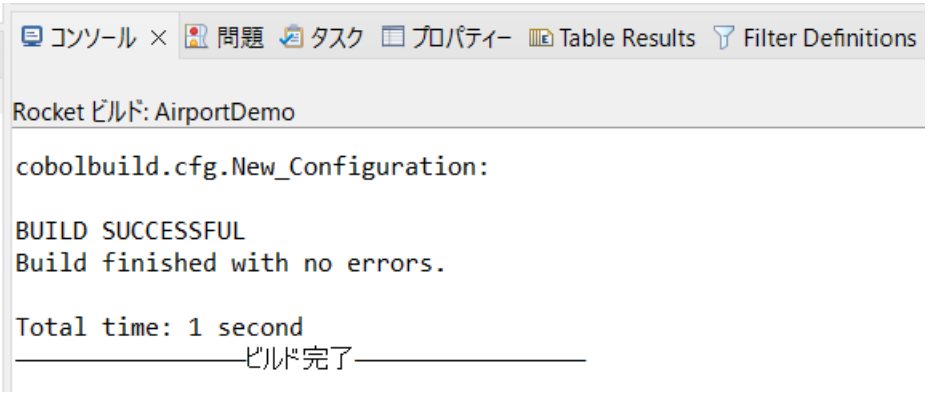
COBOL パースペクティブに紐づくエディターには、COBOL 予約語とデータ名や手続き名などの利用者語を色分け表示や、COBOL スニペットなど COBOL 言語固有の機能拡張が含まれます。ソースコードを入力するとバックグラウンドチェックを実行して、赤の波線でエラー箇所を強調表示します。 そのエラー箇所にマウスポインタを移動すればエラー内容の確認、定義への移動、他の参照検索などの操作が可能です。



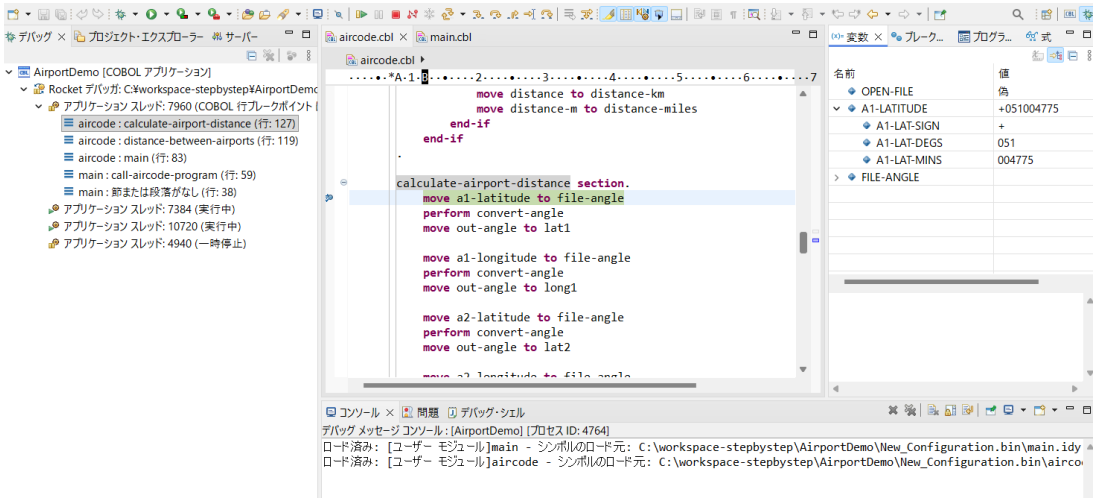
プロジェクト名を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、 [プロパティ(R)] を選択します。その後、表示される画面にて、 [Rocket Software] > [プロジェクト構成] > [COBOL] と遷移する画面では COBOL アプリケーションのビルド方法等を構成します。



コンソールビューにはビルド時のメッセージやアプリケーションのコンソール出力等が表示されます。問題ビューには、不正な構文、キーワードのスペルミス、型の不一致などのコンパイルエラーが表示されます。



ビルドしたアプリケーションは、実行時の論理エラーやセマンティックエラーなどの問題を検出して修正するために、デバッグ機能を利用します。 Eclipse のデバッガーは、コードのステップ実行や、様々な条件を設定したブレークポイントで実行を中断させ、変数ビューを使用してローカル変数やその他の関連データを調べることができます。



デバッグが完了したアプリケーションは、アプリケーションサーバ等との連携機能を利用して自動配備するか、ファイルを手動でコピーして、本番環境に配置します。なお、本番環境には COBOL Server が事前にインストールされている必要があります。

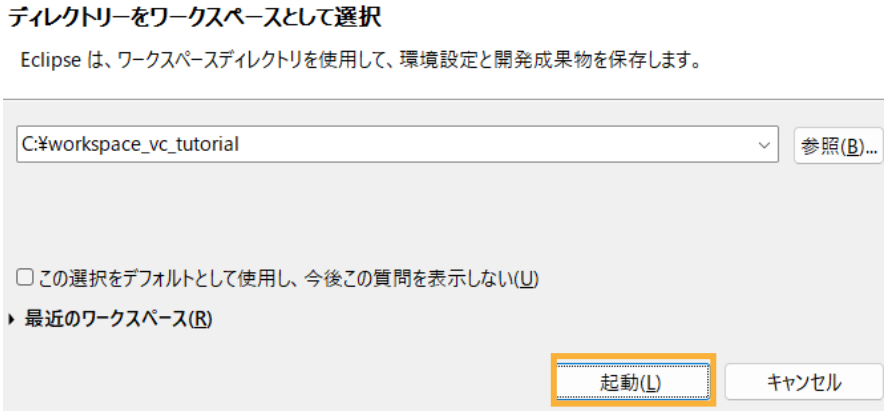
3.2 はじめての Visual COBOL プロジェクト

1) Visual COBOL for Eclipse を起動します。

Windows のスタートメニューから [Rocket Visual COBOL] > [Visual COBOL for Eclipse] を選択します。

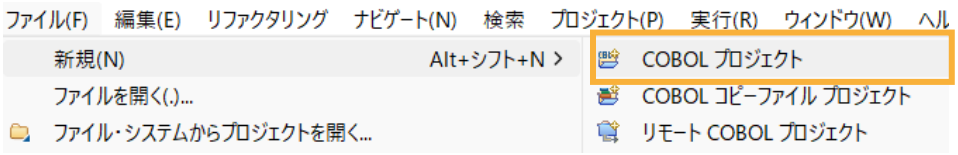
2) ワークスペースの指定

ワークスペースを保存する任意のフォルダを指定します。[参照(B)] をクリックしエクスプローラー経由で選択することも可能です。ワークスペースを選択後、[起動(L)] をクリックします。起動したら「ようこそ」は閉じてください。

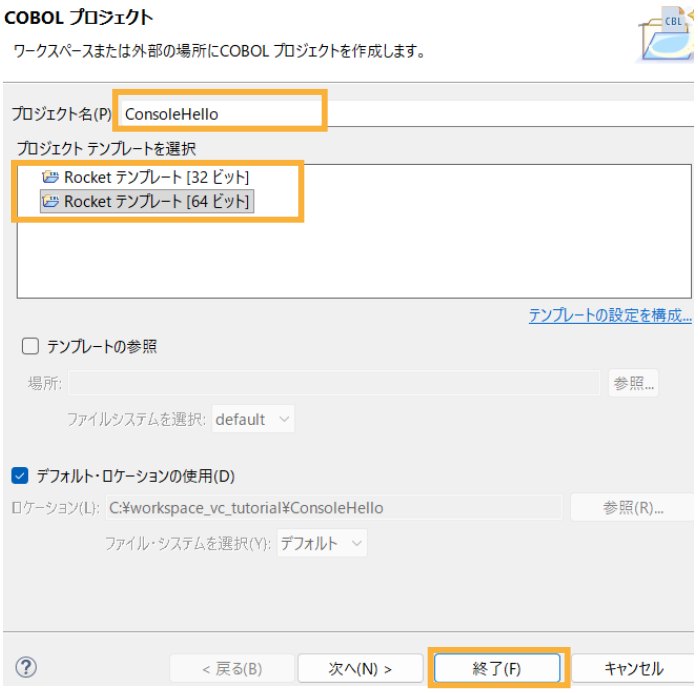


3) COBOL プロジェクトの作成

メニューより、[ファイル(F)] > [新規(N)] > [COBOL プロジェクト] を選択します。

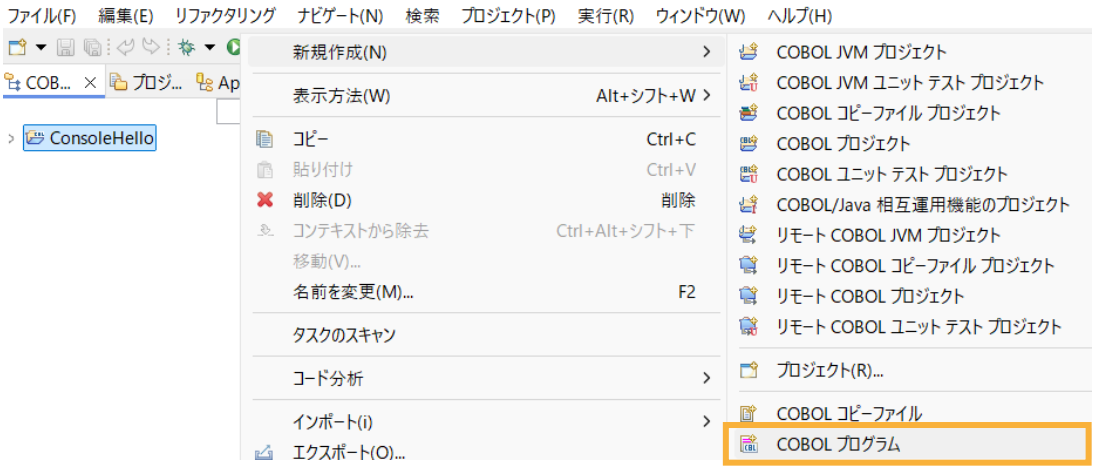


プロジェクト名に “ConsoleHello” を入力し、プロジェクトテンプレートを稼働環境に合わせて選択した上、[終了(F)] をクリックします。



4) COBOL プログラムの追加

COBOL エクスプローラービューにて ConsoleHello プロジェクトを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[新規作成(N)] > [COBOL プログラム] を選択します。

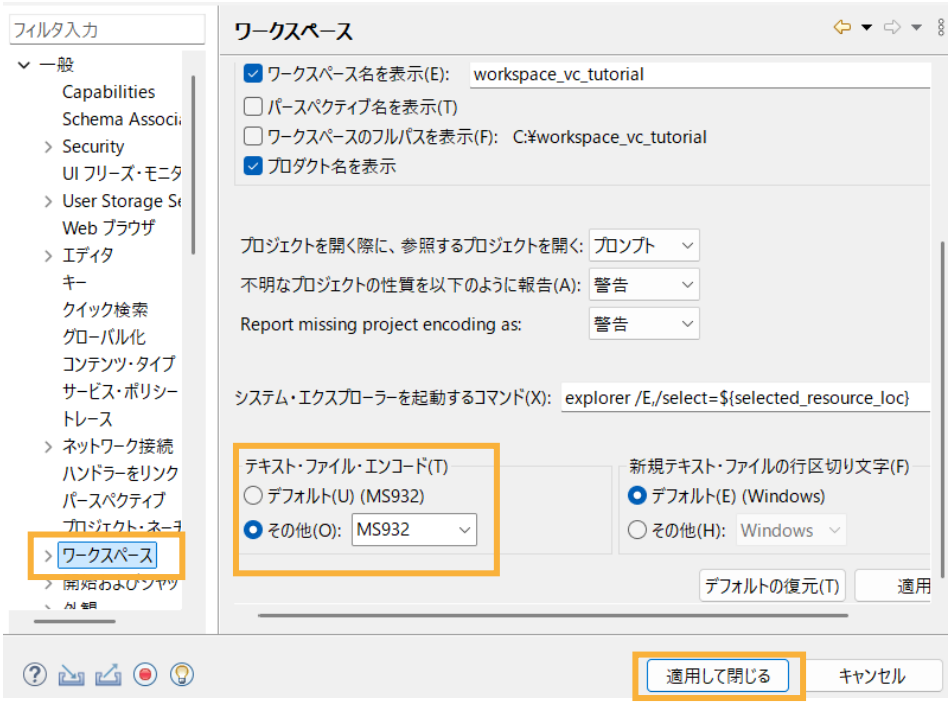


続くダイアログでは、そのまま [終了(F)] をクリックします。

5) 文字コードの指定

Shift_JIS を指定して日本語を表示する場合、文字コードの指定を明確に行う必要があります。

[Window(W)] > [設定(P)] より [一般] > [ワークスペース] とナビゲートし、テキストファイルエンコードに “MS932” が選択されているかを確認します。異なる値が設定されている場合は、”MS932” を選択して、 [適用して閉じる] をクリックします。

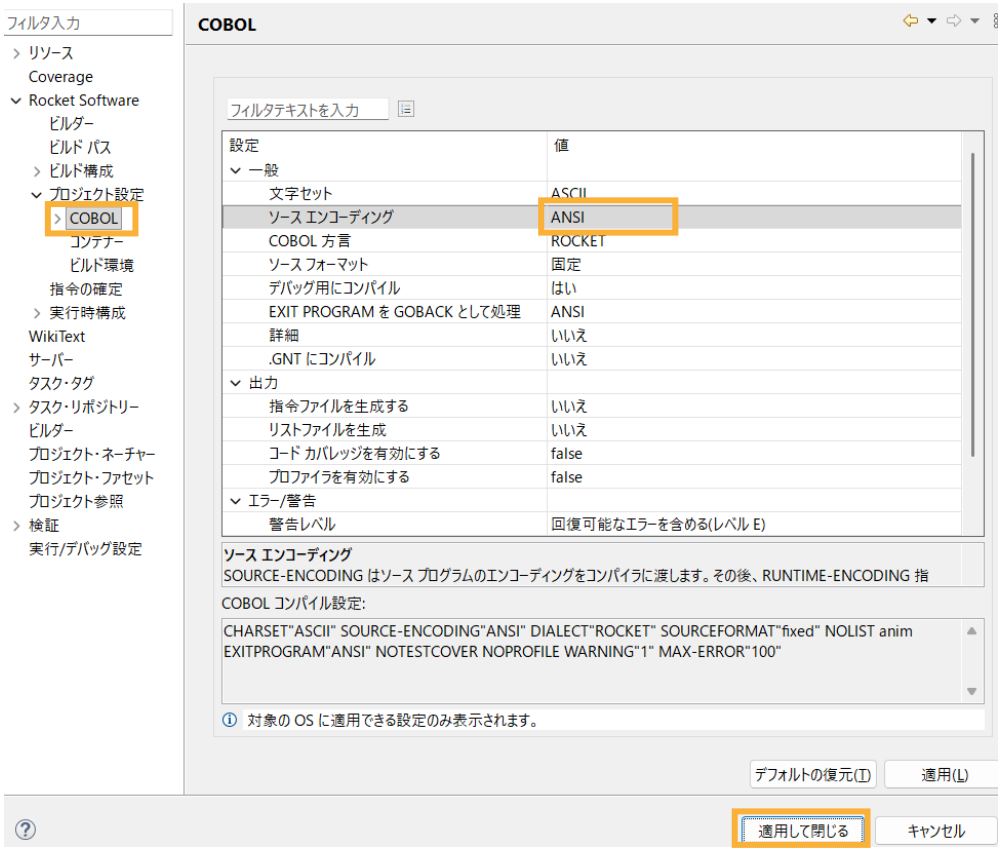


補足）

環境によっては、Windows-31J と表示されることがあります。その場合は、MS932 を Windows-31J に読み替えてください。

[Preference Recorder] ダイアログが表示された場合は、[キャンセル] をクリックしてください。

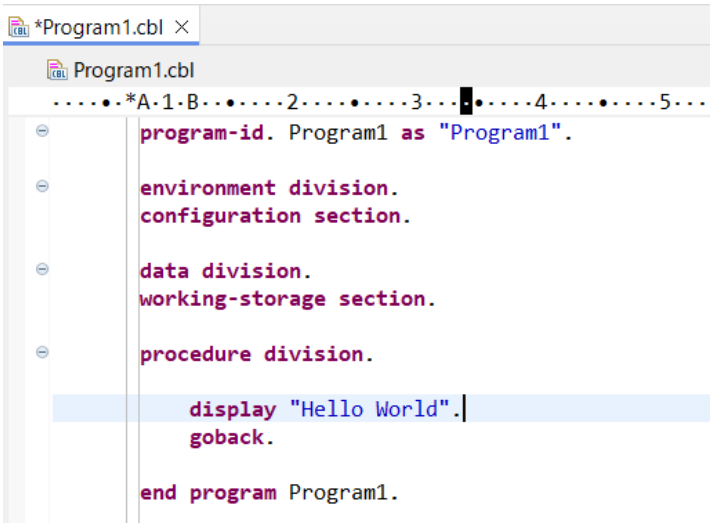
次に作成した COBOL プロジェクトを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[プロパティ] を選択します。[Rocket Software] > [プロジェクト構成] > [COBOL]とナビゲートし、 [一般] > [ソース エンコーディング]を ”UTF-8” から ”ANSI” に変更し、[適用して閉じる] をクリックします。



6) エディターで COBOL のソースコードを入力

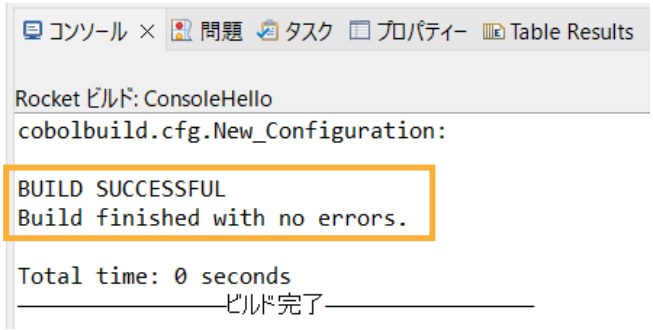
COBOL プロジェクトにて COBOL プログラムを新規に作成するとプログラムを構成する見出し部（identification division)、環境部（environment division)、データ部（data division)、手続き部（procedure division）や program-id 段落が埋め込まれたかたちで生成されます。今回は “Hello World” を表示して終了するプログラムなので、手続き部３行目の goback 文の手前に「display "Hello World".」を追加します。

なお、COBOL 正書法ではエディタービュー左右にある線で区切られた領域を特別な領域として利用するので、通常のソースコードはこれを避けて入力します。



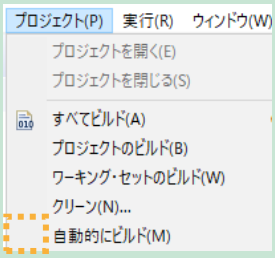
7) COBOL アプリケーションのビルド

終止符（ピリオド）を含めてスペルミスがなければ、エディタービュー上の Program1.cbl をアクティブにした状態で、メニューより、[ファイル(F)] > [保存(S)] または Ctrl + S キーで保存します。これにより自動的にコンパイラが起動されビルド処理が始まります。コンソールビューに正常にビルドできた旨のメッセージが出力されていることを確認します。



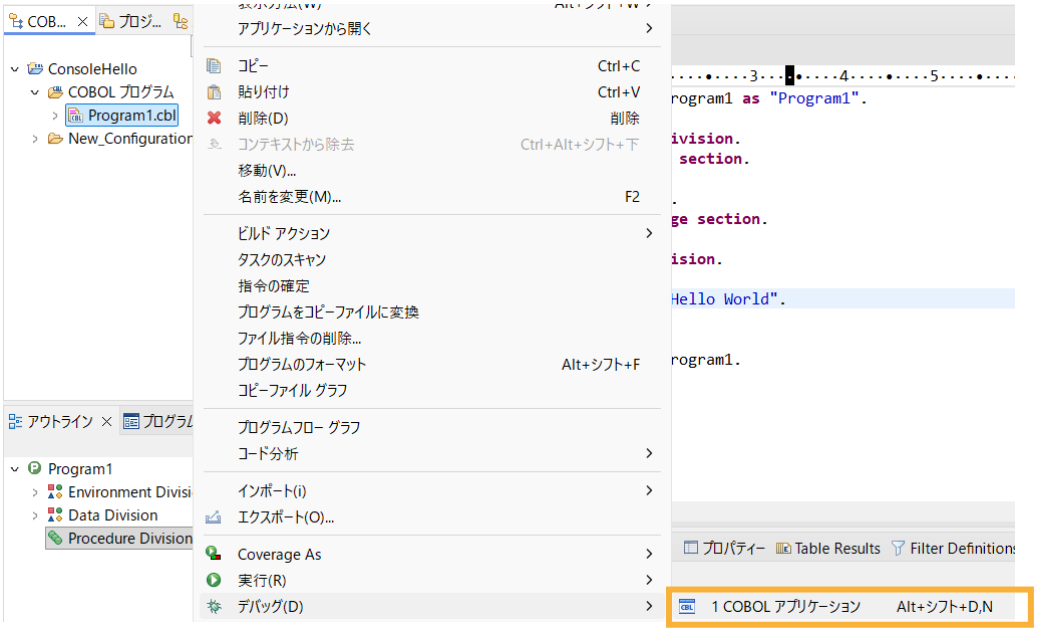
補足）

Visual COBOL for Eclipse では、上記のように保存したタイミングでビルドが自動実行されるようになっています。自動実行を抑止する場合は、メニューより、[プロジェクト(P)] > [自動的にビルド(M)] を選択し、未チェック状態にします。

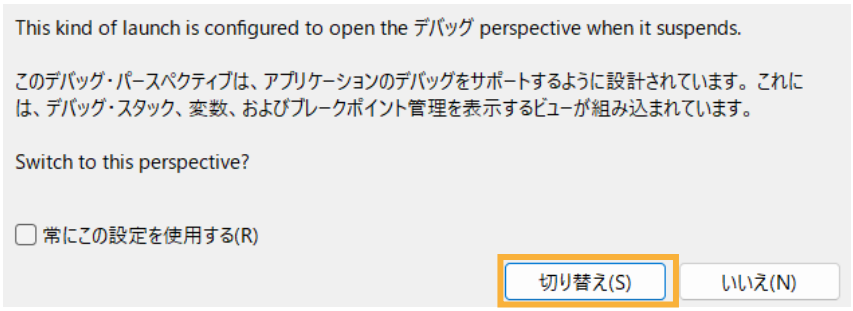


8) COBOL アプリケーションのデバッグ実行

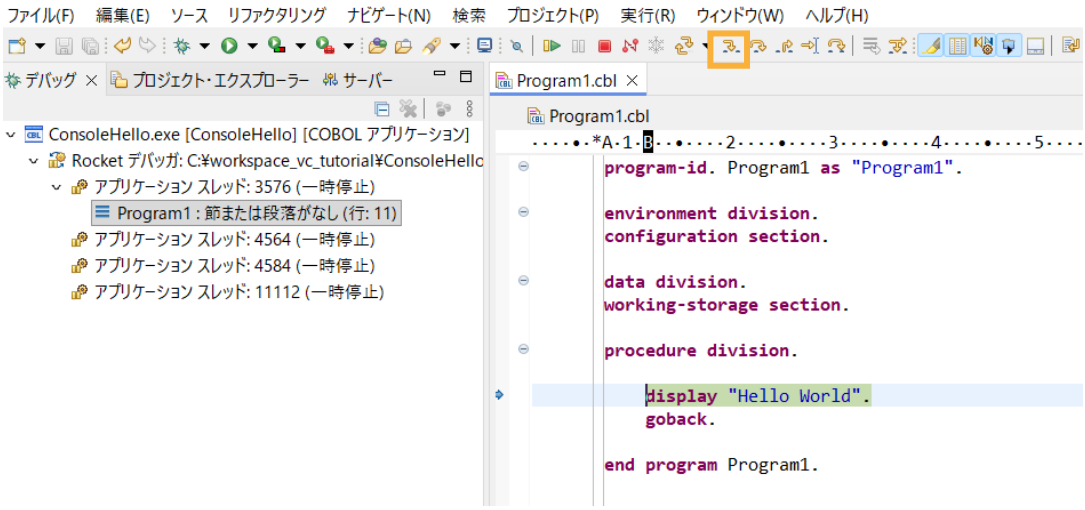
COBOL エクスプローラー上の Program1.cbl を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[デバッグ(D)] > [COBOL アプリケーション] を選択します。



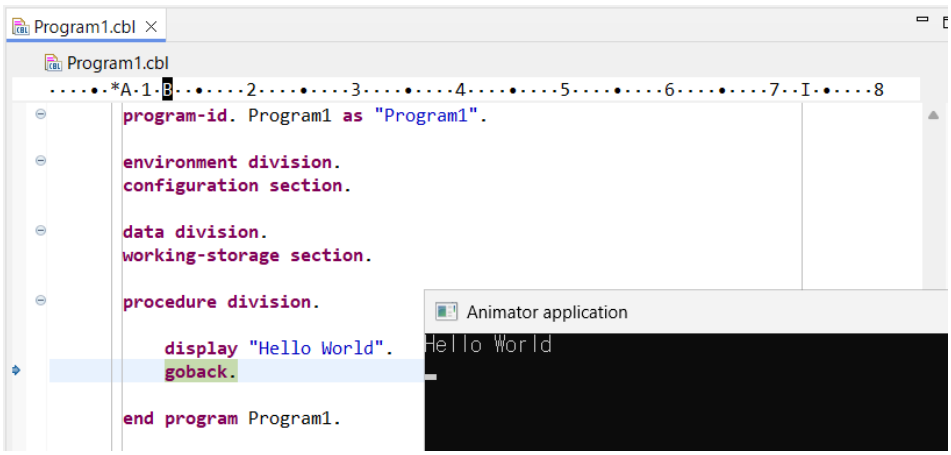
パースペクティブ切り替えの確認ダイアログが表示された場合、[切り替え(S)] をクリックします。なお、「常にこの設定を使用する(R)」にチェックをすることで、今後、この問合せを抑止できます。



デバッグパースペクティブに切り替わりましたら、DISPLAY 文を実行する手前でステップが一時停止しています。[ステップイン] アイコンを一回クリックし DISPLAY 文を実行します。



Animator application 画面に「Hello World」が表示されたことを確認できましたら、[ステップイン] アイコンを再度クリックしデバッグを終了します。

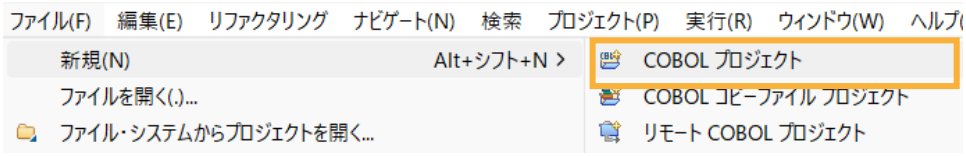


3.3 ファイルの入出力

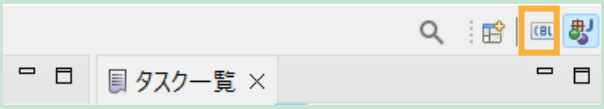
エクセルやメモ帳で作成した CSV ファイルを読み込んで、固定長順編成ファイルを作成する COBOL アプリケーションを Visual COBOL for Eclipse で作成します。

1) Visual COBOL for Eclipse を終了した場合は起動します。
Windows のスタートメニューから [Rocket Visual COBOL] > [Visual COBOL for Eclipse] を選択します。
ワークスペースは 3.2 で使用したものでも、新規のものでも構いません。（本チュートリアルでは、同一のワークスペースを使用します。）ワークスペースを選択後、[起動(L)] をクリックします。

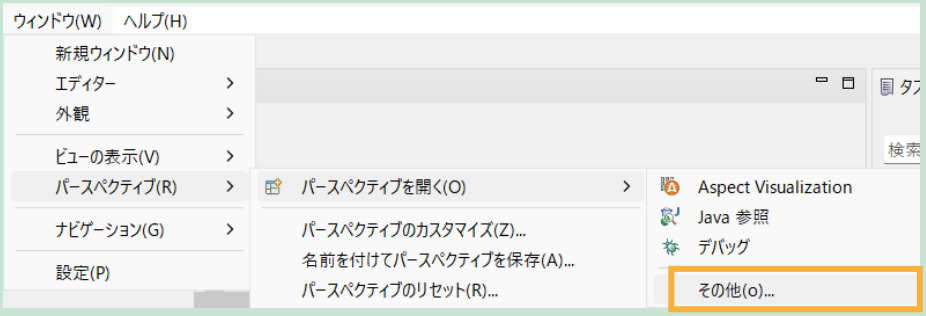
2) COBOL プロジェクトの作成
メニューより、[ファイル(F)] > [新規(N)] > [COBOL プロジェクト] を選択します。

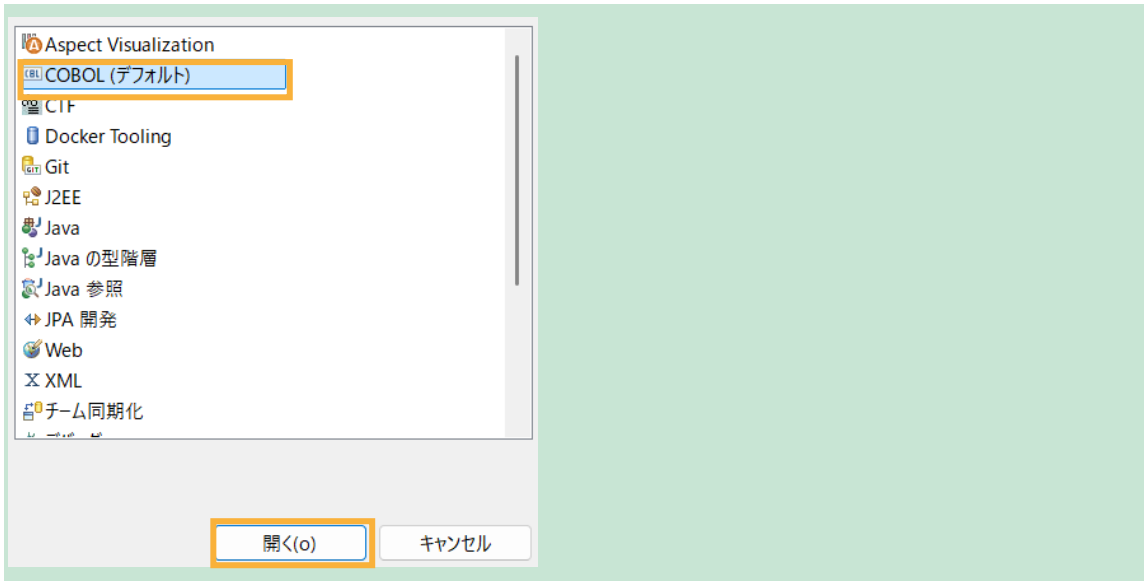


補足）
新規メニューに表示される項目は、現在、アクティブのパースペクティブにより変更されます。例えば、COBOL パースペクティブがアクティブとなっている場合、上記 COBOL プロジェクトは表示されますが、Java パースペクティブがアクティブの場合、異なる内容が表示されます。この場合は、Eclipse 画面の右上よりパースペクティブを切り替えます。
以下は、Java パースペクティブがアクティブな状態です。COBOL パースペクティブアイコンをクリックすることで、パースペクティブの切り替えできます。



もしくは、メニューより、[ウィンドウ(W)] > [パースペクティブ(R)] > [パースペクティブを開く(O)] > [その他(O)] を選択した上で、表示されるダイアログ上にて、「COBOL (デフォルト)」を選択し、[開く(o)] をクリックします。

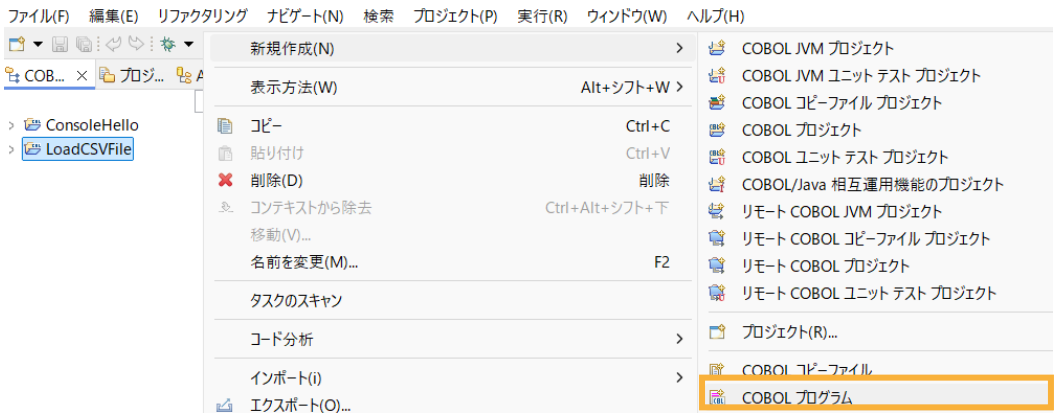




プロジェクト名に “LoadCSVFile” を入力し、プロジェクトテンプレートを稼働環境に合わせて選択した上、[終了(F)] をクリックします。

3) COBOL プログラムの追加

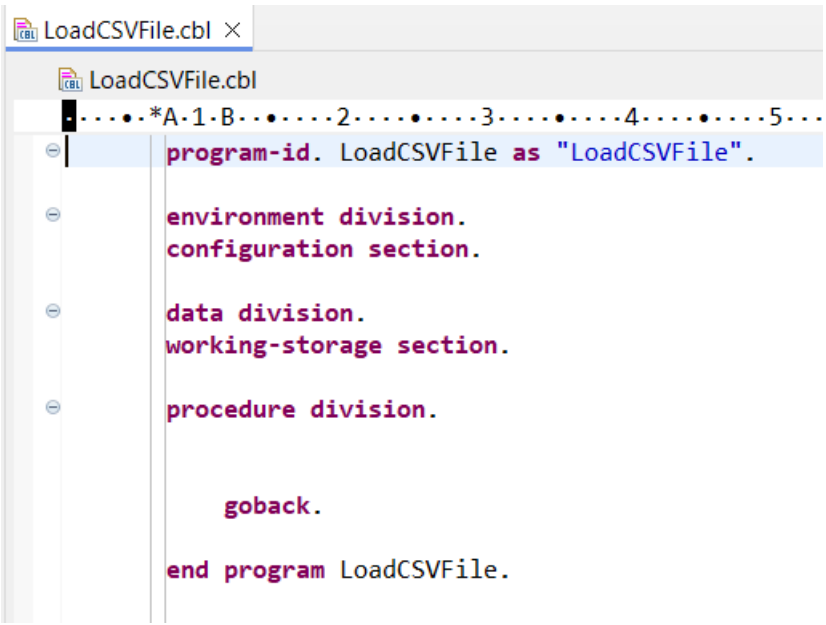
COBOL エクスプローラービューにて LoadCSVFile プロジェクトを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[新規作成(N)] > [COBOL プログラム] を選択します。



プログラム名に “LoadCSVFile.cbl” を入力し、[終了(F)] をクリックします。

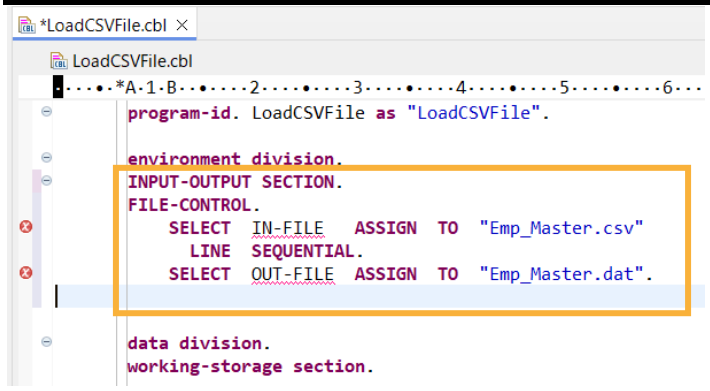
4) エディターで COBOL のソースコードを入力

COBOL ソースファイルを新規に作成した直後の段階ではコンソールアプリケーションのひな形が埋め込まれています。ここでは、環境部 (environment division)、データ部 (data division)、手続き部 (procedure division) を書き換えます。



まず、環境部の構成節（configuration section）を削除し、以下の入出力節（input-output section）を追加します。
まだ、データ部のファイル定義が未入力なので IN-FILE と OUT-FILE がエラーとなりますが、ここでは無視して構いません。

```
    INPUT-OUTPUT SECTION.
    FILE-CONTROL.
        SELECT  IN-FILE   ASSIGN  TO  "Emp_Master.csv"
          LINE  SEQUENTIAL.
        SELECT  OUT-FILE  ASSIGN  TO  "Emp_Master.dat".
```



次に、データ部の作業場所節（working-storage section）を削除し、以下のファイル節（FILE SECTION）を追加します。 なお、データ部のファイル定義を入力したので、環境部のエラーは無くなります。

```
FILE SECTION.
FD  IN-FILE.
01  IN-REC              PIC  X(50).
FD  OUT-FILE.
01  OUT-REC.
    05  OUT-EMPNO       PIC  9(8).
    05  FILLER          PIC  X.
    05  OUT-JNAME1      PIC  X(10).
    05  OUT-JNAME2      PIC  X(10).
    05  OUT-NAME1       PIC  X(5).
    05  OUT-NAME2       PIC  X(5).
    05  OUT-GENDER      PIC  X.
    05  FILLER          PIC  X.
    05  OUT-DIV         PIC  X(10).
    05  OUT-EMPDATE     PIC  9(8).
    05  FILLER          PIC  X.
```

*LoadCSVFile.cbl

LoadCSVFile.cbl

```
environment division.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT  IN-FILE   ASSIGN  TO  "Emp_Master.csv"
      LINE  SEQUENTIAL.
    SELECT  OUT-FILE  ASSIGN  TO  "Emp_Master.dat".


data division.
FILE SECTION.
FD  IN-FILE.
01  IN-REC              PIC  X(50).
FD  OUT-FILE.
01  OUT-REC.
    05  OUT-EMPNO       PIC  9(8).
    05  FILLER          PIC  X.
    05  OUT-JNAME1      PIC  X(10).
    05  OUT-JNAME2      PIC  X(10).
    05  OUT-NAME1       PIC  X(5).
    05  OUT-NAME2       PIC  X(5).
    05  OUT-GENDER      PIC  X.
    05  FILLER          PIC  X.
    05  OUT-DIV         PIC  X(10).
    05  OUT-EMPDATE     PIC  9(8).
    05  FILLER          PIC  X.
```

最後に、手続き部の goback 文を削除し、以下の手続き文を追加します。

```
PROC1.
    OPEN   INPUT    IN-FILE.
    OPEN   OUTPUT   OUT-FILE.

PROC2.
    READ   IN-FILE   AT END    GO TO PROC9.
    INITIALIZE   OUT-REC.
    UNSTRING   IN-REC   DELIMITED   BY   ","
      INTO   OUT-EMPNO
             OUT-JNAME1
             OUT-JNAME2
             OUT-NAME1
             OUT-NAME2
             OUT-GENDER
             OUT-DIV
             OUT-EMPDATE
    END-UNSTRING.
    WRITE   OUT-REC.
    GO   TO   PROC2.

PROC9.
    CLOSE   IN-FILE   OUT-FILE.
    STOP   RUN.
```

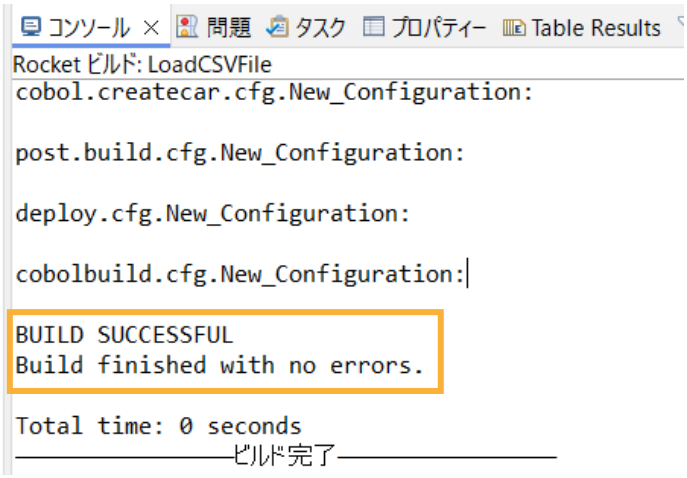
*LoadCSVFile.cbl

LoadCSVFile.cbl

```
      *A-1-B--+----2----+----3----+----4----+----5----+----6----+
       procedure division.

       PROC1.
           OPEN  INPUT   IN-FILE.
           OPEN  OUTPUT  OUT-FILE.

       PROC2.
           READ  IN-FILE  AT END   GO TO PROC9.
           INITIALIZE  OUT-REC.
           UNSTRING  IN-REC  DELIMITED  BY  ","
             INTO  OUT-EMPNO
                   OUT-JNAME1
                   OUT-JNAME2
                   OUT-NAME1
                   OUT-NAME2
                   OUT-GENDER
                   OUT-DIV
                   OUT-EMPDATE
           END-UNSTRING.
           WRITE  OUT-REC.
           GO  TO  PROC2.

       PROC9.
           CLOSE  IN-FILE  OUT-FILE.
           STOP  RUN.
```

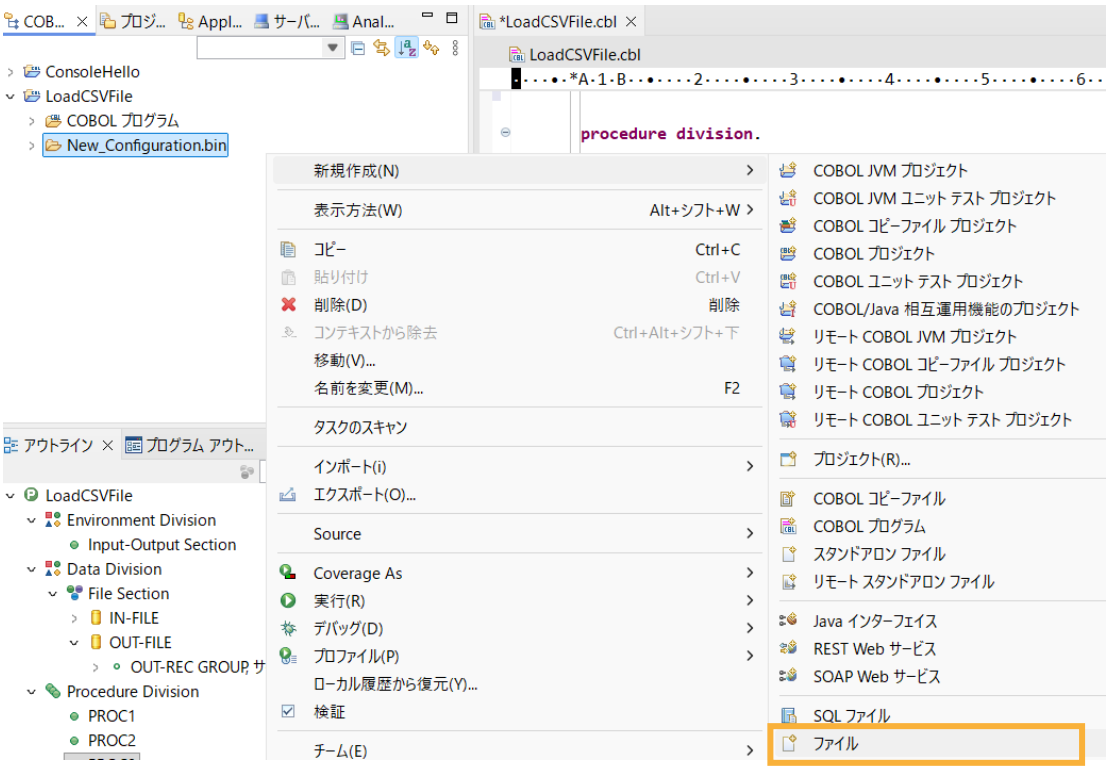
5) COBOL アプリケーションのビルド

終止符（ピリオド）を含めてエラーがなければ、LoadCSVFile.cbl への変更を、メニューより、[ファイル(F)] > [保存(S)] または Ctrl + S キーで保存します。自動的にビルドが実行され、コンソールビューに正常にビルドできた旨のメッセージが出力されていることを確認します。

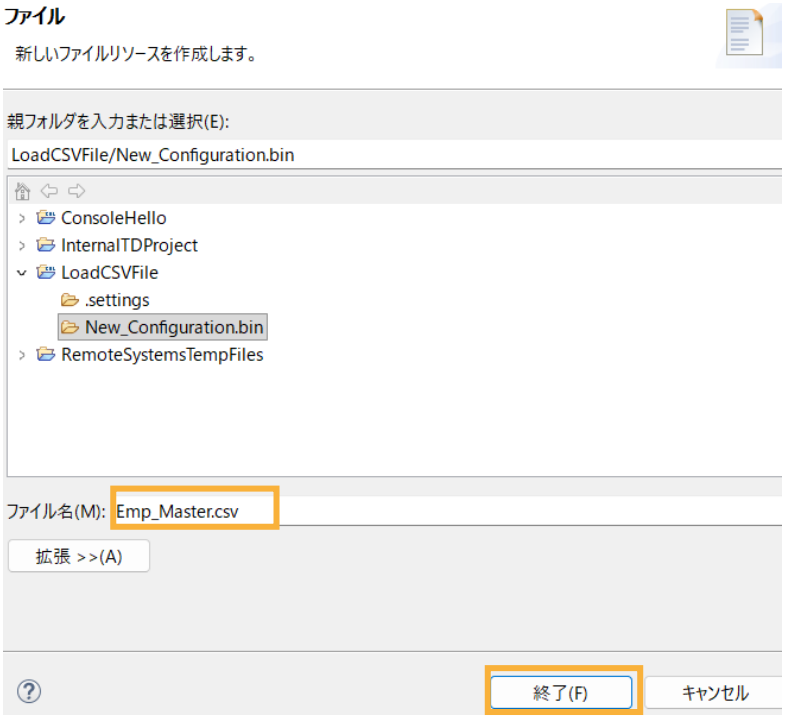


6) CSV ファイルの作成

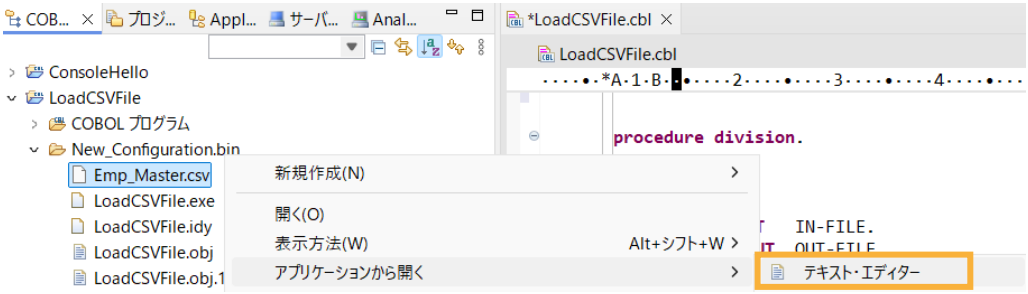
LoadCSVFile プロジェクト配下の New_Configuration.bin フォルダを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[新規作成(N)] > [ファイル] を選択します。



ファイル名に “Emp_Master.csv” を入力し、[終了(F)] をクリックします。



Excel など Windows でデフォルト設定されたアプリケーションで CSV ファイルが自動的に起動した場合は、そのまま終了した後、New_Configuration.bin 配下の Emp_Master.csv を選択した状態で、マウスの右クリックにてコンテクストメニューを開き、[アプリケーションから開く] > [テキスト・エディター] を選択してください。



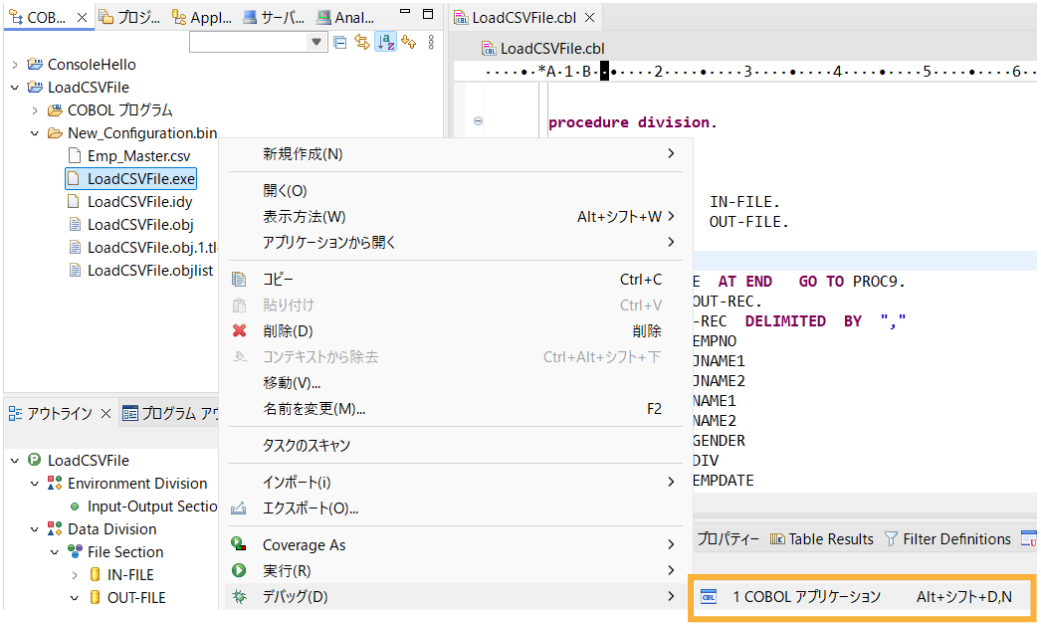
エディター上で、以下のデータを入力し、保存した上で、エディターを閉じます。

```
11111113,佐藤,隆,ｻﾄｳ,ﾀｶｼ,M,営業部,19980401,0
22222226,鈴木,尚之,ｽｽﾞｷ,ﾅｵﾕｷ,M,技術部,19981015,0
33333339,田中,直美,ﾀﾅｶ,ﾅｵﾐ,F,総務部,19990401,0
44444442,山田,洋一,ﾔﾏﾀﾞ,ﾖｳｲﾁ,M,営業部,20000701,0
55555555,伊藤,弘子,ｲﾄｳ,ﾋﾛｺ,F,技術部,20010401,0
66666668,木村,貴弘,ｷﾑﾗ,ﾀｶﾋﾛ,M,営業部,20021220,0
77777771,中村,慎司,ﾅｶﾑﾗ,ｼﾝｼﾞ,M,技術部,20030401,0
88888884,橋本,悦子,ﾊｼﾓﾄ,ｴﾂｺ,F,総務部,20040805,0
99999997,三井,薫,ﾐﾂｲ,ｶｵﾙ,F,営業部,20050401,0
```

テキスト・エディターを選択後に、編集するかを確認するダイアログが表示された場合は、[はい(Y)] をクリックしてください。

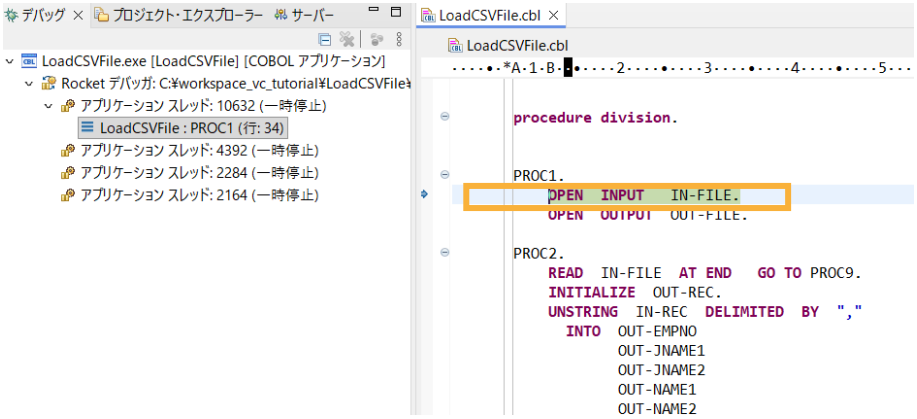
7) COBOL アプリケーションのデバッグ実行

LoadCSVFile プロジェクトの New_Configuration.bin フォルダ配下にある LoadCSVFile.exe を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[デバッグ(D)] > [COBOL アプリケーション] を選択します。



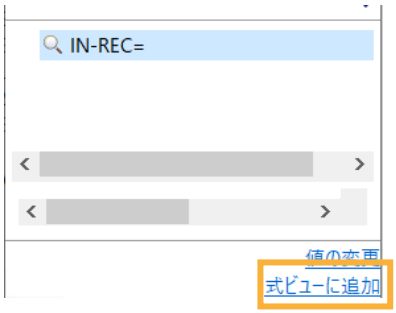
パースペクティブ切り替えの確認ダイアログが表示された場合、[切り替え(S)] をクリックします。

デバッガーは手続き部の最初の COBOL 文である open 文の処理前に一時停止している状態となっています。



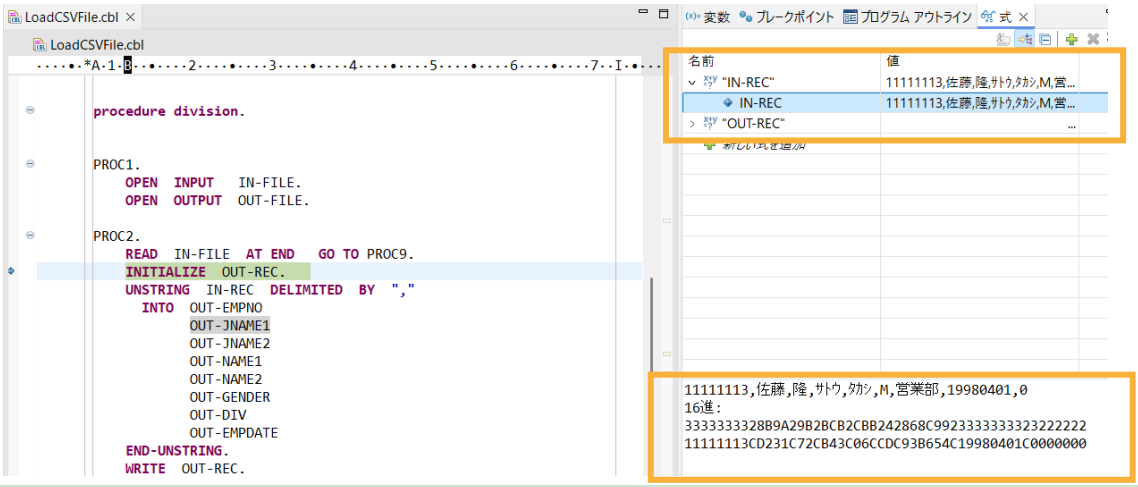
入力ファイルから読み込んだレコードの内容を確認するため、IN-REC に格納される値の変遷を追います。UNSTRING 文の IN-REC を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[項目を検査] を選択します。

[式ビューに追加] をクリックします。



出力ファイルに書き出すレコードの内容もトレースするため、INITIALIZE 文の OUT-REC を同様の手順で式ビューに追加します。

F5 キー（ステップイン）を 3 回押すと、デバッガーは READ 文実行後、処理を中断します。式ビューの IN-REC の値には CSV ファイルから読み込んだ最初のレコードが表示されます。

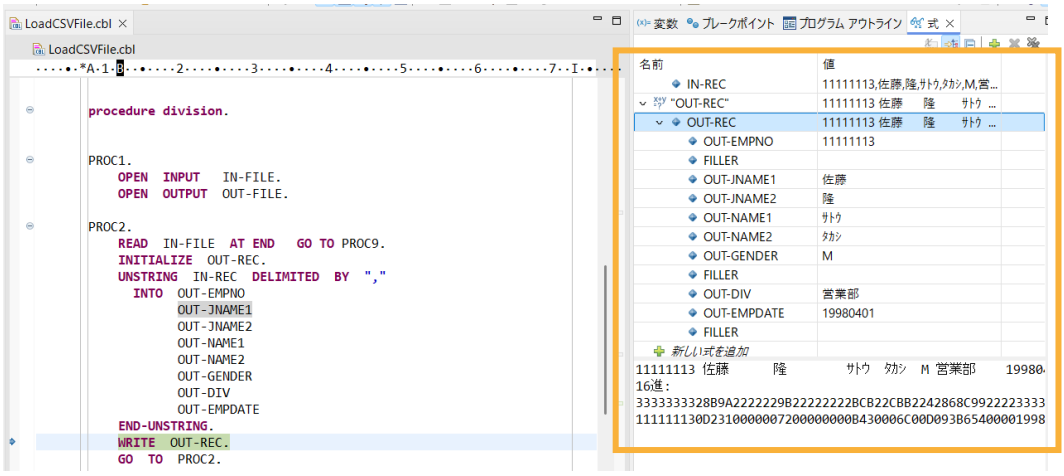


補足）

ステップインには、F5 キーがホットキーとして割り当てられているため、F5 キーを押すことでステップイン指示が可能です。

メニューからは、[実行(R)] > [ステップイン(I)] を選択します。

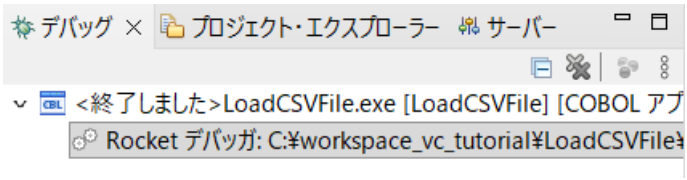
さらに F5 キーを 2 回押すと、デバッガーは UNSTRING 文を実行後、処理を中断します。式ビューの OUT-REC の値には出力ファイルへ書き出す最初のレコードが表示されます。



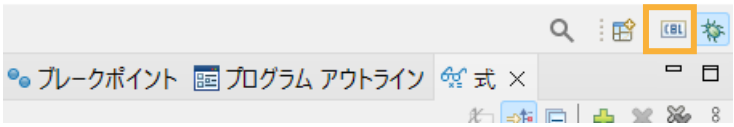
さらに F5 キーを 4 回押すと、デバッガーは INITIALIZE 文を実行後、処理を中断します。

ウォッチ式の IN-REC の値には CSV ファイルから読み込んだ 2 番目のレコードが表示され、OUT-REC の値は INITIALIZE 文で初期化されています。

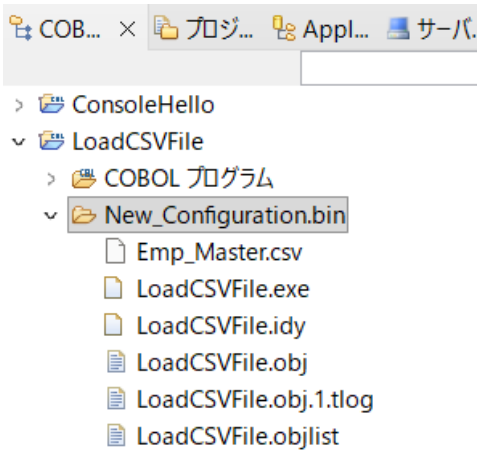
F8(再開) キーを打鍵するか CSV ファイルからすべてのレコードを読み込むまで F5 キーを押すと、デバッガーは終了します。



ワークスペース右上にて COBOL アイコン を選択し、COBOL パースペクティブに戻ります。

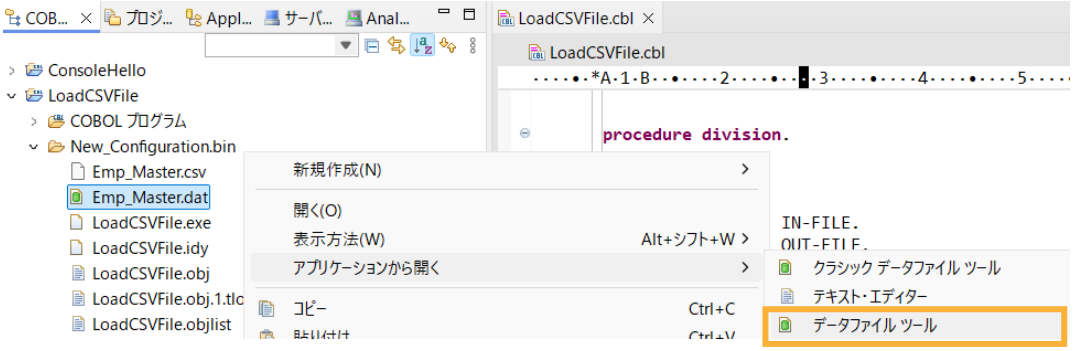


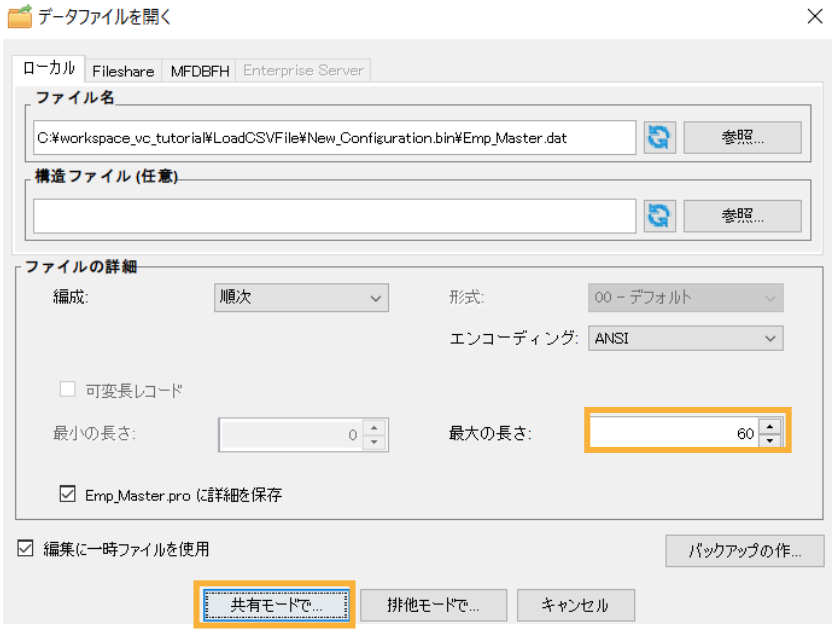
COBOL エクスプローラービュー上で LoadCSVFile プロジェクトの New_Configuration.bin フォルダを選択した状態で、F5 を押すか、[ファイル(F)] > [更新(F)] を選択して情報を更新します。



Emp_Master.dat が作成されていることを確認します。

Emp_Master.csv を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[アプリケーションから開く] > [データファイルツール] を選択します。Visual COBOL に付属する データファイルエディタが起動します。「最大の長さ」欄には “60” を指定して、[共有モードで開く] をクリックします。



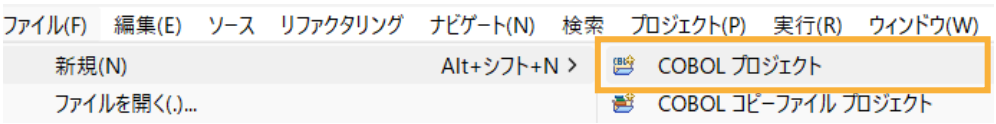


社員９名分のデータが正常に固定長順編成ファイルに書き込まれていることを確認します。

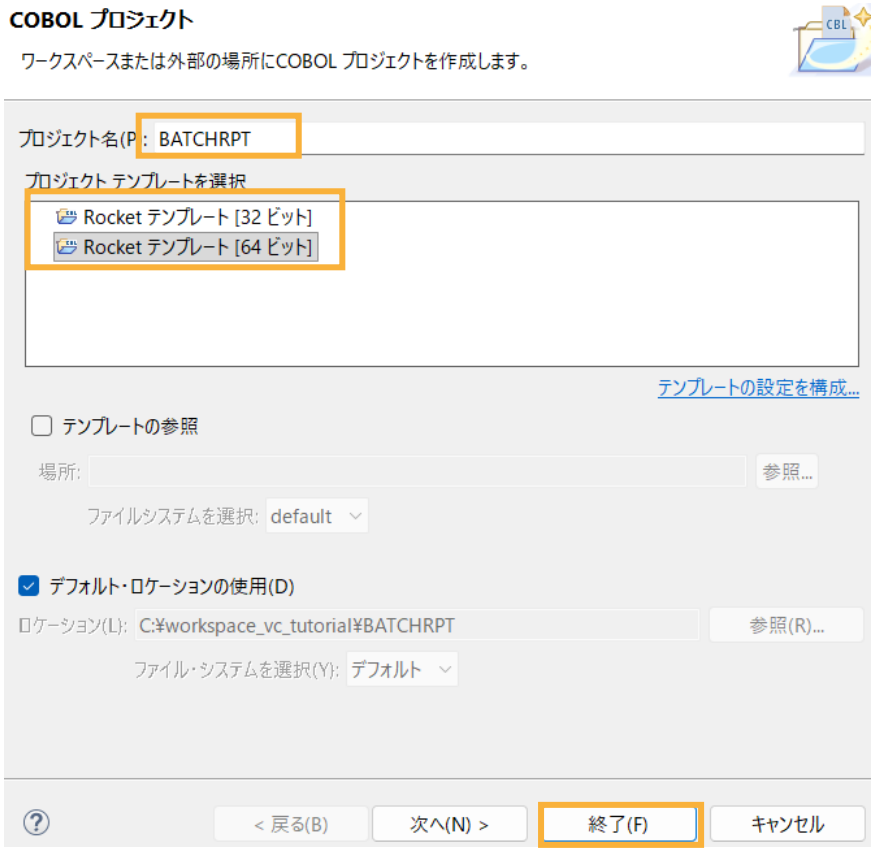
続いて、この固定長順編成ファイルを読み込んでレポートファイルを作成するバッチアプリケーションを作成します。

8) COBOL プロジェクトの作成

メニューより、[ファイル(F)] > [新規(N)] > [COBOL プロジェクト] を選択します。



プロジェクト名に “BATCHRPT” を入力し、プロジェクトテンプレートを稼働環境に合わせて選択した上、[終了(F)] をクリックします。

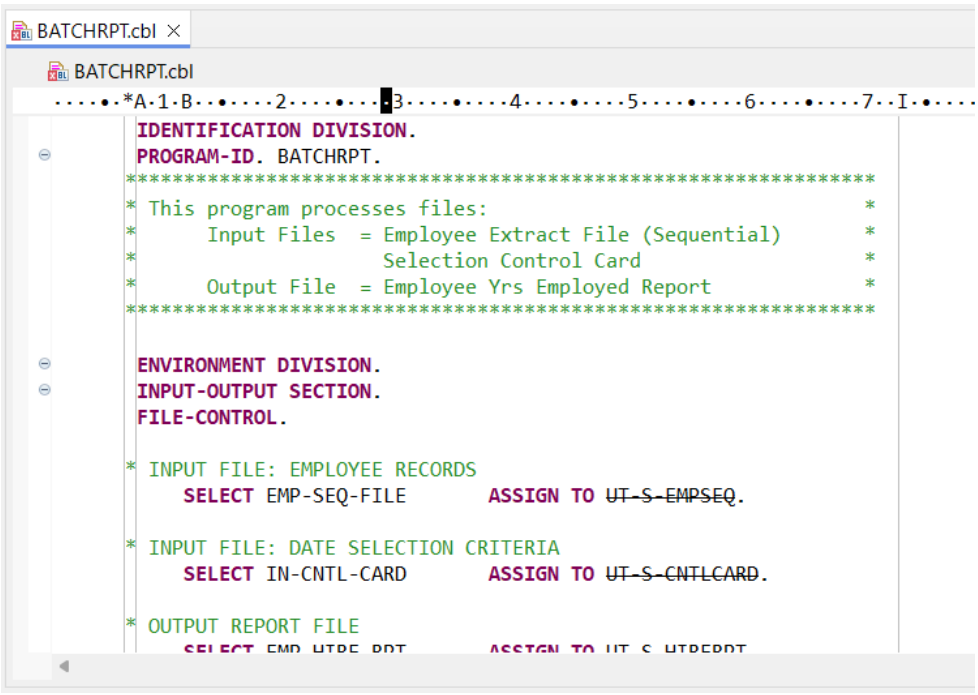


9) COBOL プログラムの追加

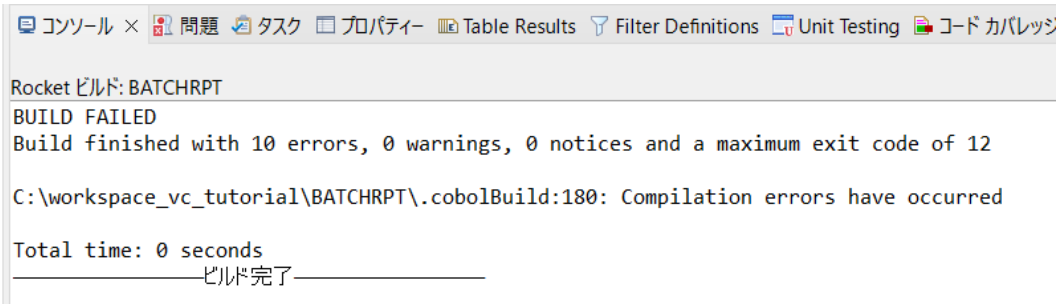
COBOL エクスプローラービューにて BATCHRPT プロジェクトを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[新規作成(N)] > [COBOL プログラム] を選択します。

ファイル名を BATCHRPT.cbl" として、[終了(F)] をクリックします。

BATCHRPT.cbl の中身を、サンプルプログラム BATCHRPT.cbl の内容で上書き保存します。



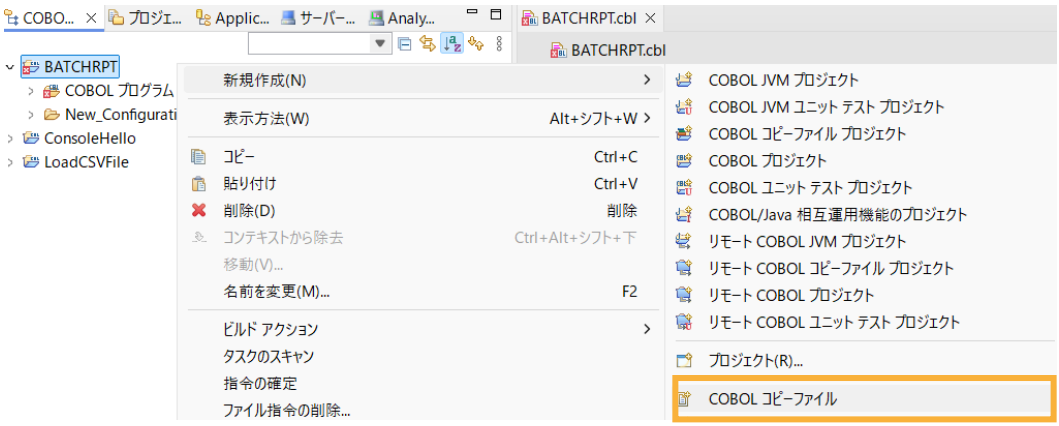
自動ビルドが実行された結果、エラーが報告されますが、これは、参照しているコピーブックが存在しないためです。



次の手順にて、コピーブックを作成するため、ここでは無視して構いません。

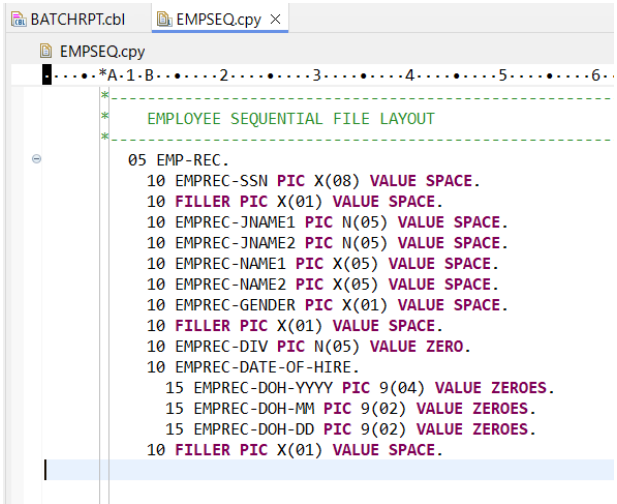
10) コピーブックの作成

COBOL エクスプローラービューにて BATCHRPT プロジェクトを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[新規作成(N)] > [COBOL コピーファイル] を選択します。

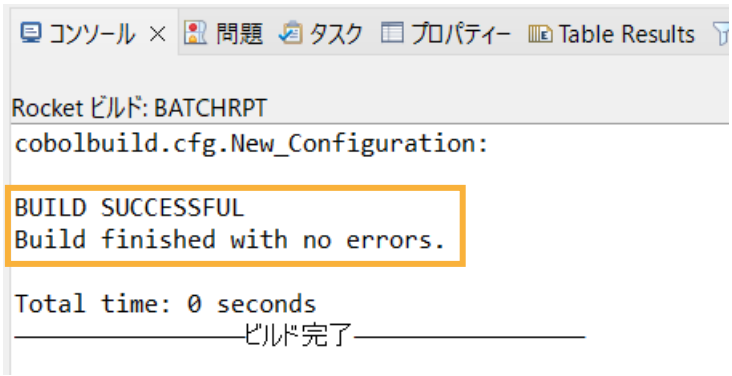


新規ファイル名に “EMPSEQ.cpy” を入力し、[終了(F)] をクリックします。

作成された EMPSEQ.cpy を、サンプルプログラム EMPSEQ.cpy の内容で上書き保存します

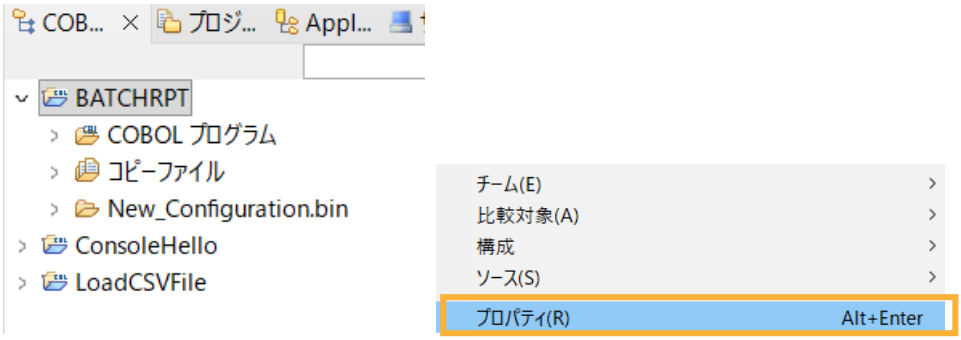


自動的にビルドが実行され、正常にビルドが行われたことを確認します。



11) プロジェクト設定、コンパイラ指令の追加

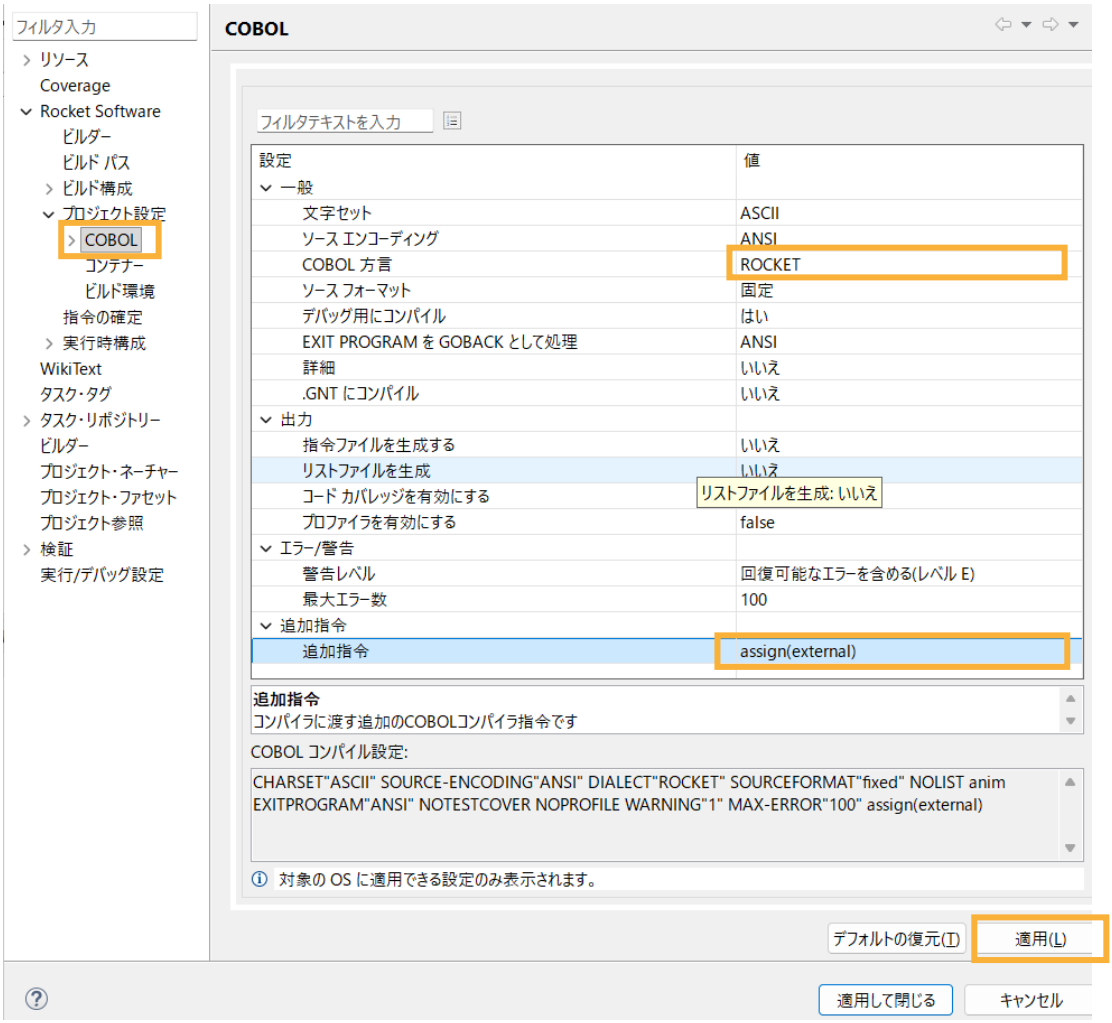
COBOL エクスプローラービューにて BATCHRPT プロジェクトを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[プロパティ(R)] を選択します。



左側のツリーより、[Rocket Software] > [プロジェクト設定] > [COBOL] を選択します。以下の設定を行ったうえで、[適用(L)] をクリックします。

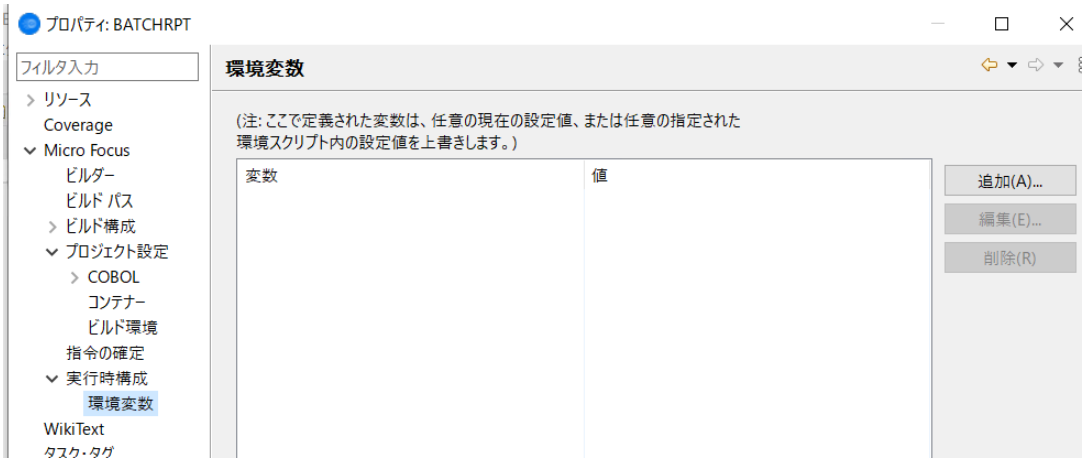
ソース エンコーディング： ANSI

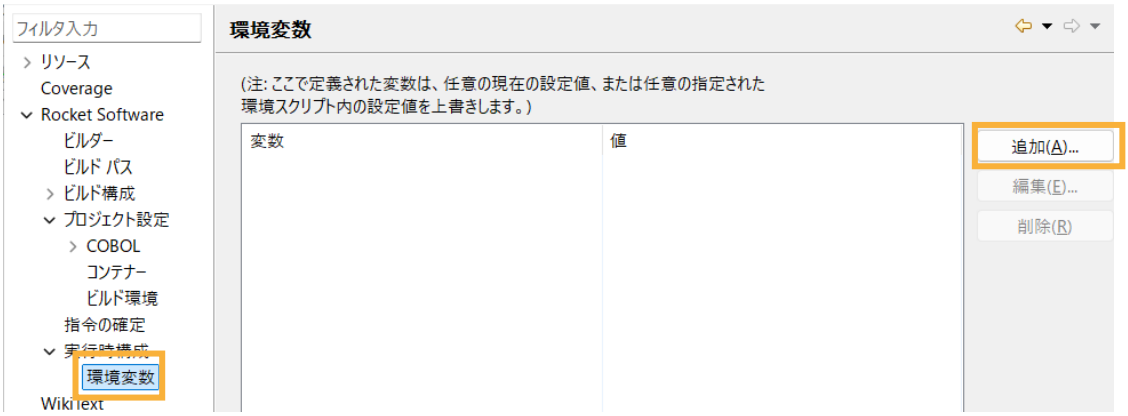
追加指令： “assign(external)”



12) 環境変数の構成

左側のツリーより、[Rocket Software] > [実行時構成] > [環境変数] を選択し、[追加(A)] をクリックします。

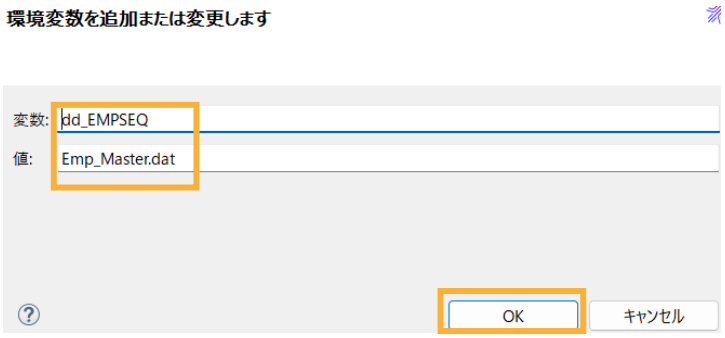




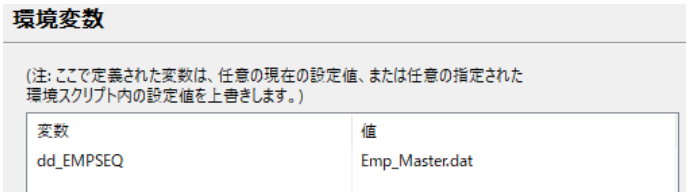
以下の入力を行い、[OK] をクリックします。

変数： dd_EMPSEQ

値： Emp_Master.dat



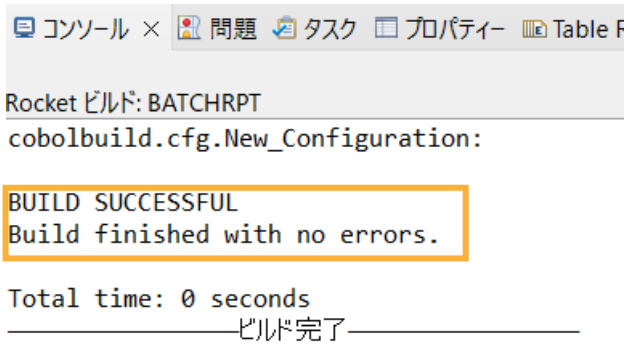
観光変数の一覧に dd_EMPSEQ の情報が登録されます。



同じ手順で、以下の環境変数の情報を追加してください。

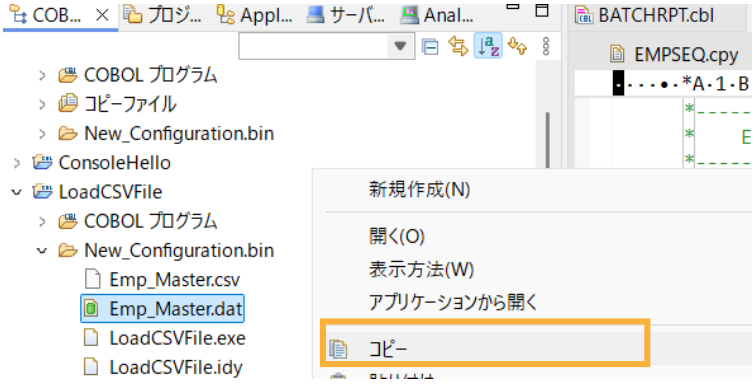
変数	値
dd_CNTLCARD	Cntl_Card.dat
dd_HIRERPT	Hire_Report.dat

ダイアログをクローズすると、自動的にビルドが実行され、正常に終了することを確認します。

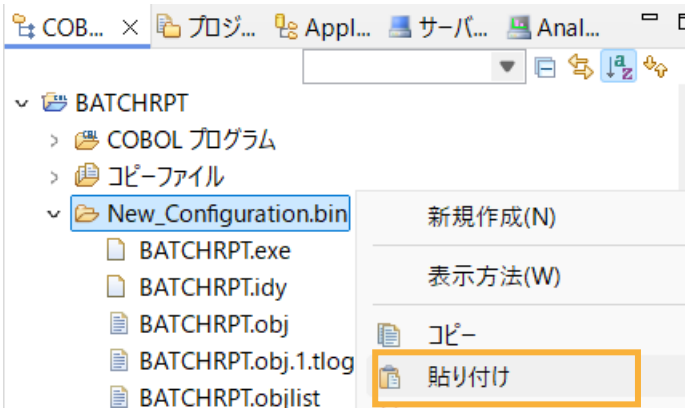


13) 入力ファイルのコピー

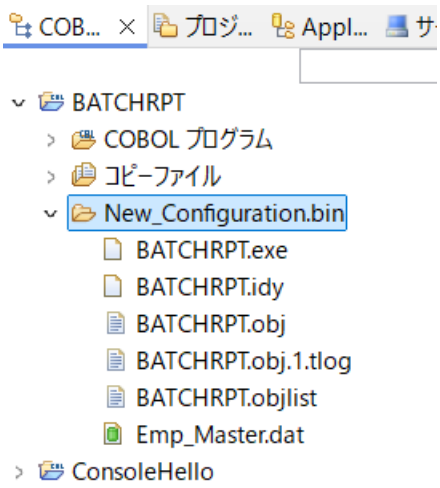
7) で作成したファイルを入力ファイルとして使用します。COBOL エクスプローラービューにて LoadCSVFile プロジェクトの New_Configuration.bin フォルダ配下にある Emp_Master.dat を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[コピー] を選択します。



COBOL エクスプローラービューにて BATCHRPT プロジェクト配下の New_Configuration.bin フォルダを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[貼り付け] を選択します。

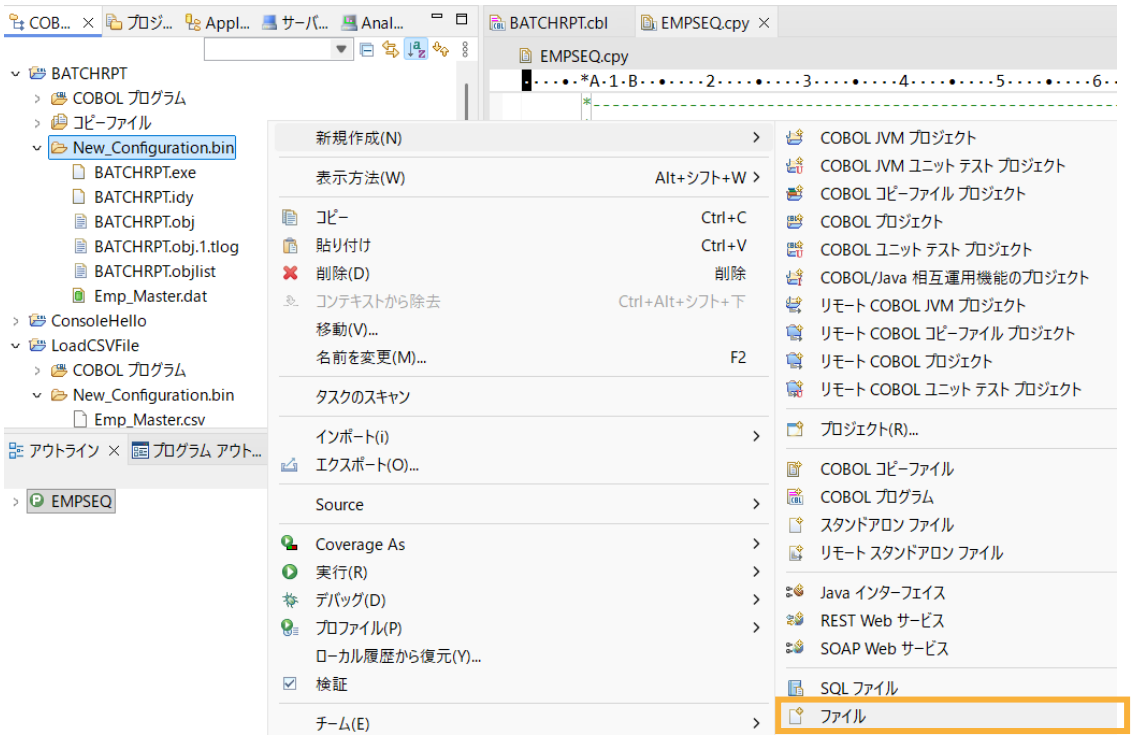


Emp_Master.dat がコピーされます。

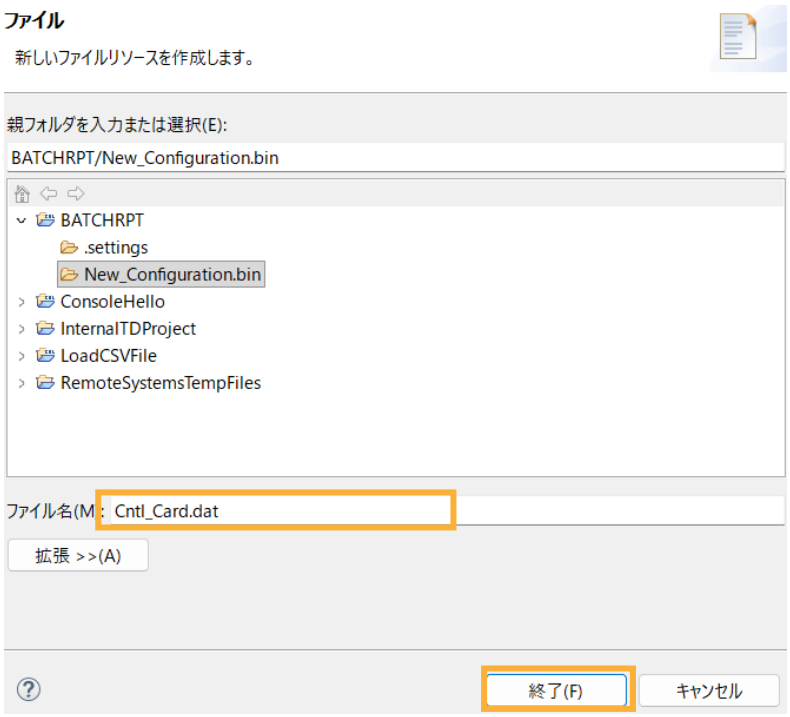


14) 制御ファイルの作成

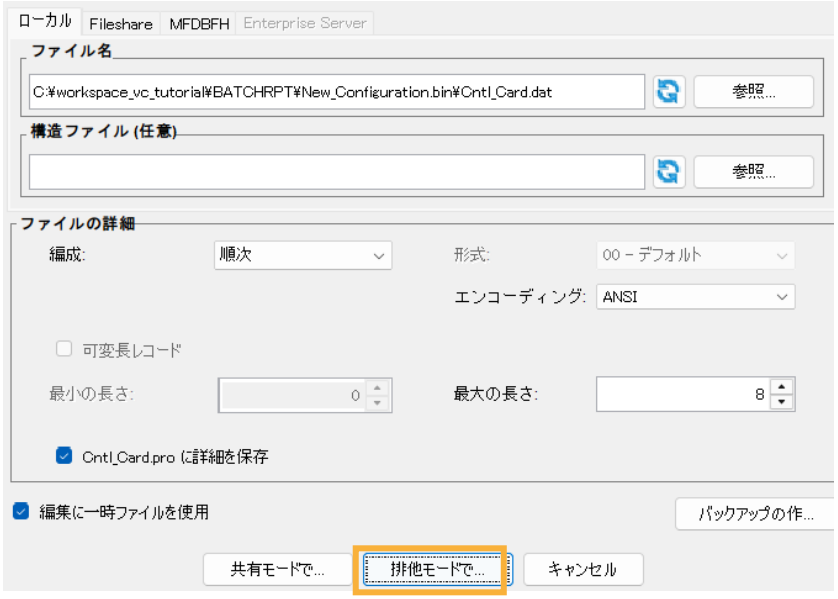
COBOL エクスプローラービューにて BATCHRPT プロジェクト配下の New_Configuration.bin フォルダを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[新規作成(N)] > [ファイル] を選択します。



ファイル名に “Cntl_Card.dat” を指定し [終了(F)] をクリックします。

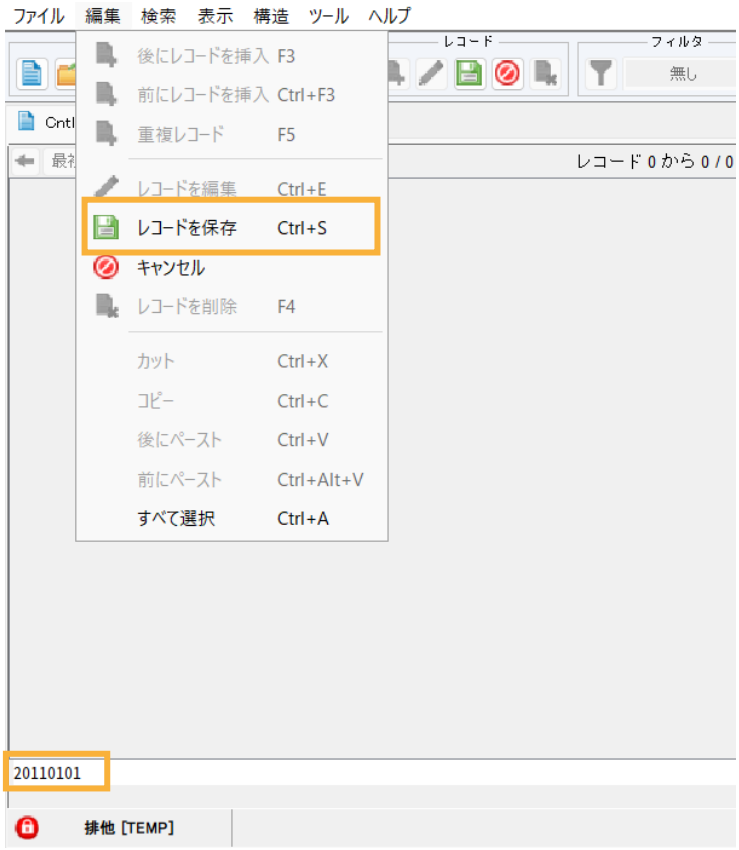


データファイルツールが起動します。「最大の長さ」に 8 を入力して、[排他モードで開く] をクリックします。

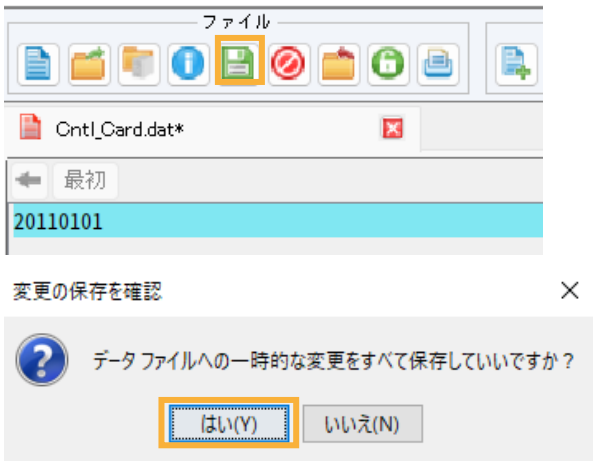


ツール上にて、マウスの右クリックにてコンテクストメニューを表示し、[レコードを挿入] を選択します。

画面下の入力欄に 20110101 を入力し、[編集] > [レコードを保存] を選択します。



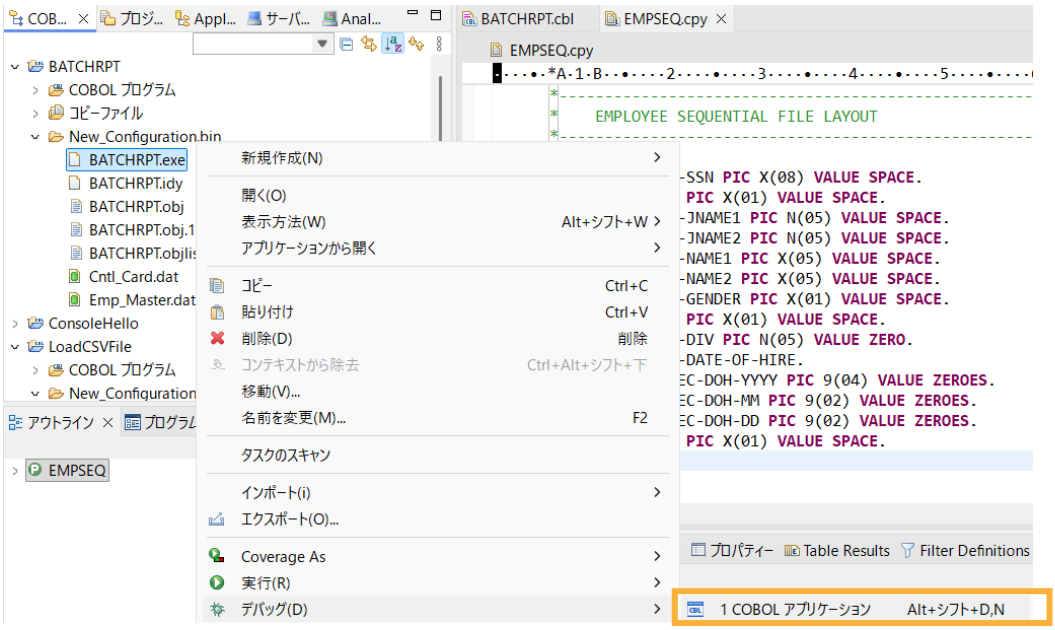
[ファイルのすべての変更を保存する] アイコンをクリックします。確認ダイアログには、[はい(Y)] をクリックします。



データファイルツールを終了します。

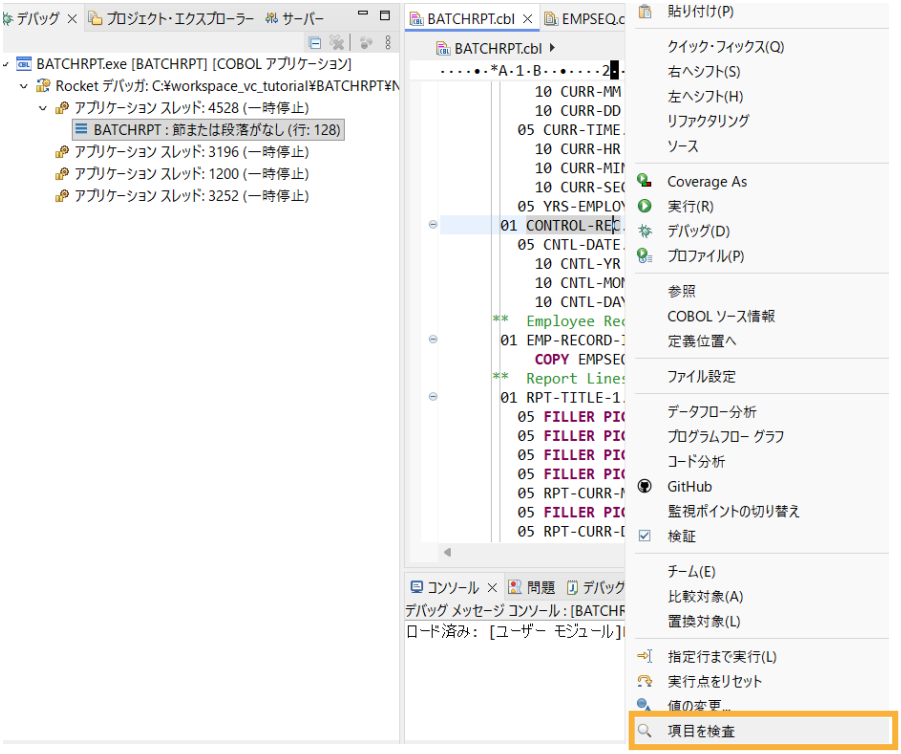
15) COBOL アプリケーションのデバッグ

BATCHRPT プロジェクトの New_Configuration.bin フォルダ配下にある BATCHRPT.exe を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[デバッグ(D)] > [COBOL アプリケーション] を選択します。

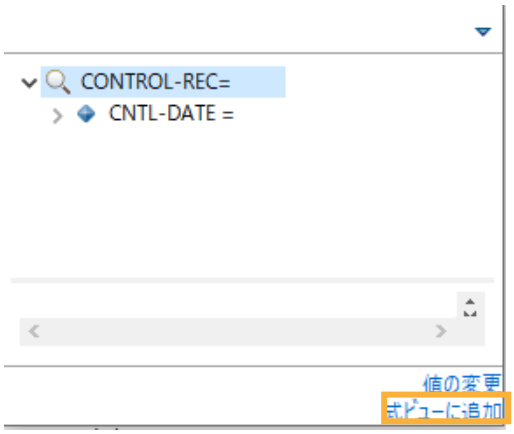


パースペクティブ切り替えの確認ダイアログが表示された場合は[切り替え(S)] をクリックします。

制御ファイルから読み込んだレコードの内容を確認するため、CONTROL-REC に格納される値の変遷を追います。データ部の CONTROL-REC を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[項目を検査] を選択します。

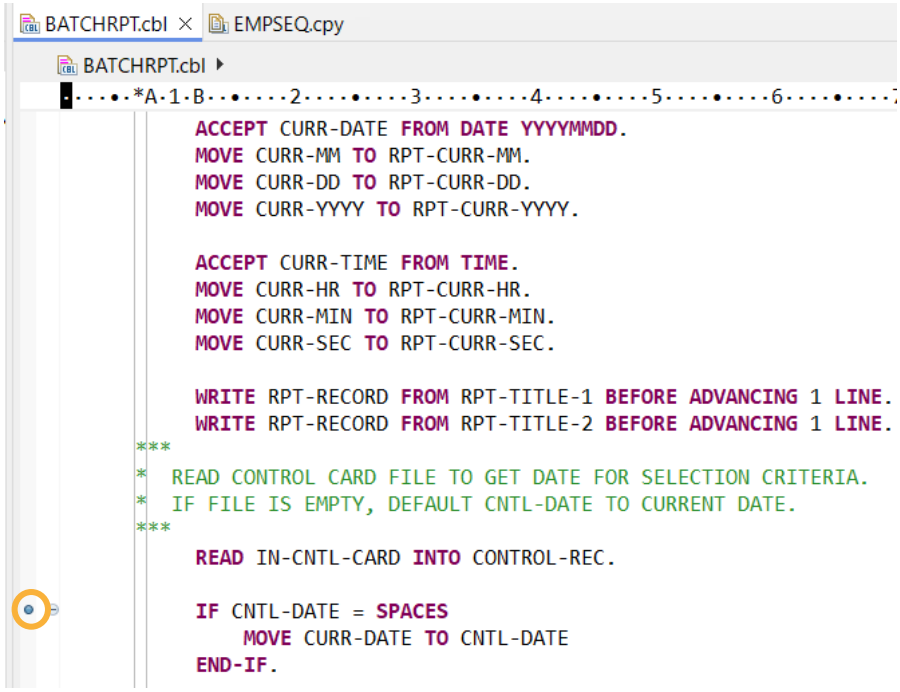


[式ビューに追加] をクリックします。

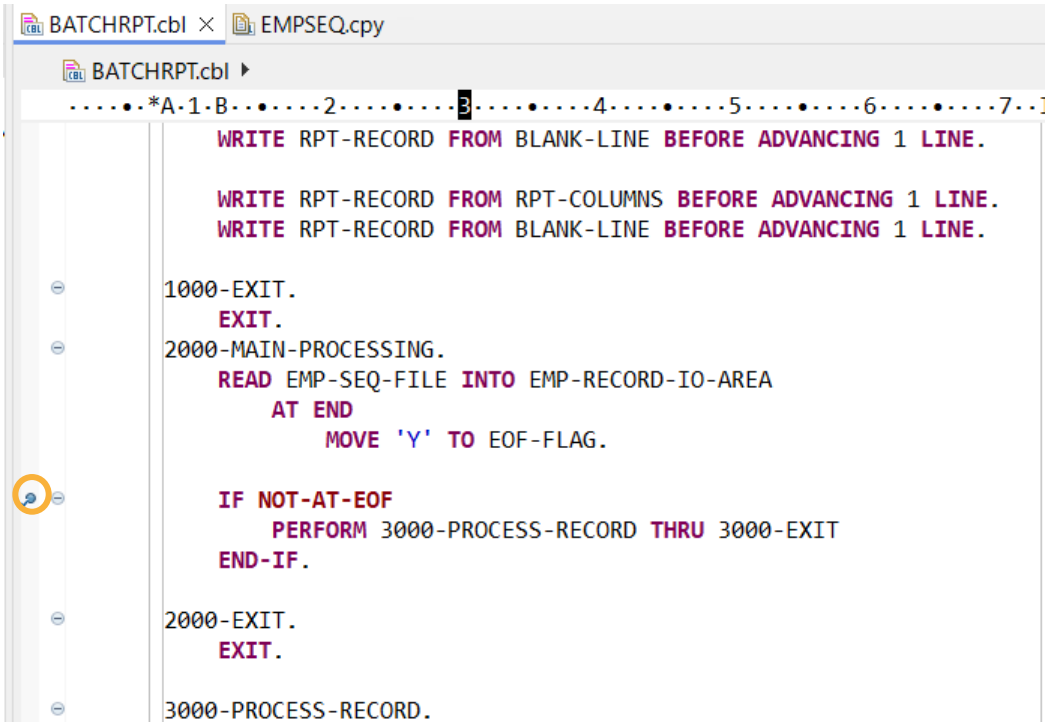


入力ファイルから読み込んだレコードの内容を確認するため、データ部の EMP-RECORD-IO-AREA を同じ要領で式ビューに追加します。(以前の手順で追加した IN-REC 及び OUT-REC は式ビューから削除しても構いません。以下は削除した状態のものです。)

手続き部 1000-START 節の READ 文に続く IF 文でエディタービューの左端をダブルクリックし、ブレークポイントを設定します。



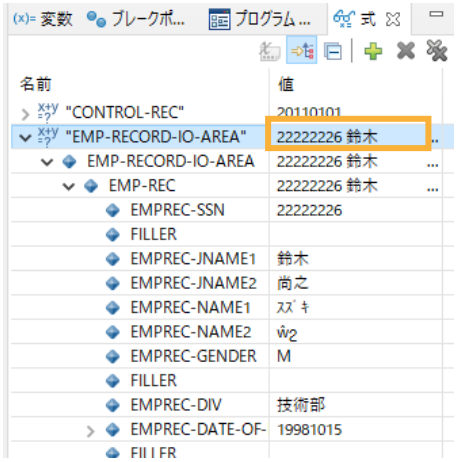
同様に手続き部 2000-MAIN-PROCESSING 段落の READ 文に続く IF 文に、ブレークポイントを設定します。



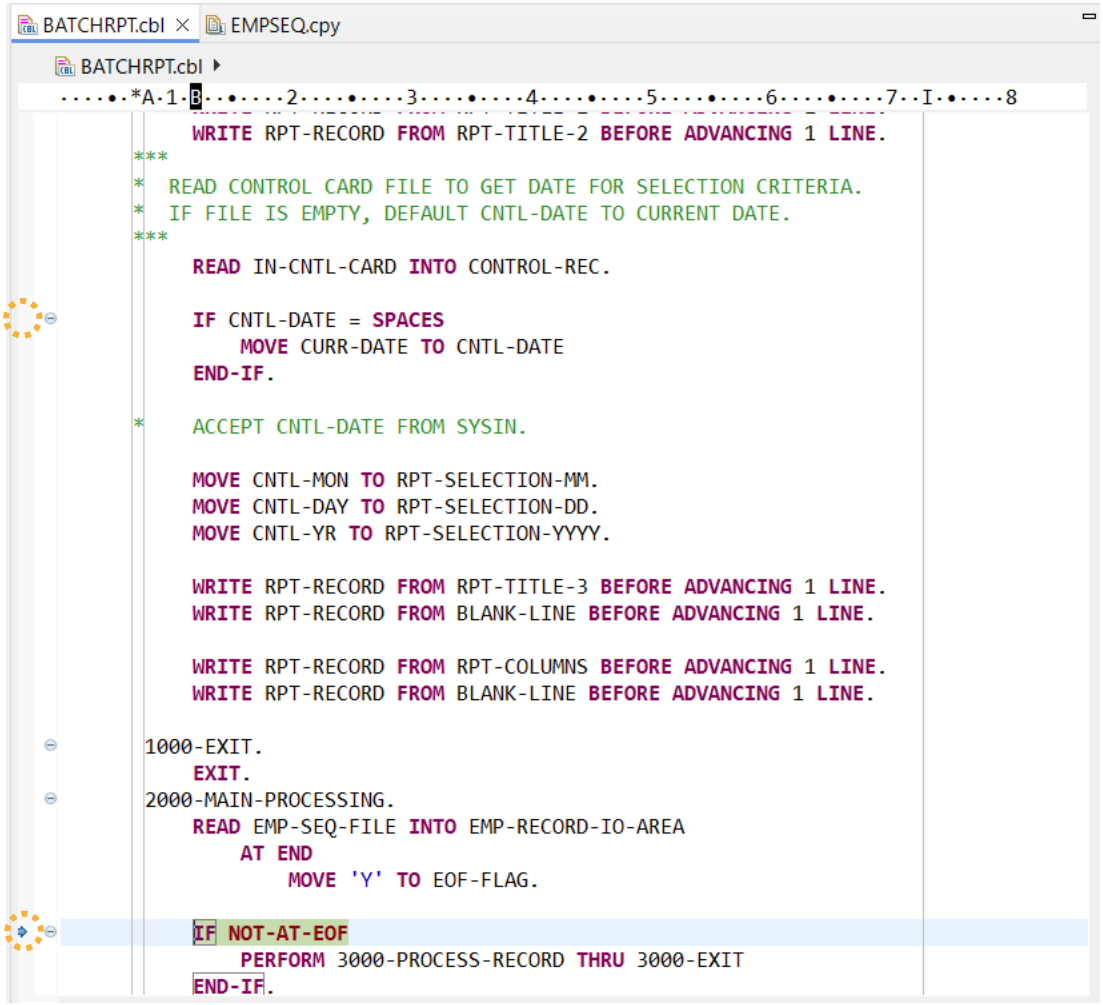
メニューより、[実行(R)] > [再開(M)] を選択するか、ホットキーである F8 キーを押すことで、デバッガーは最初のブレークポイントで実行を中断します。式ビュー中の CONTROL-REC の値に制御ファイルから読み込んだレコードが表示されます。

再度、F8 キーを押すと、デバッガーは 2 番目のブレークポイントで実行を中断します。式ビューの EMP-RECORD-IO-AREA の値に入力ファイルから読み込んだ 1 番目のレコードが表示されます。

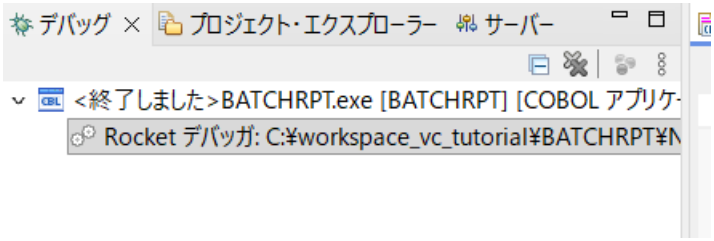
再度 F8 キーを押すと、デバッガーは同一のブレークポイントで実行を中断します。ウォッチ式の EMP-RECORD-IO-AREA の値には、入力ファイルから読み込んだ 2 番目のレコードが表示されます



ブレークポイントは、現在設定中の行の左端をダブルクリックすることで解除できます。さきほど設定した２つのブレークポイントを解除してください。

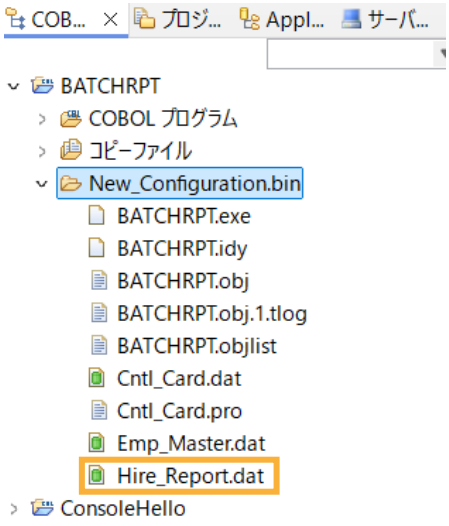


メニューより、[実行(R)] > [再開(M)] を選択するか F8 キーを押してプログラムを実行します。今は、ブレークポイントが設定されていないため、そのままプログラムは正常終了します。

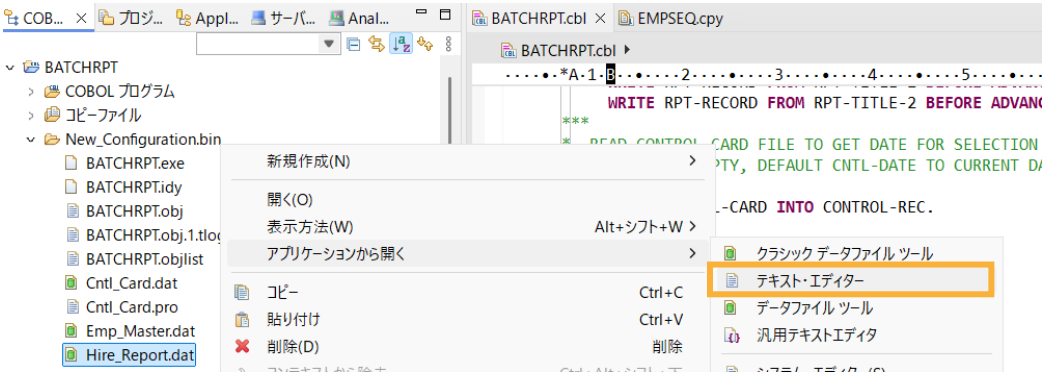


画面右上の COBOL アイコン を選択し、COBOL パースペクティブに戻ります。

COBOL エクスプローラービューにて BATCHRPT プロジェクト配下の New_Configuration.bin フォルダを選択した状態で、F5 キーを押す、もしくは、[ファイル(F)] > [更新(F)] を選択すると、Hire_Report.dat が作成されていることが確認できます。

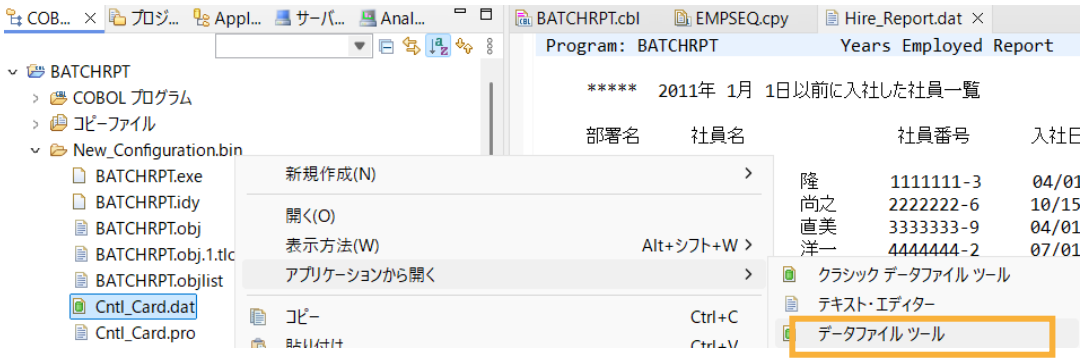


Hire_Report.dat を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[アプリケーションから開く] > [テキスト・エディター] を選択します。

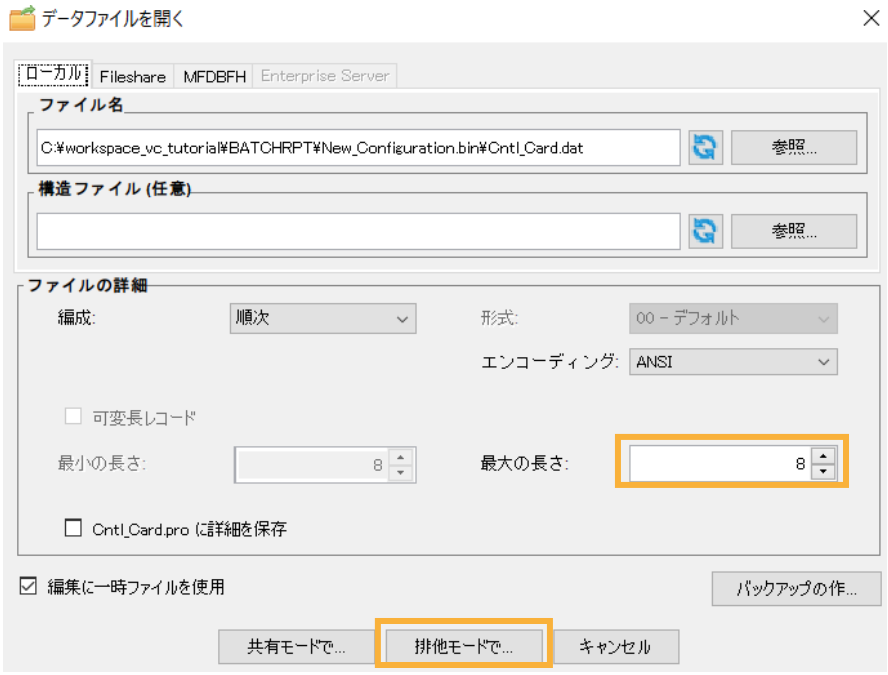


社員 9 名分のデータが表示されることを確認します 。

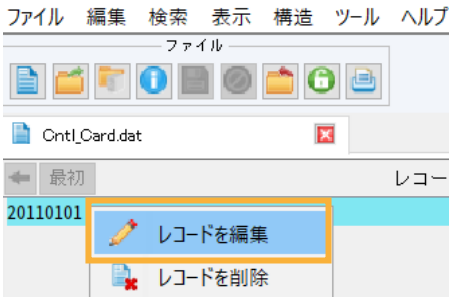
続いて、制御ファイルの値を変更してアプリケーションを実行します。COBOL エクスプローラー上で Cntl_Card.dat ファイルを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[アプリケーションから開く] > [データファイル ツール] を選択します。



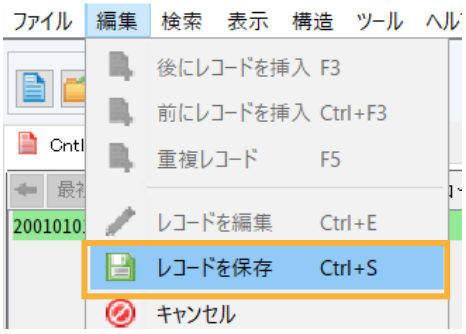
「最大の長さ」に 8 が入力されていることを確認して、[排他モードで開く] をクリックします。



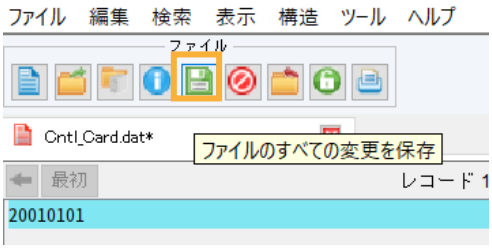
20110101 のレコードを選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[レコードを編集] を選択します。



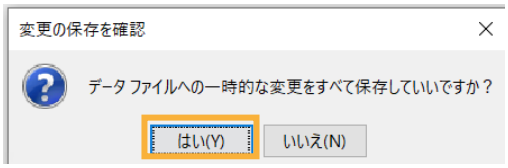
ツール上で、”20110101” の値を “20010101” に更新したうえで、[編集] > [レコードを保存] をクリックします。



[ファイルのすべての変更を保存] アイコンをクリックします。

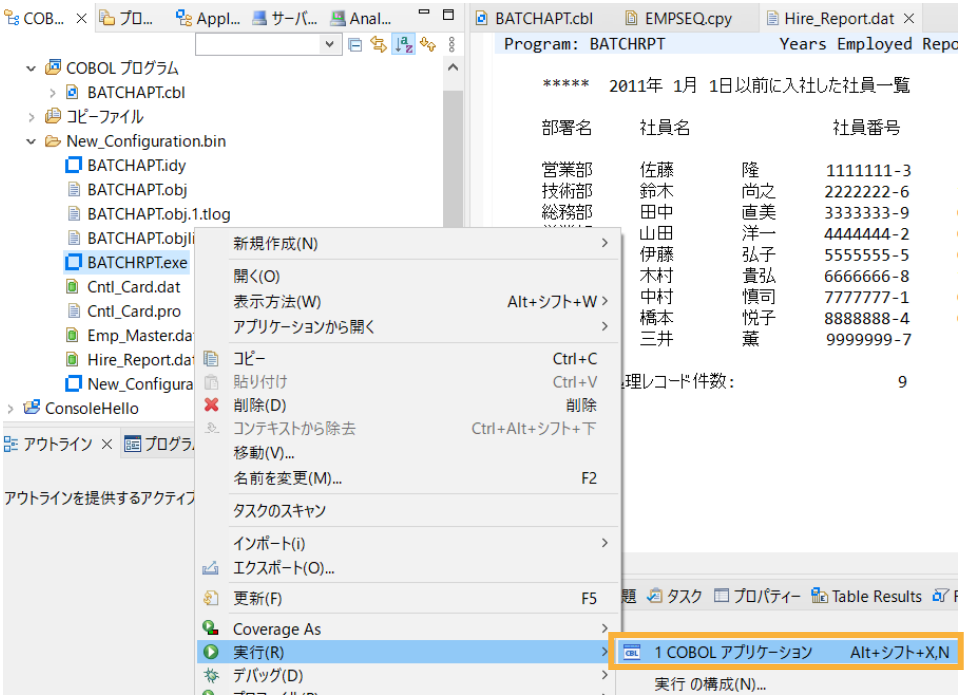


確認ダイアログが表示されますので、[はい(Y)] をクリックします。

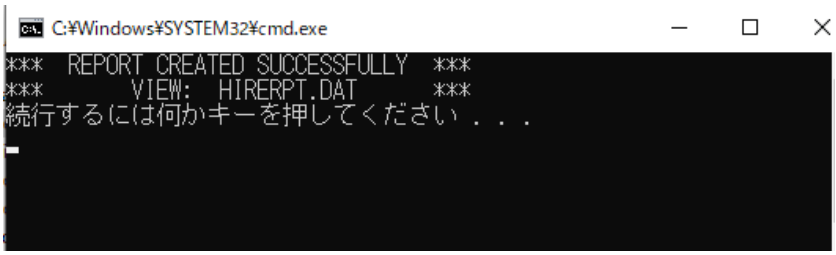


保存後、ツールを終了してください。

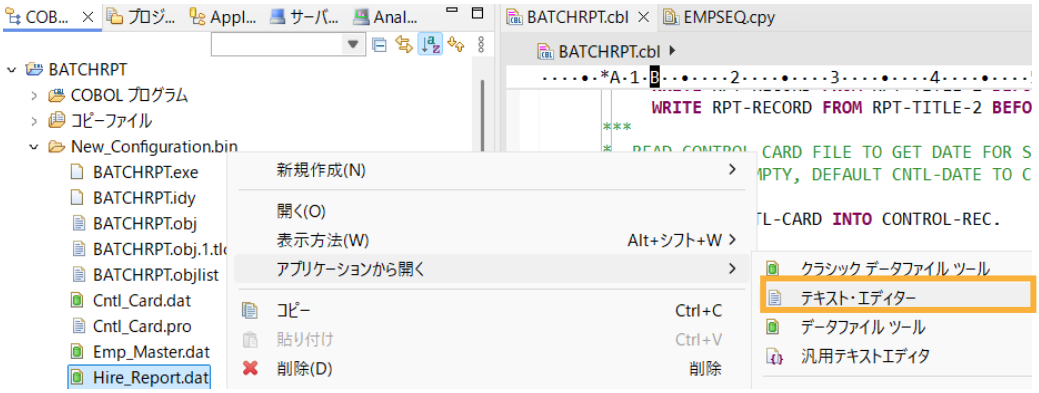
BATCHRPT プロジェクトの New_Configuration.bin フォルダ配下にある BATCHRPT.exe を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[実行(R)] > [COBOL アプリケーション] を選択します。



以下のコンソール画面がポップアップされたら、メッセージに従い、任意のキーを押し、アプリケーションを終了します。



Hire_Report.dat を選択した状態で、マウスの右クリックにてコンテクストメニューを表示し、[更新(F)] を選択します。



2000 年 1 月 1 日以前に入社した社員 4 名分のデータだけが表示されることを確認します。

免責事項

ここで紹介したソースコードは、機能説明のためのサンプルであり、製品の一部ではございません。ソースコードが実際に動作するか、御社業務に適合するかなどに関しまして、一切の保証はございません。 ソースコード、説明、その他すべてについて、無謬性は保障されません。

ここで紹介するソースコードの一部、もしくは全部について、弊社に断りなく、御社の内部に組み込み、そのままご利用頂いても構いません。

本ソースコードの一部もしくは全部を二次的著作物に対して引用する場合、著作権法の精神に基づき、適切な扱いを行ってください。