Micro Focus メインフレームソリューション

スターターズキット

9. PL/I によるバッチジョブチュートリアル

9.1 チュートリアルの準備

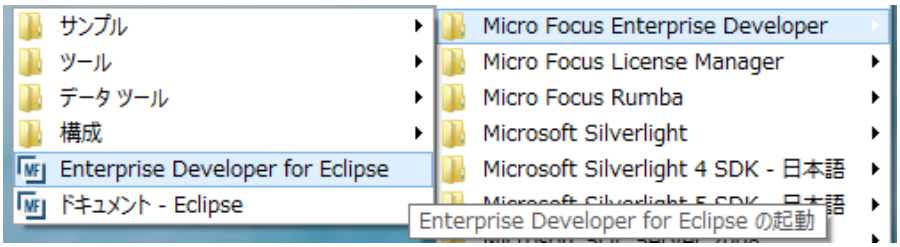
本チュートリアルで使用する例題プログラムは、キットに添付されている PLIJCLtutorial.zip に圧縮されています。これを C:¥ の直下に解凍しておきます。

また、作業用に C:¥work というフォルダを作成しておきます。

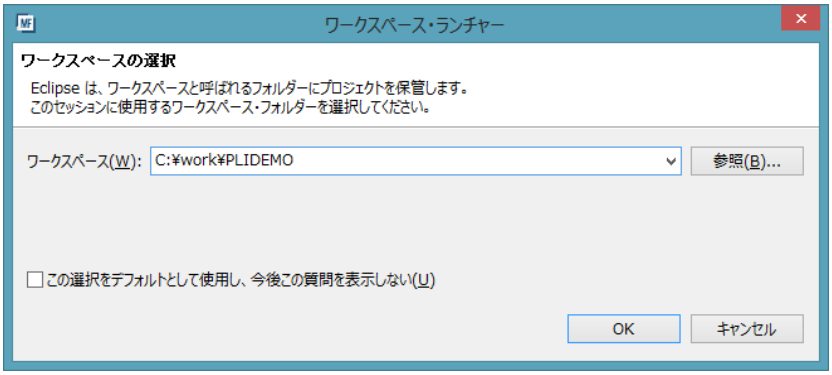
9.2 Enterprise Developer の起動

まず、Enterprise Developer を起動し、新たなワークスペースを作成します。

1) Windows スタートメニューから Enterprise Developer for Eclipse を選択して起動します。

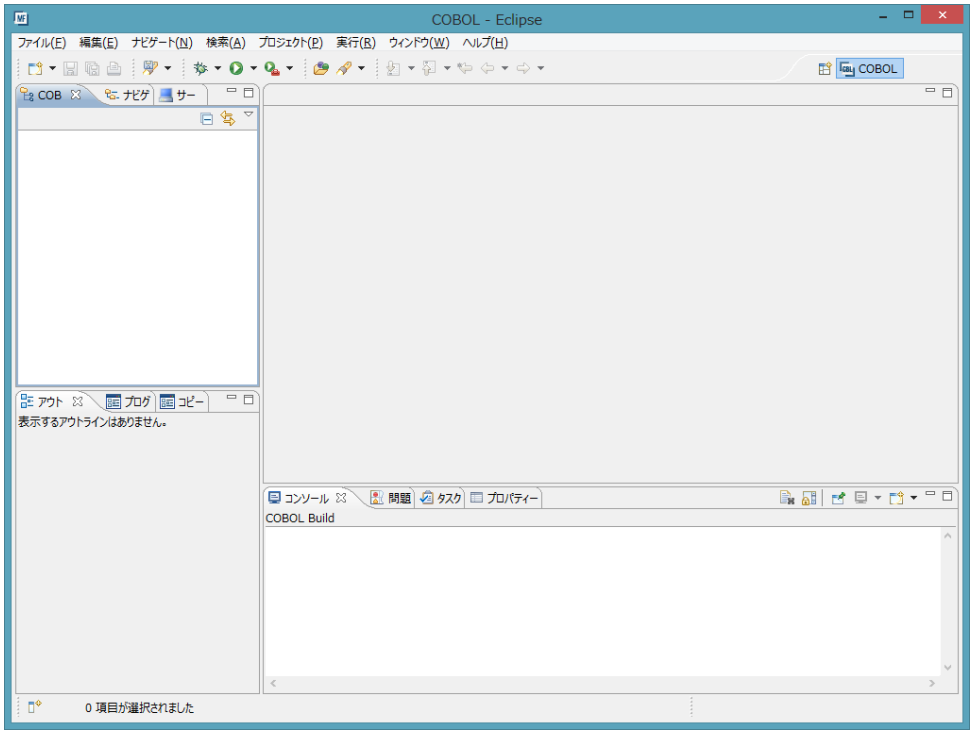


2) 以下のダイアログでは C:¥work¥PLIDEMO を指定し [OK] をクリックします。



3) 以下のバナー画面が現れます。

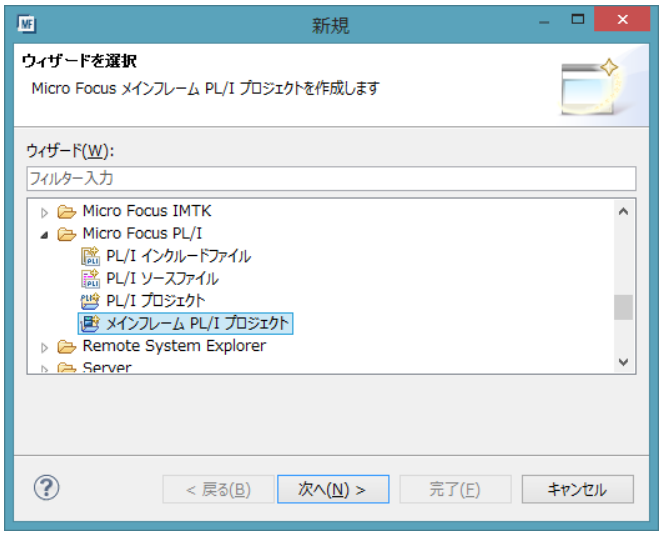
4) 左上の「ようこそ」の右の X をクリックしてバナーを閉じます。以下のように Eclipse の COBOL パースペクティブが開きます。



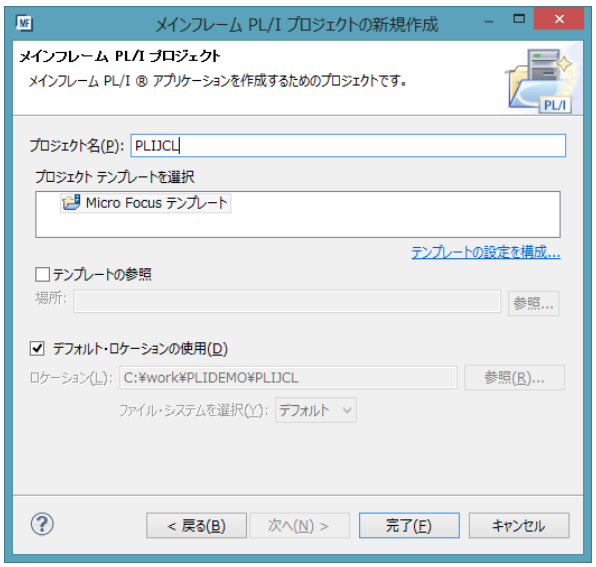
9.3 プロジェクトの新規作成

作成されたワークスペースに新たなプロジェクトを作成します。

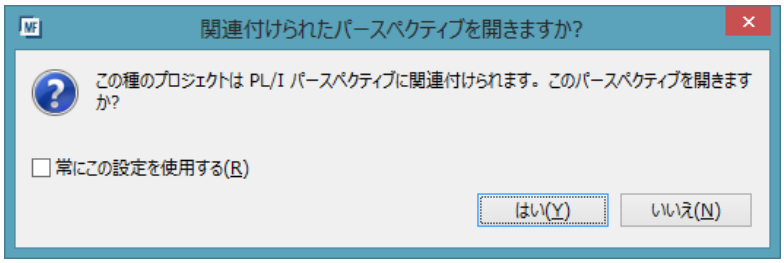
1) [ファイル] > [新規] > [その他…] を選択します。
2) 以下のダイアログで [Micro Focus PL/I] > [メインフレーム PL/I プロジェクト] を選択し [次へ] をクリックします。



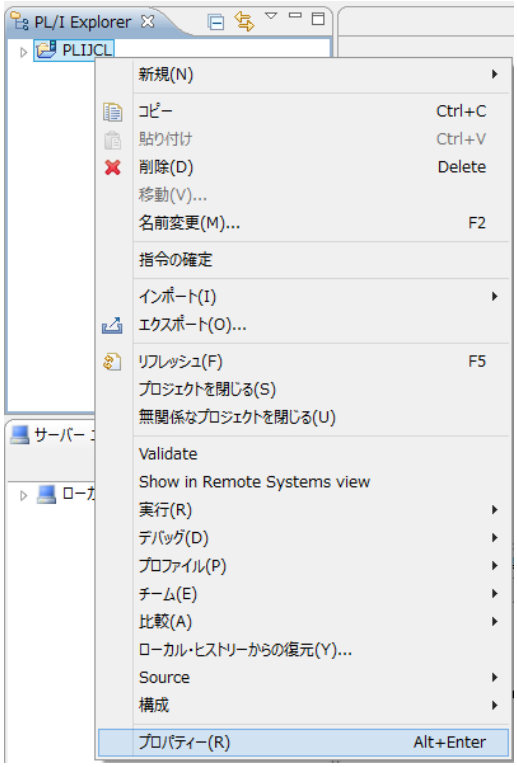
3) 以下のダイアログでプロジェクト名を指定します。ここでは “PLIJCL” と命名します。[完了] をクリックします。



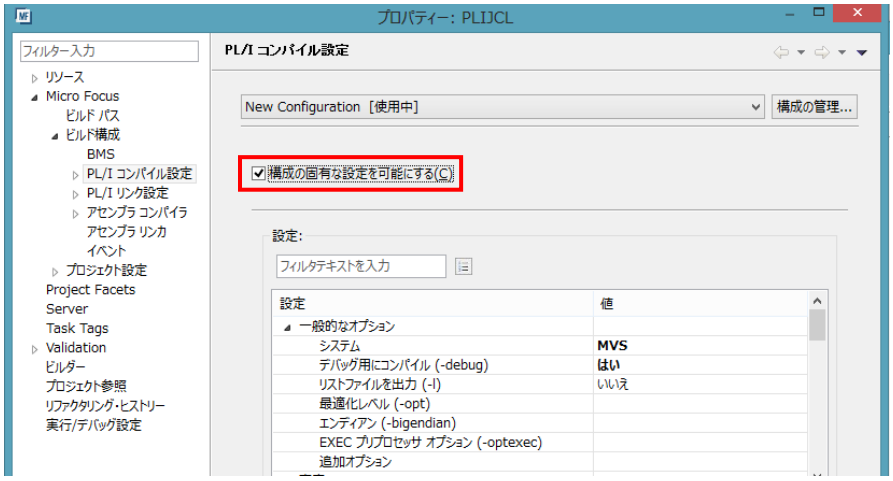
4) 以下のダイアログに対して [はい] をクリックします。



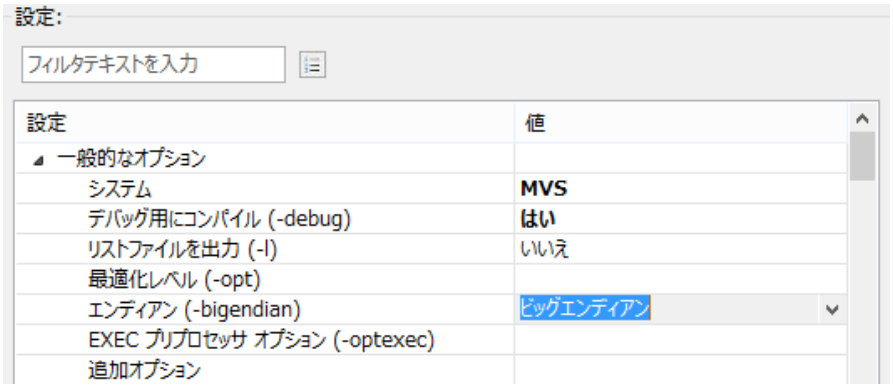
5) 空の PL/I プロジェクトが作成され PL/I エクスプローラ内に以下のようなツリー構造 PLIJCL のノードが作成されます。作成されたプロジェクトに必要なプロパティの設定を行います。PL/I エクスプローラ内で PLIJCL を右クリックして [プロパティー] を選択します。



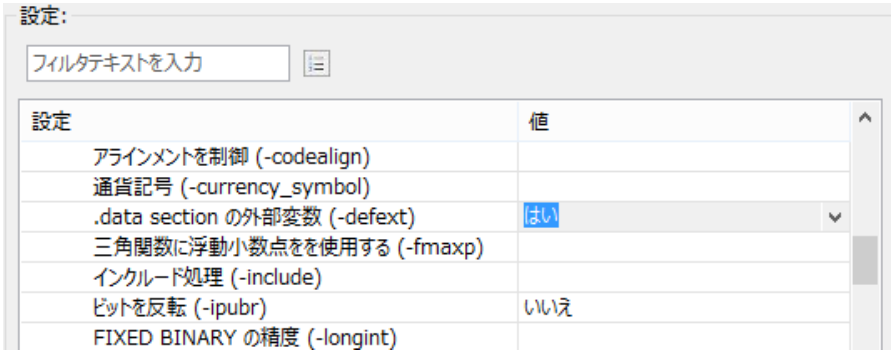
6) 以下のようにプロジェクトのプロパティダイアログが開きます。ここでプロジェクトの各種設定を行うことができます。左側ペインのツリービューにて [Micro Focus] > [ビルド構成] > [PL/I コンパイル設定] を開き、以下のように [構成の固有な設定を可能にする] をチェックオンしてください。



7) その下の指令の設定テーブルで以下のように以下のようにエンディアンとして [ビッグエンディアン] を指定してください。



8) 更にテーブルを下にスクロールし、以下の [data section の外部変数] を [はい] に設定します。

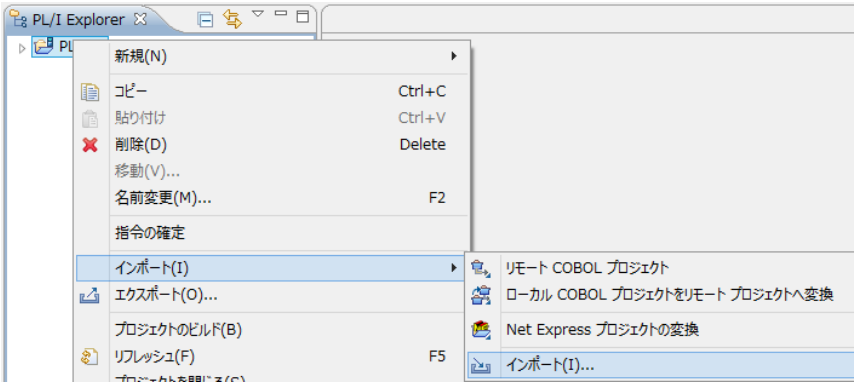


9) [OK] をクリックします。

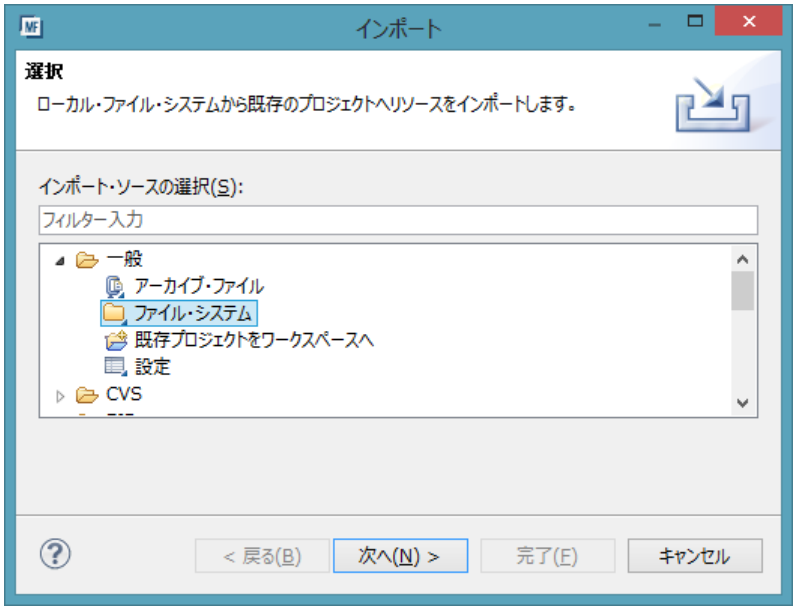
9.4 例題プログラムのインポート

作成されたプロジェクトに例題プログラムをインポートします。

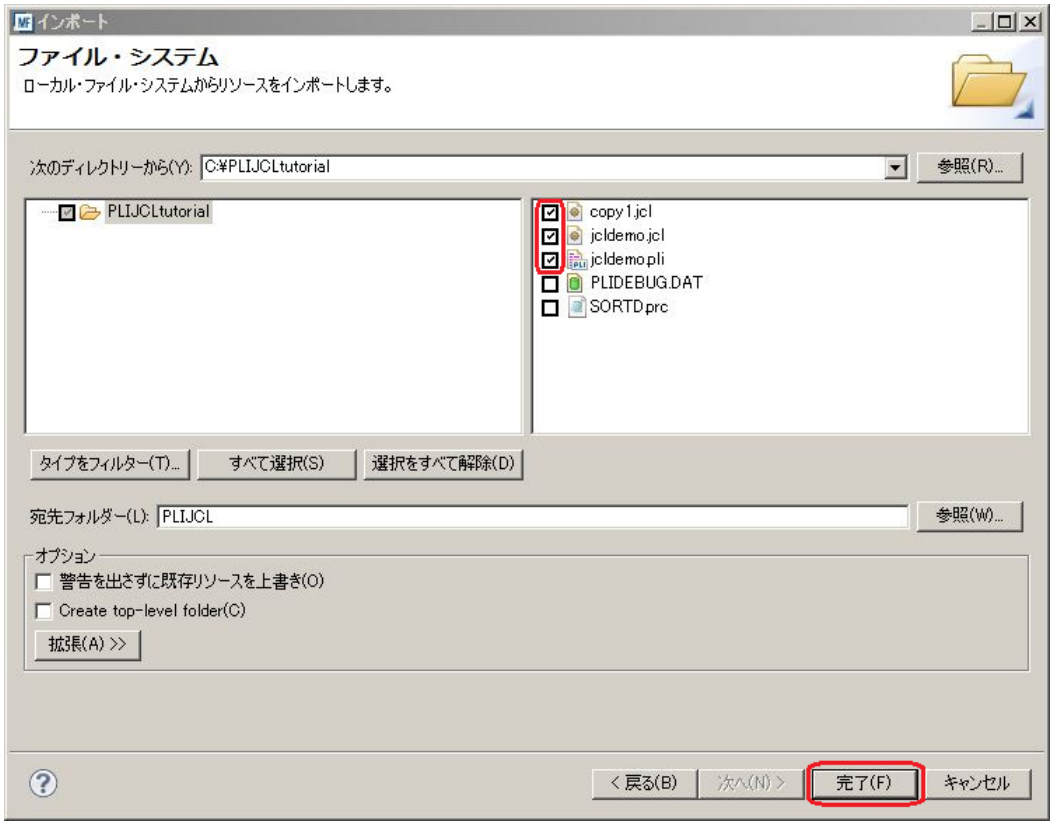
1) PL/I エクスプローラ内で PLIJCL を右クリックして [インポート] > [インポート] を選択します。



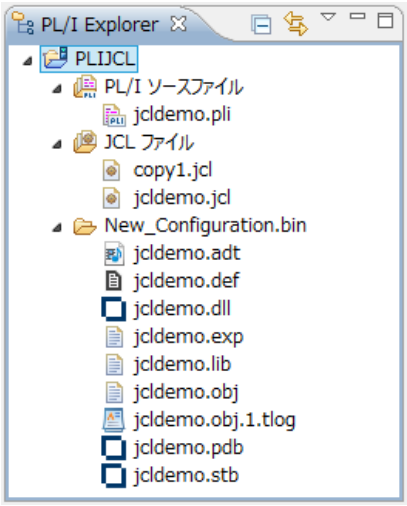
2) 以下のダイアログで [一般] > [ファイル・システム] を選択し、[次へ] をクリックします。



3) 以下のダイアログで [参照…] ボタンをクリックして C:¥PLIJCLtutorial を選択し、以下のように .jcl 拡張子と .pli 拡張子のファイルをチェックオンし、[完了] をクリックします。



4) 以下のように PL/I プログラムと JCL がインポートされ、PL/I エクスプローラのツリービューに配備されます。同時に自動的にコンパイルがなされ、New_Configuration.bin の下に DLL が生成されていることが確認できます。

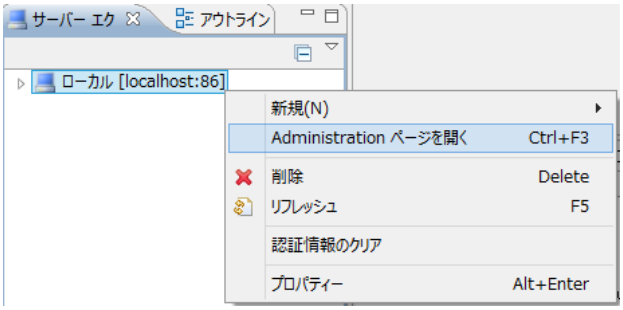


5) 右下のコンソールにエラーなくコンパイルが完了した旨が表示されればプロジェクトは完成となります。

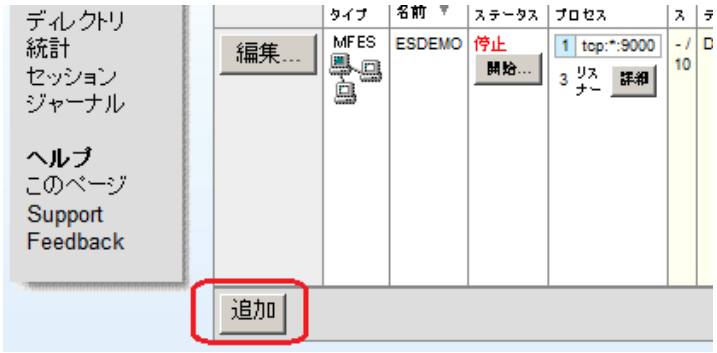
9.5 JES リージョンの作成

コンパイルされたバッチアプリケーションを実行する JES リージョンを作成します。これには Enterprise Developer に内蔵されているテスト用のメインフレームランタイム環境を使用します。これは Enterprise Server と呼ばれるミドルウェアであり、Enterprise Developer には開発用の Enterprise Server が内蔵されています。これがメインフレームアプリケーションのテスト・デバッグのために使用されます。またマイグレーションにおいては本番実行用の Enterprise Server 製品を使用します。

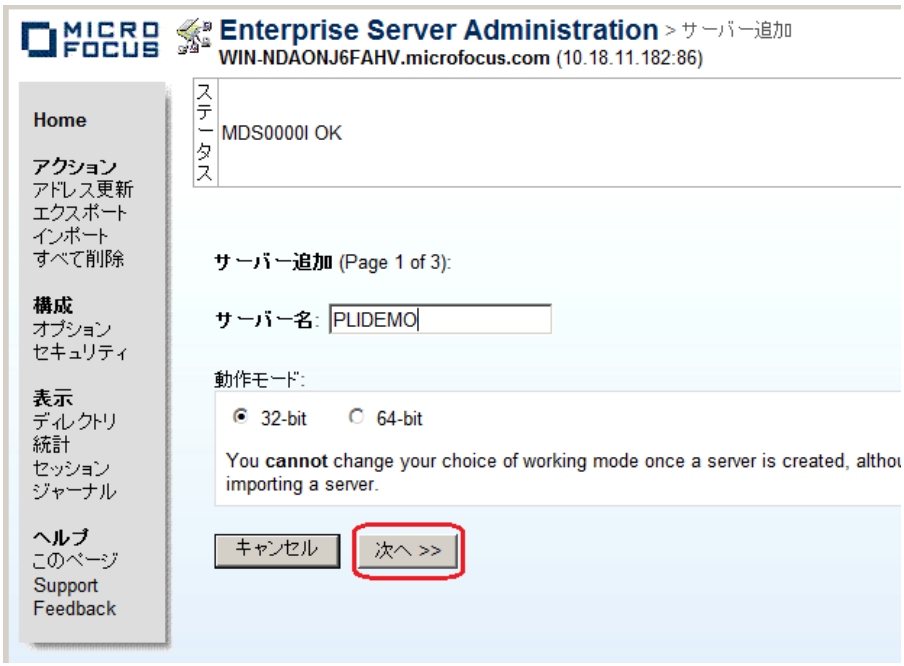
1) Enterprise Developer 内で開発用の Enterprise Server を操作するにはサーバーエクスプローラを使用します。サーバーエクスプローラ内の [ローカル] を右クリックして [Administration ページを開く] を選択します。



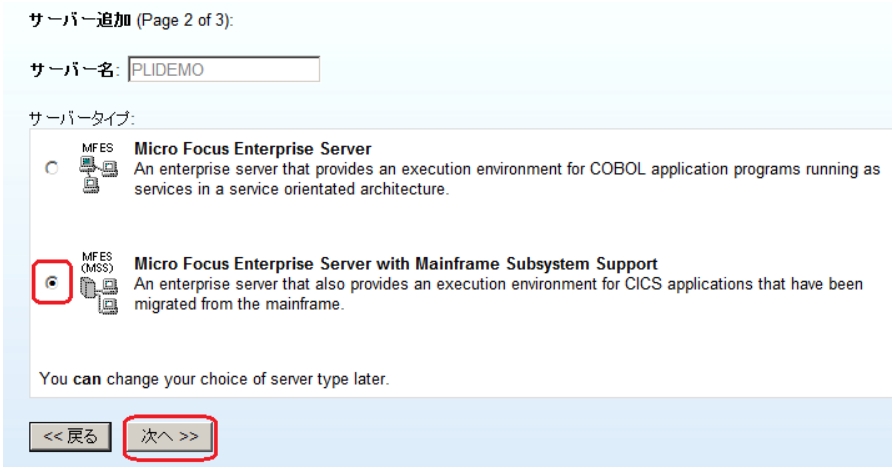
2) 以下のように Enterprise Server の管理コンソールが開きます。既定義の ESDEMO というサーバーが作成されているのがわかります。メインフレームアプリケーションの実行のためには新たなサーバー (JES リージョン) を定義する必要があります。画面下部の [追加] ボタンをクリックします。



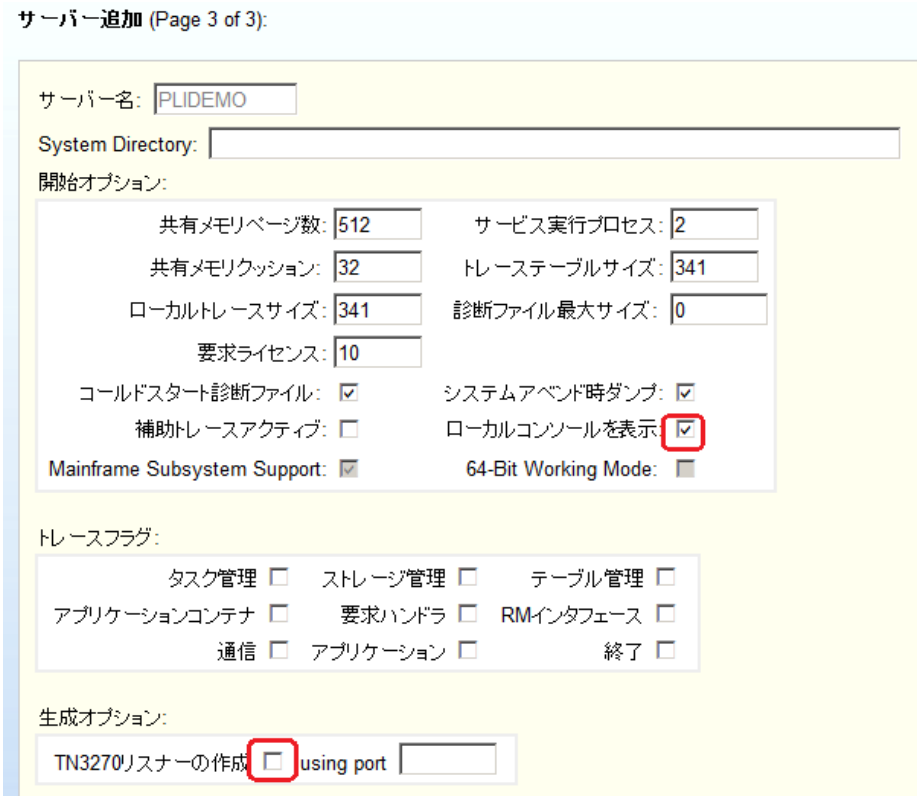
3) 以下の画面に遷移します。新規に作成するサーバー名として PLIDEMO を入力し、[次へ] をクリックします。



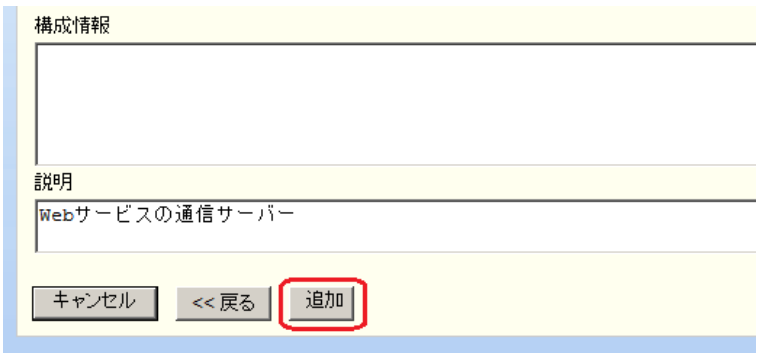
4) 以下の画面では “Micro Focus Enterprise Server with Mainframe Subsystem Support” のラジオボタンを選択し、[次へ] をクリックします。



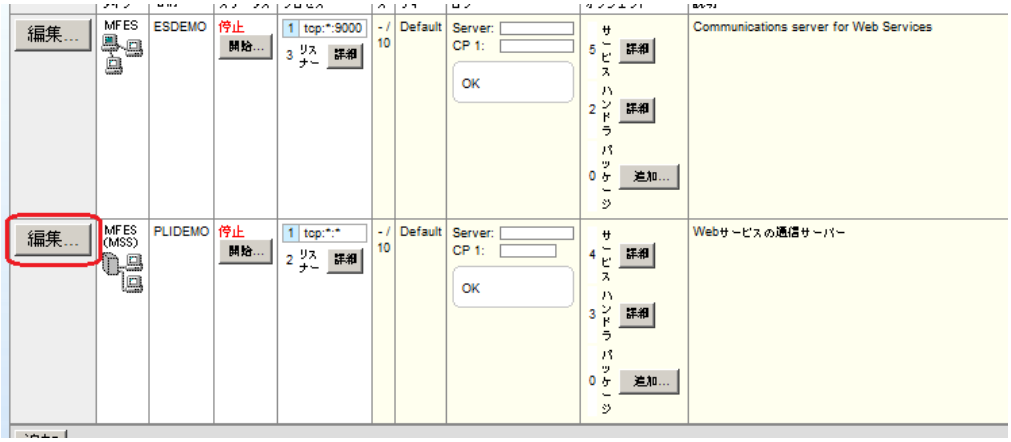
5) 以下の画面では、[ローカルコンソールを表示] のチェックをオンにし、[TN3270 リスナーの作成] のチェックをオフにします。



6) [追加] をクリックします。

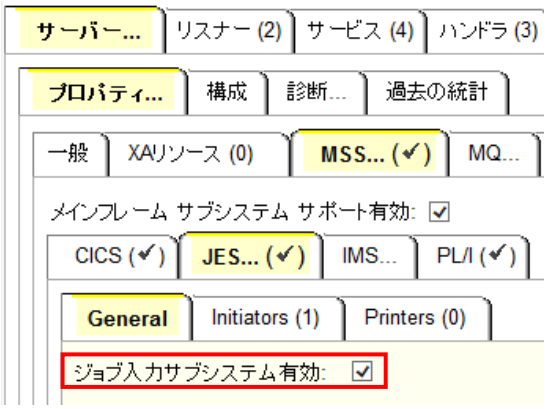


7)　以下のように PLIDEMO が新規に追加されました。左端の [編集…] ボタンをクリックします。



8)　管理コンソールの上端のタブをダブルクリックすると Eclipse 内に管理コンソールが最大化表示されます。

9)　[サーバー] > [プロパティ] > [MSS…] > [JES] のタブを開き、以下のように [ジョブ入力サブシステム有効] のチェックをオンにします。



10) 以下のように各フィールドに値を設定します。

[JES プログラムパス] はジョブステップで実行されるアプリケーションプログラムの探索先パスですので、開発プロジェクトの bin ディレクトリを指定しています。

[システムカタログ] は、JES リージョンで仮定されるマスターカタログの置き場所です。パス名を入力した後に改行を入れないように注意してください。

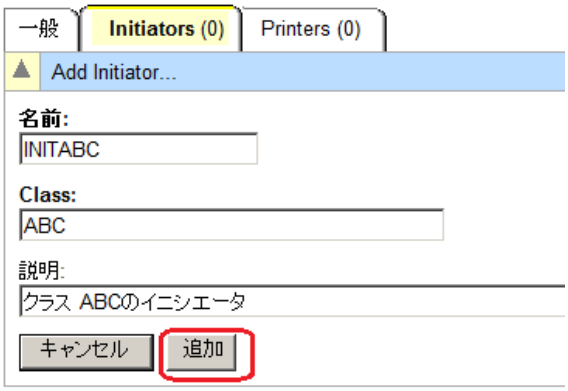
[データセットの省略時ロケーション] はジョブの実行ともに生成されるスプールデータやカタログされるデータセットの置き場所です。ここで指定している C:¥work¥PLIDEMO¥DATASETS は存在していませんので Windows エクスプローラを使用して空のフォルダを作成しておきます。こちらもパス名を入力した後に改行を入れないように注意してください。

[システムプロシージャライブラリ] は、ジョブの実行時に使用されるプロシージャライブラリの名前です。

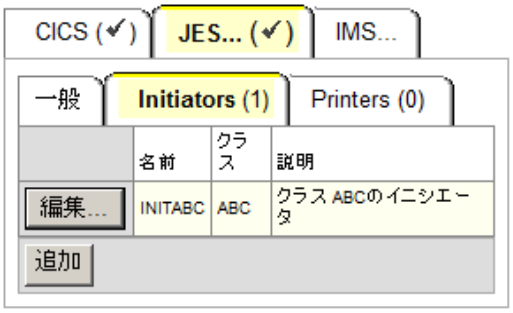
11) [Apply] ボタンをクリックします。

12)「JES…」> [Initiators] タブを開き、[追加] をクリックします。

13) 以下のように入力し [追加] ボタンをクリックします。



14) 以下のようにジョブクラス A, B, C に対する JES イニシエータが定義されます。



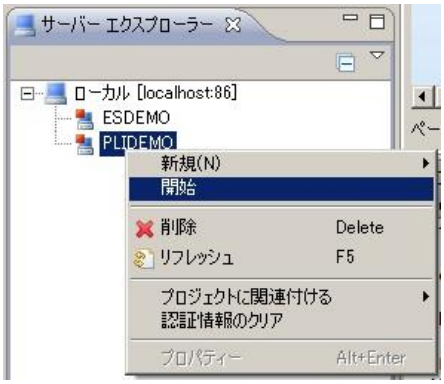
15) [サーバー] > [プロパティ] > [MSS…] > [PL/I] のタブを開き、以下のように [PL/I 有効] のチェックをオンにし、以下の通り入力します。

16) [Apply] をクリックし、JES リージョンが作成されました。

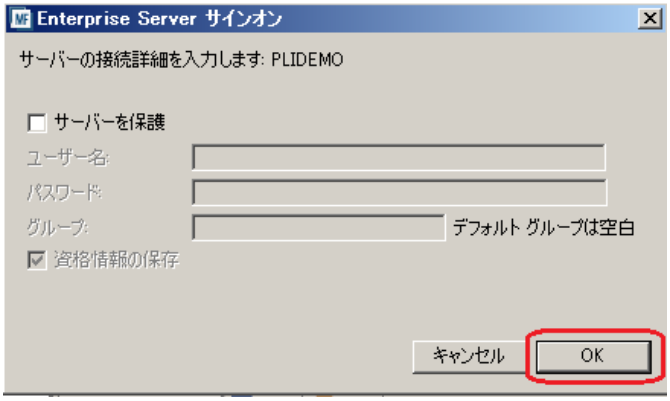
9.6 JES リージョンの起動

作成された JES リージョンを起動します。

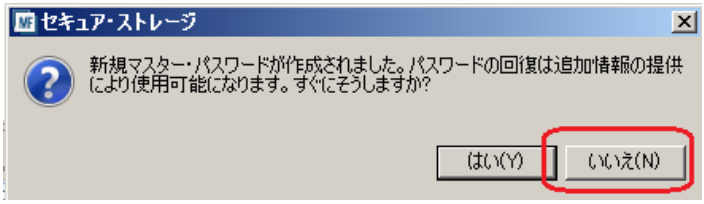
1) サーバーエクスプローラ内に新規作成された PLIDEMO が表示されていることを確認します。もし表示されていなければ [ローカル] を右クリックして [リフレッシュ] を選択してください。
2) PLIDEMO を右クリックし [プロジェクトに関連付ける] > [PLIJCL] を選択します。
3) PLIDEMO を右クリックし [開始] を選択します。



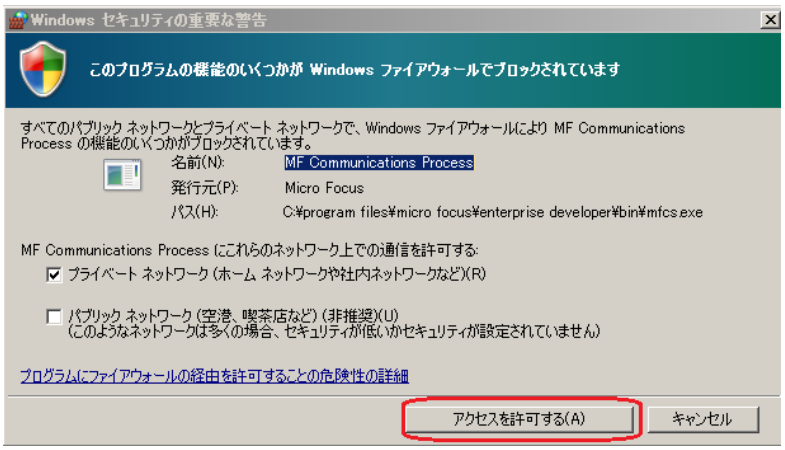
4) このチュートリアルではセキュリティプロテクトはかけないで進めます。以下のダイアログが現れたらそのまま [OK] をクリックします。



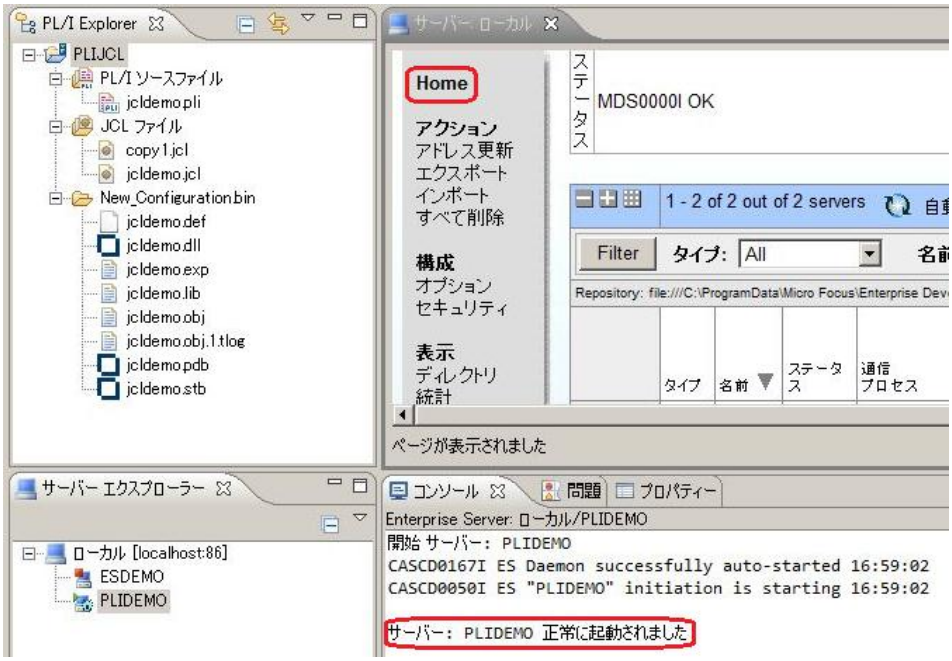
5) 以下のダイアログでは [いいえ] をクリックします。



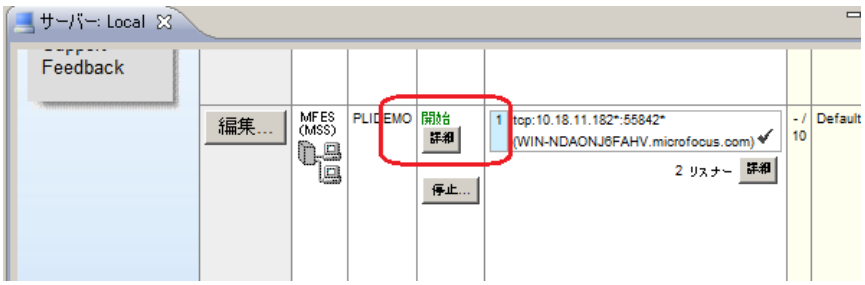
6) Firewall の警告が出る場合には [アクセスを許可する] をクリックします。



7) しばらく待つとコンソールにリージョンが起動された旨のメッセージが現れます。ここで Enterprise Server 管理コンソールの左上の [Home] をクリックします。

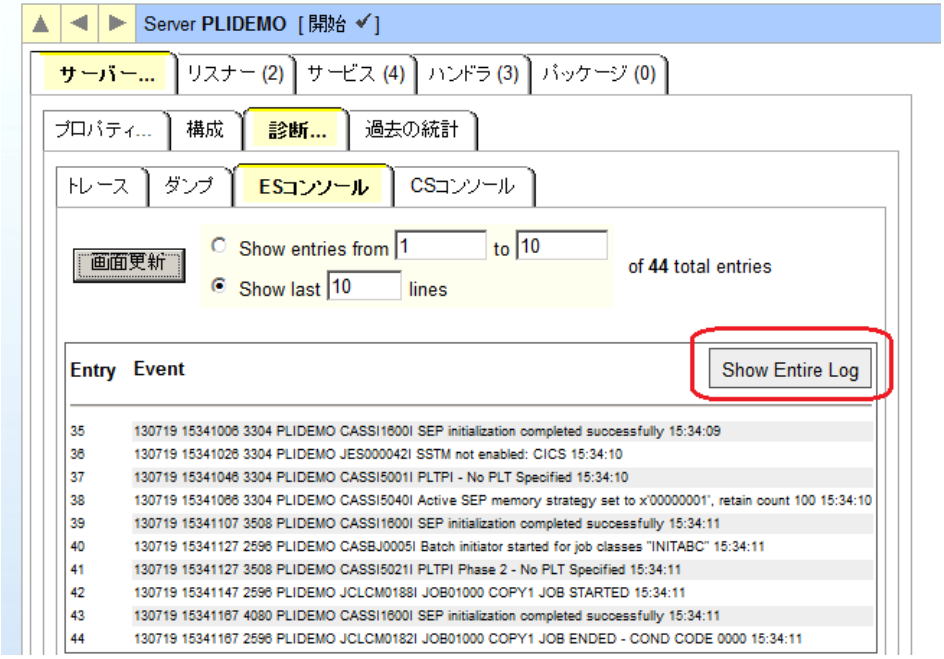


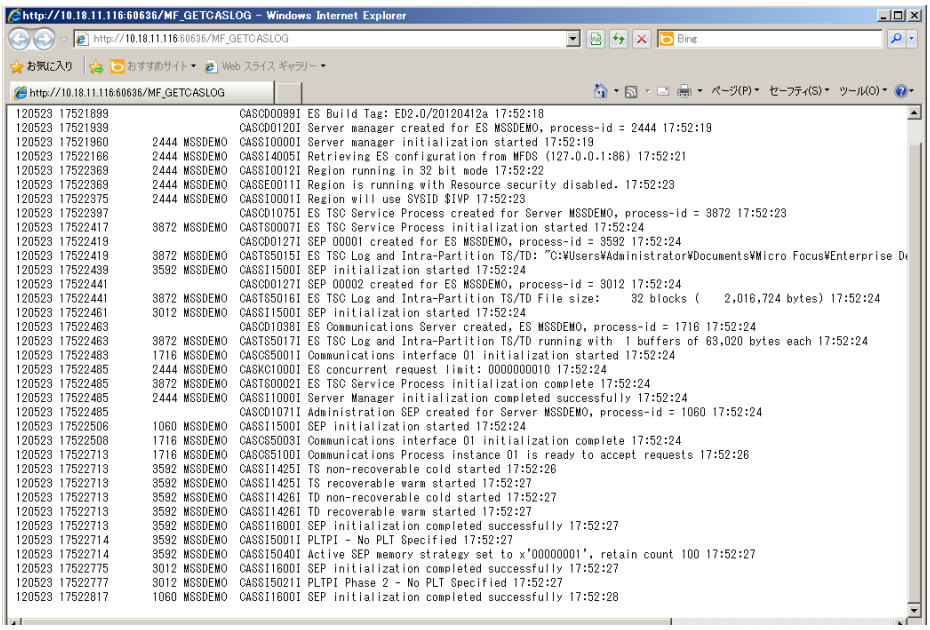
8) PLIDEMO が開始状態になっていることを確認します。



9) PLIDEMO の [詳細] ボタンをクリックします。

10) [サーバー…] > [診断…] > [ES コンソール] で PLIDEMO のコンソールログをリアルタイムにチェックすることができます。また [Show Entire Log] をクリックしてログ全体を表示させることもできます。

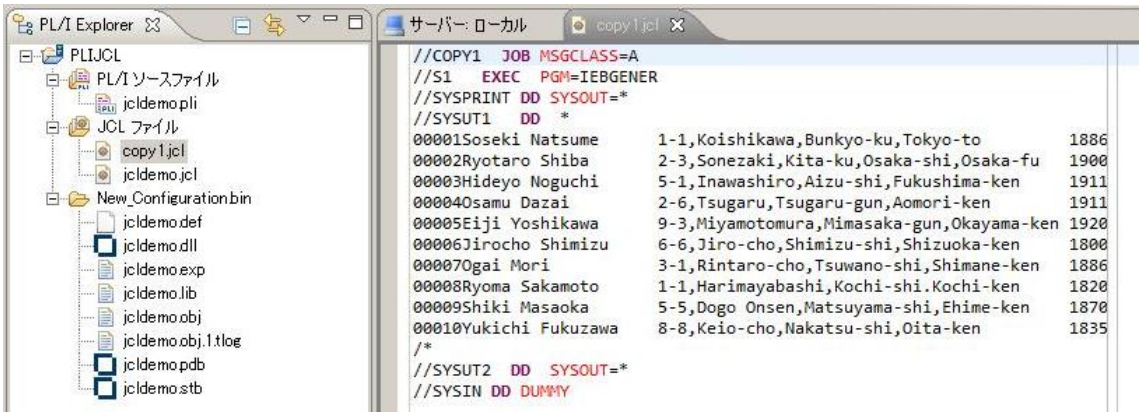




9.7 簡単な JCL の実行

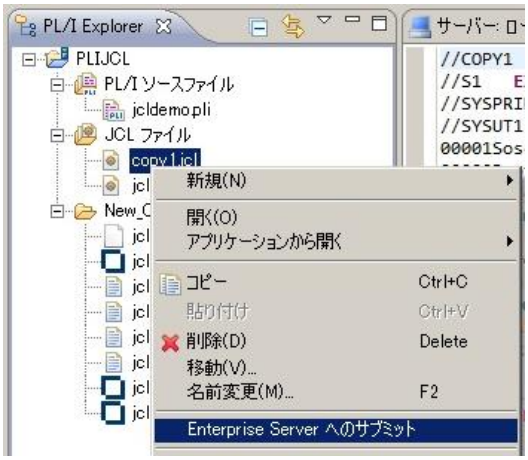
まずもっとも簡単な JCL をこの JES リージョンにサブミットして実行してみます。

1) プロジェクト内の PL/I エクスプローラで copy1.jcl をダブルクリックし、エディタでその内容を確認します。

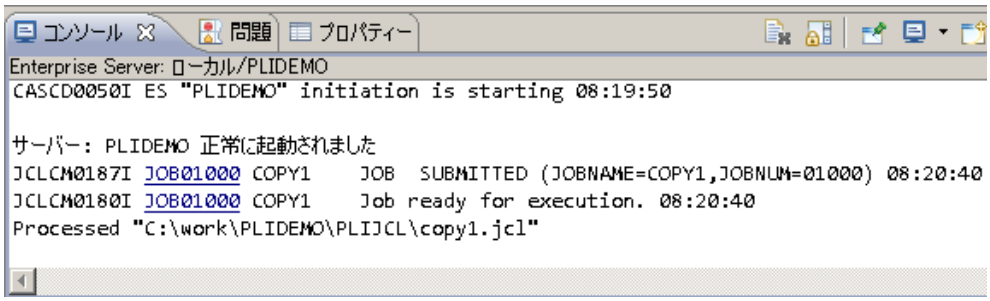


2) このジョブは IEBGENER ユーティリティを起動して JCL 内に書かれたインラインデータを SYSOUT に書き出しているだけのものです。

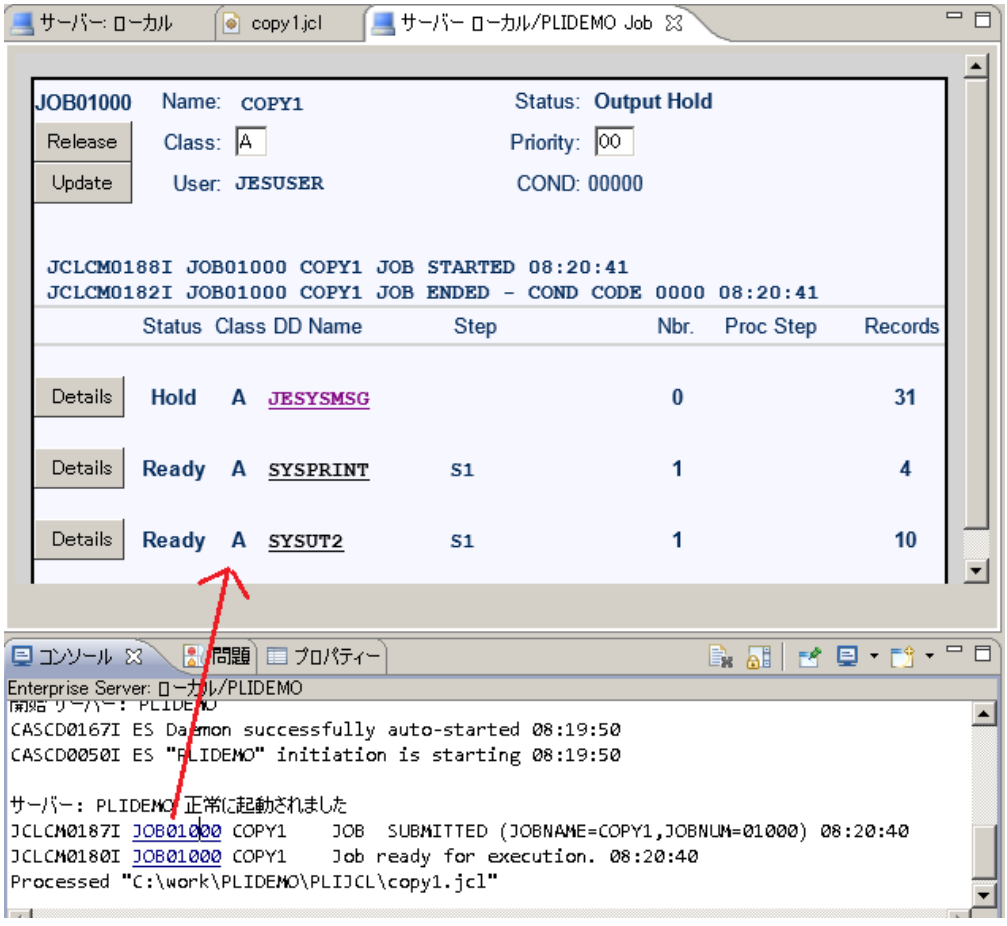
3) PL/I エクスプローラ内で copy1.jcl を右クリックし「Enterprise Server へのサブミット」を選択します。



4) コンソールにジョブがサブミットされたことを示すメッセージが表示されます。



5) コンソールに表示されたジョブ番号をクリックすると、以下のように実行された COPY1 ジョブの出力スプールが表示されます。



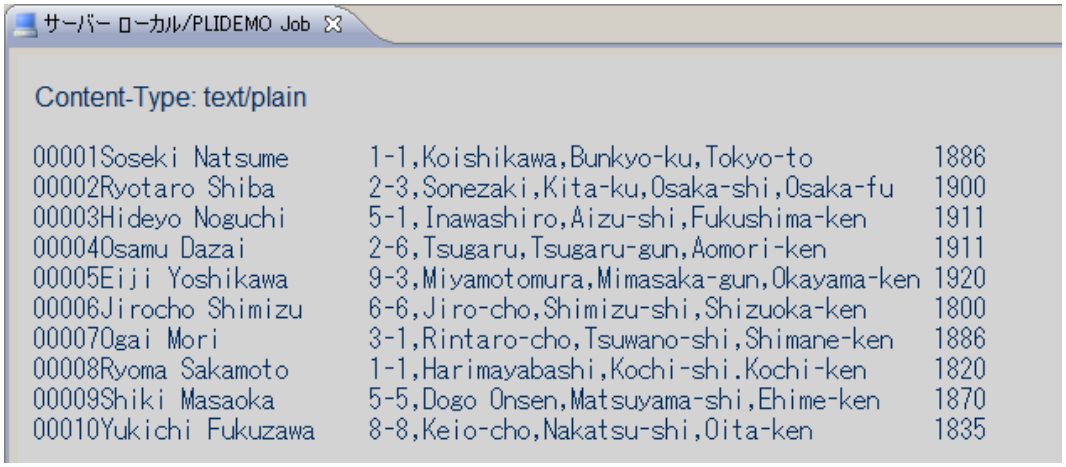
6) JESYSMSG をクリックすると以下のように COPY1 ジョブ実行結果のジョブログが表示されます。

```
サーバー ローカル/PLIDEMO Job

Content-Type: text/plain

     *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
     *-*-*  Micro Focus ESJCL  ASCII  JES2 Version ED22.00.00_022
     *-*-*  Copyright (C) 1997-2013 Micro Focus. All rights reserved.
     *-*-*  Job: 01000 Name: COPY1    User: JESUSER  Date: 01/06/14 Time: 07:48:03
     *-*-*  File: $TXRFDIR/T000000014.T
     *-*-*  DSN:
     *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-

   1 //COPY1   JOB MSGCLASS=A
   2 //S1    EXEC  PGM=IEBGENER
   3 //SYSPRINT DD SYSOUT=*
   4 //SYSUT1   DD  *
  16 //SYSUT2  DD  SYSOUT=*
  17 //SYSIN DD DUMMY
**** JCLCM0180I Job ready for execution.
**** Execution on Server PLIDEMO   Process      7024

     07:48:05 JCLCM0188I JOB  STARTED

     07:48:05 JCLCM0190I STEP STARTED   S1
     07:48:05 JCLCM0199I Program MFJGENER is COBOL      ASCII  Big-Endian    NOAMODE.
     MFE2014.S0106.S074803.J01000.D00001.SYSPRINT                      SYSPRINT
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*4803.J01000.D00001.SYSPRINT.DAT  SPOOLED
     MFE2014.S0106.S074803.J01000.D00002.SYSUT1                        SYSUT1
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*074803.J01000.D00002.SYSUT1.DAT  DELETED
     MFE2014.S0106.S074803.J01000.D00003.SYSUT2                        SYSUT2
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*074803.J01000.D00003.SYSUT2.DAT  SPOOLED
     DUMMY                                                             SYSIN
---> 07:48:05 JCLCM0191I STEP ENDED     S1 - COND CODE 0000
```

7) 画面を右クリックし [前に戻る]で前画面に戻り、SYSPRINT をクリックすると以下のように COPY1 ジョブから起動された IEBGENER ユーティリティの実行ログが記録されていることが確認できます。

```
サーバー ローカル/PLIDEMO Job

Content-Type: text/plain

      Micro Focus  MFJGENER Utility Version ED22.00.00_022
      Copyright (C) 1997-2013 Micro Focus. All rights reserved.

 JCLGN0110I(00) - 00000010 RECORDS COPIED FROM SYSUT1 TO SYSUT2
```

8) 画面を右クリックし [前に戻る]で前画面に戻り、SYSUT2 をクリックします。以下のように出力されたスプールの内容が表示されます。



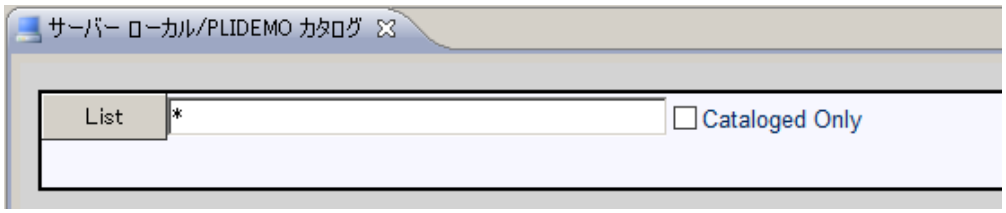
以上で簡単なジョブの実行が確認できました。

9.8 プロシージャライブラリの作成

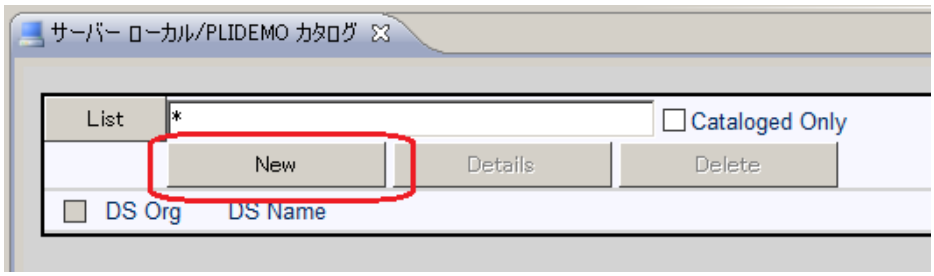
本チュートリアルでは使用する例題 JCL ではプロシージャを使用しています。Enterprise Server ではジョブプロシージャは区分データセットのメンバーとして配置します。このためまずプロシージャライブラリを作成し、プロシージャを配備しておきます。

1) サーバーエクスプローラ内で PLIDEMO を右クリックし [カタログ表示] を選択します。

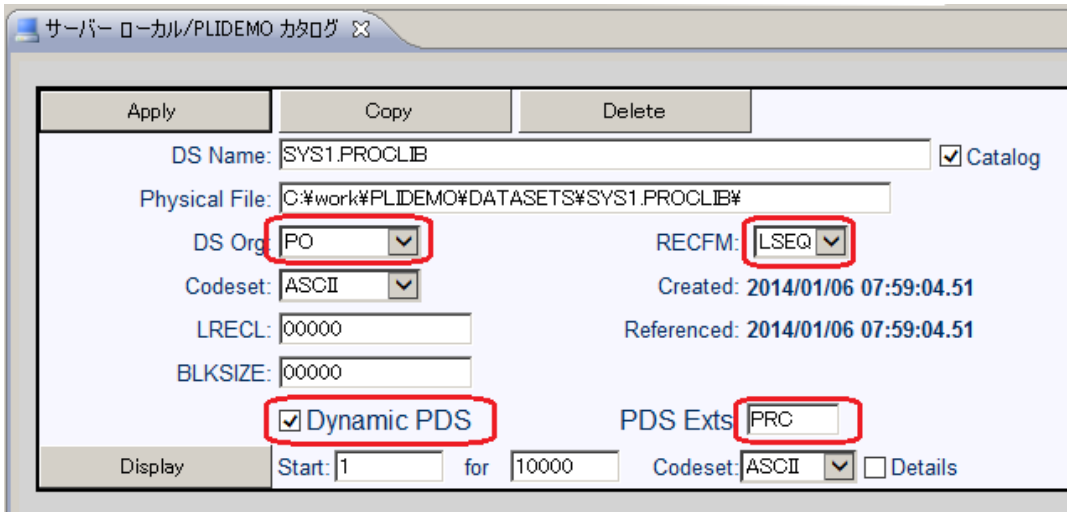
2) 以下のようにカタログビューが表示されます。ここで [List]ボタンをクリックします。



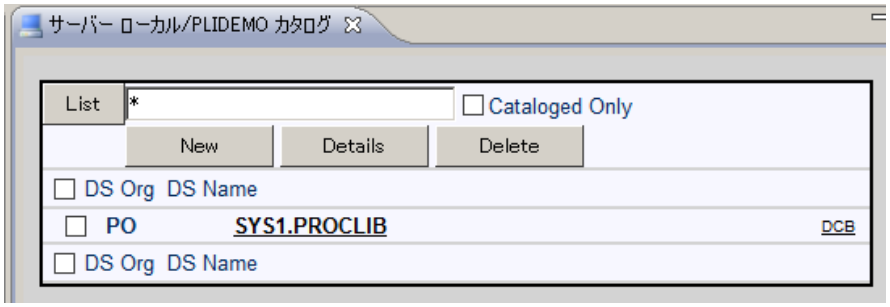
3) 現在カタログされているデータセットは何もありません。そこで [New] ボタンをクリックします。



4) 以下のカタログエントリの新規作成ダイアログが現れます。以下のように入力して [Apply] をクリックします。「PO」はパーティションドデータセットであることを示します。また、この PO が拡張子 .PRC のテキストファイルをフォルダ配下に保持する動的 PDS であることを指定しています。

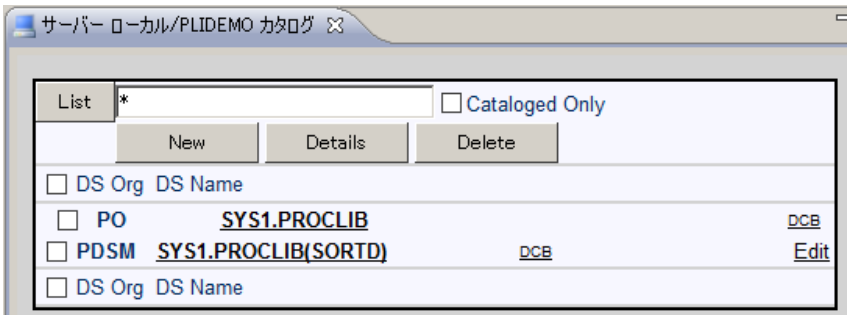


5) カタログビューを再表示すると、以下のようにカタログエントリ SYS1.PROCLIB が作成されています。

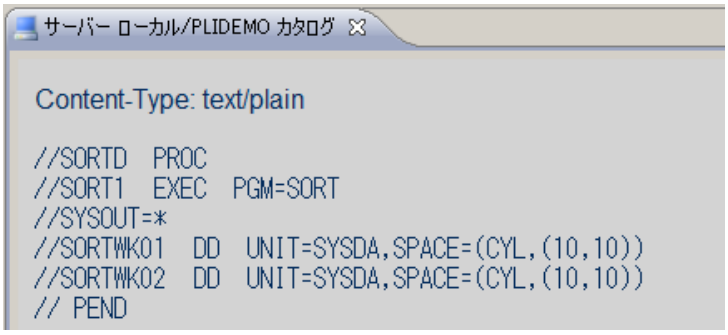


6) 指定した物理フォルダ C:¥work¥PLIDEMO¥DATASETS¥SYS1.PROCLIB を Windows エクスプローラで作成し、その下に C:¥PLIJCLtutorial¥SORTD.prc をコピーします。

7) カタログビューで SYS1.PROCLIB をクリックします。以下のようにメンバー SYS1.PROCLIB(SORTD) が登録されたことが確認できます。



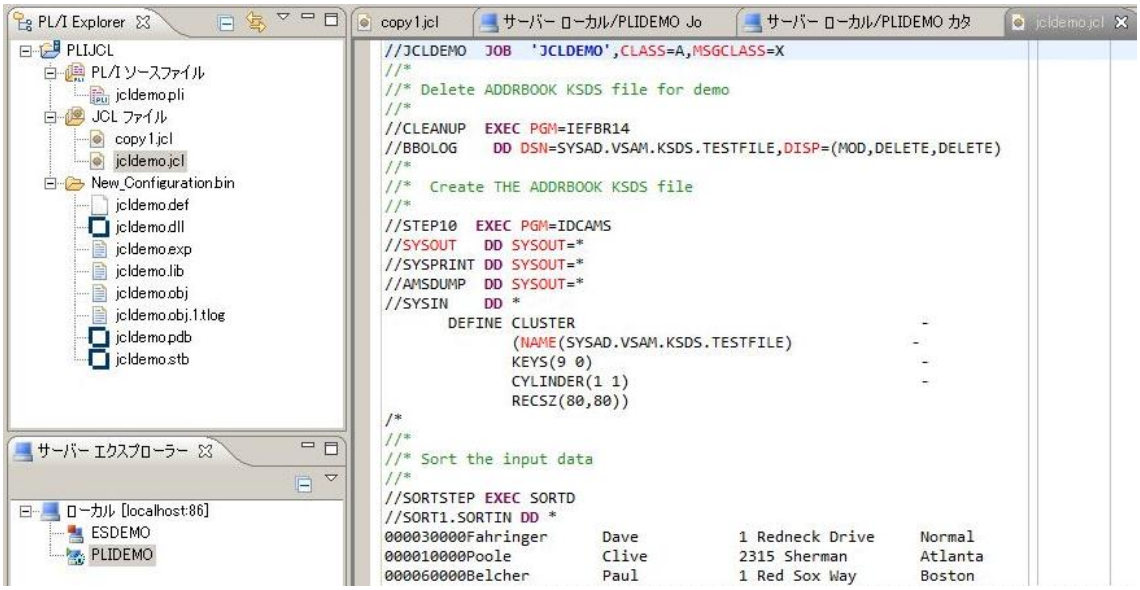
8) SYS1.PROCLIB(SORTD) をクリックすると以下のようにその内容を表示させることができます。



9.9 PL/I バッチプログラムの実行

PL/I プログラムを含むより実践的なジョブを実行してみます。

1) PL/I エクスプローラ内で jcldemo.jcl をダブルクリックしエディタで開きます。



このジョブは4つのステップから構成され下記の様に連携されています。

- STEP1: CLEANUP

IEFBR14 を使用してデモで使用する KSDS ファイルをからログから削除しています。

- STEP2: STEP10

IDCAMS を使用してデモで使用する KSDS ファイルを新規作成しています。

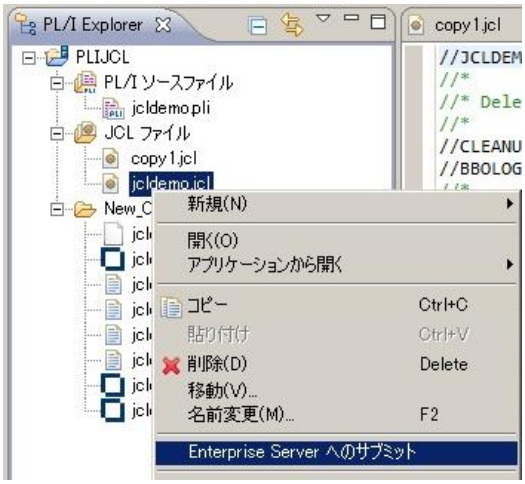
- STEP3: SORTSTEP

事前に登録したカタログ式プロセージャSORTDを使用して &&ADDRDATファイルへの書き込み用データをソートして出力しています。

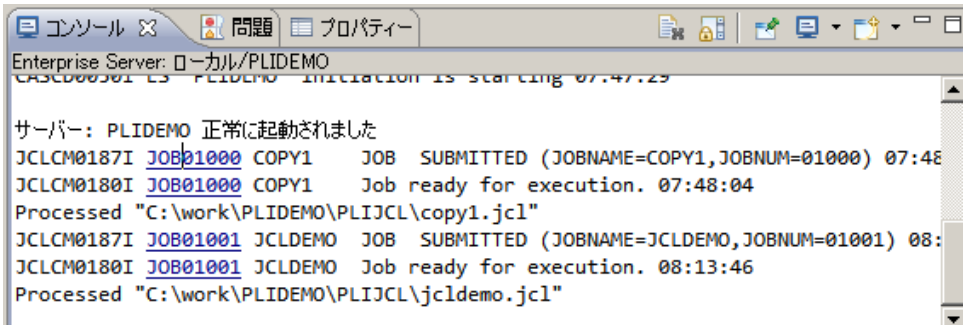
- STEP4: STEP20

PL/I プログラムを起動し &&ADDRDAT ファイルから入力したソート済みレコード KSDS ファイルに書き込んでいます。JCLDEMO.pli のソースも一読してください。

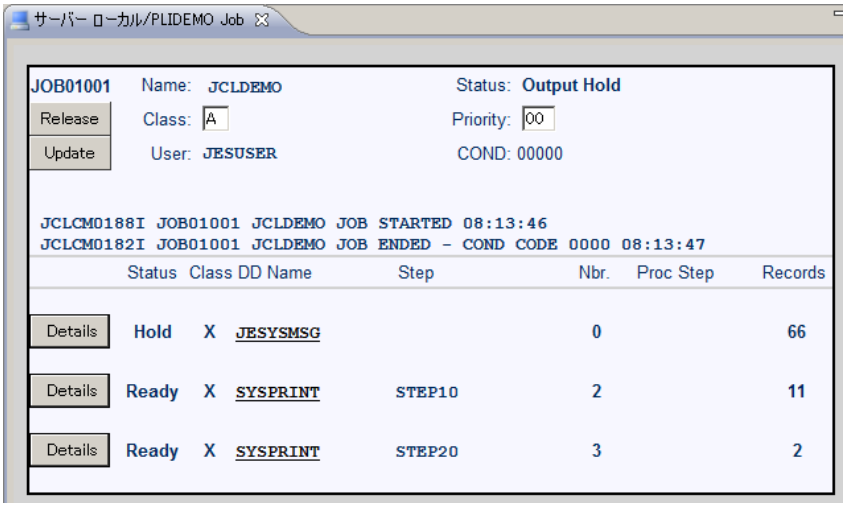
2) PL/I エクスプローラ内で jcldemo.jcl を右クリックして [Enterprise Server へのサブミット] を選択します。



3) 以下のように JCLDEMO ジョブが実行された旨のメッセージがコンソールに表示されます。



4) コンソール中の JOB01001 をクリックすると、以下のようにスプール出力の一覧が表示されます。ここでジョブの実行結果を確認してゆきます。



5)　JESYSMSG をクリックすると以下のようにジョブログが表示されます。

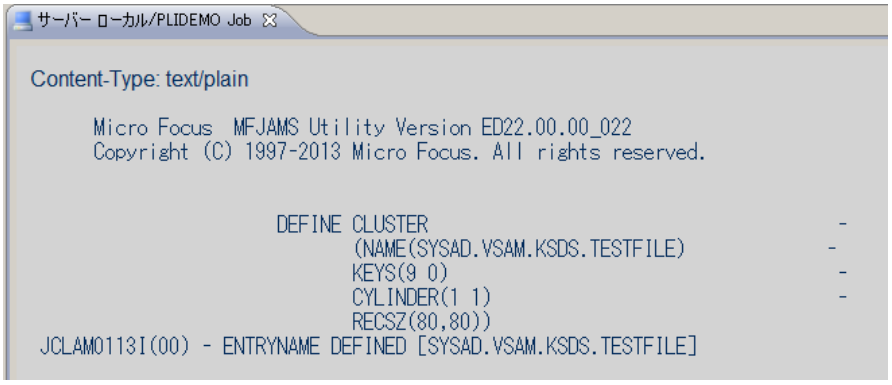
サーバー ローカル/PLIDEMO Job

```
Content-Type: text/plain

     *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-
     *-*-*  Micro Focus ESJCL  ASCII  JES2 Version ED22.00.00_022
     *-*-*  Copyright (C) 1997-2013 Micro Focus. All rights reserved.
     *-*-*  Job: 01001 Name: JCLDEMO  User: JESUSER  Date: 01/06/14 Time: 08:13:46
     *-*-*  File: $TXRFDIR/T000000185.T
     *-*-*  DSN:
     *-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-

   1 //JCLDEMO   JOB  'JCLDEMO',CLASS=A,MSGCLASS=X
   2 //*
   3 //* Delete ADDRBOOK KSDS file for demo
   4 //*
   5 //CLEANUP   EXEC PGM=IEFBR14
   6 //BBOLOG     DD DSN=SYSAD.VSAM.KSDS.TESTFILE,DISP=(MOD,DELETE,DELETE)
   7 //*
   8 //*  Create THE ADDRBOOK KSDS file
   9 //*
  10 //STEP10   EXEC PGM=IDCAMS
  11 //SYSOUT    DD SYSOUT=*
  12 //SYSPRINT DD SYSOUT=*
  13 //AMSDUMP   DD SYSOUT=*
  14 //SYSIN     DD *
  21 //*
  22 //* Load the ADDRBOOK KSDS using a PLI Program
  23 //*
  24 //STEP20   EXEC PGM=JCLDEMO
  25 //SYSOUT     DD SYSOUT=*,HOLD=Y
  26 //SYSPRINT  DD SYSOUT=*
```

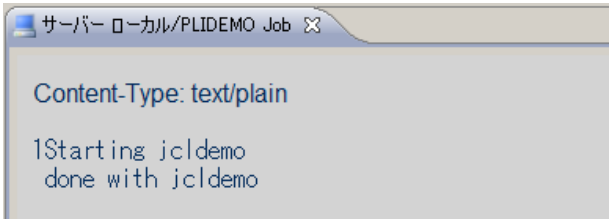
6)　以下のように各ジョブステップが正常終了している履歴が確認できます。ジョブが異常終了した場合にはここでエラーの原因を調査することができます。

```
**** JCLCM0180I Job ready for execution.
**** Execution on Server PLIDEMO   Process      7024

     08:13:46 JCLCM0188I JOB  STARTED

     08:13:46 JCLCM0190I STEP STARTED   CLEANUP
     SYSAD.VSAM.KSDS.TESTFILE                                        BBOLOG
      C:¥WORK¥PLIDEMO¥DATASETS¥SYSAD.VSAM.KSDS.TESTFILE.DAT            DELETED
---> 08:13:46 JCLCM0191I STEP ENDED     CLEANUP - COND CODE 0000

     08:13:46 JCLCM0190I STEP STARTED   STEP10
     08:13:46 JCLCM0199I Program MFJAMS   is COBOL      ASCII  Big-Endian    NOAMODE.
     MFE2014.S0106.S081346.J01001.D00001.SYSOUT                      SYSOUT
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*081346.J01001.D00001.SYSOUT.DAT  REMOVED
     MFE2014.S0106.S081346.J01001.D00002.SYSPRINT                    SYSPRINT
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*1346.J01001.D00002.SYSPRINT.DAT  SPOOLED
     MFE2014.S0106.S081346.J01001.D00003.AMSDUMP                     AMSDUMP
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*81346.J01001.D00003.AMSDUMP.DAT  REMOVED
     MFE2014.S0106.S081346.J01001.D00004.SYSIN                       SYSIN
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*S081346.J01001.D00004.SYSIN.DAT  DELETED
---> 08:13:46 JCLCM0191I STEP ENDED     STEP10 - COND CODE 0000

     08:13:46 JCLCM0190I STEP STARTED   STEP20
     08:13:47 JCLCM0199I Program JCLDEMO  is Open PL/I   ASCII  Big-Endian    NOAMODE.
     MFE2014.S0106.S081346.J01001.D00005.SYSOUT                      SYSOUT
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*081346.J01001.D00005.SYSOUT.DAT  REMOVED
     MFE2014.S0106.S081346.J01001.D00006.SYSPRINT                    SYSPRINT
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*1346.J01001.D00006.SYSPRINT.DAT  SPOOLED
     MFE2014.S0106.S081346.J01001.D00007.SYSUT1                      SYSUT1
      C:¥WORK¥PLIDEMO¥DATASETS¥MFE2014*081346.J01001.D00007.SYSUT1.DAT  DELETED
     SYSAD.VSAM.KSDS.TESTFILE                                        SYSUT2
      C:¥WORK¥PLIDEMO¥DATASETS¥SYSAD.VSAM.KSDS.TESTFILE.DAT            RETAINED
---> 08:13:47 JCLCM0191I STEP ENDED     STEP20 - COND CODE 0000

---> 08:13:47 JCLCM0182I JOB  ENDED   - COND CODE 0000
```

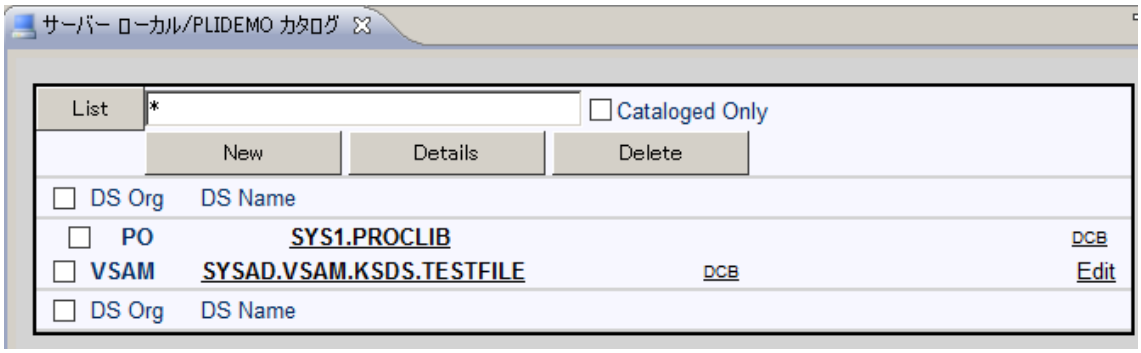
7) 4) の画面に戻り STEP10 の SYSPRINT をクリックすると、以下のように IDCAMS の実行結果を確認することができます。



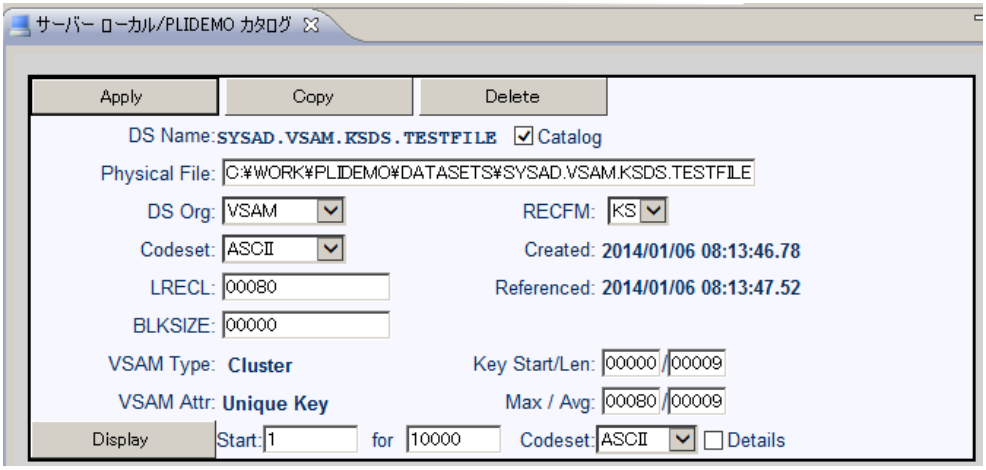
8) 4) の画面に戻り STEP20 の SYSPRINT をクリックすると、以下のように PL/I プログラムの PUT 文で書かれた内容を確認することができます。



9) 続いてこのジョブの実行によってカタログされたデータセットを見てみます。サーバーエクスプローラ内で PLIDEMO を右クリックし [カタログ表示] を選択します。 [List] ボタンをクリックすると VSAM ファイル SYSAD.VSAM.KSDS.TESTFILE がカタログされていることがわかります。



10) 右端の [DCB] をクリックすると以下のように DCB 情報が表示されます。



11) [Display]ボタンをクリックすると以下のようにデータセットの内容が表示されます。ソートされた順に VSAM レコードが格納されていることを確認できます。

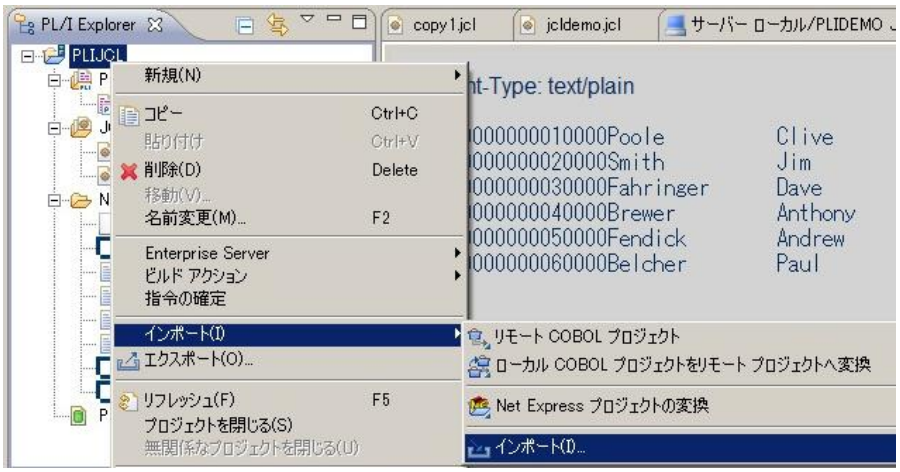
9.10 PL/I バッチプログラムのデバッグ

ただ今実行した JCL 中から起動されている PL/I プログラムをステップ実行でデバッグする方法を示します。

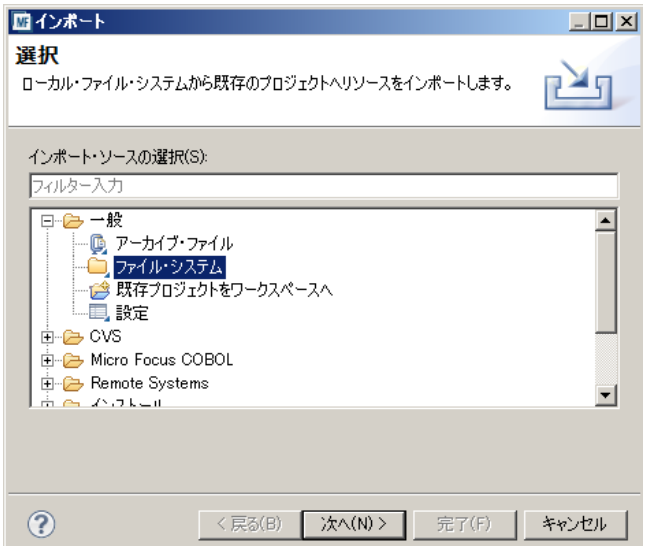
JES リージョン配下で実行される PL/I プログラムのデバッグは、PLIDEBUG.DAT という構成ファイルで指示します。インストール直後は存在しませんが PL/I プログラムを最初に実行した際に作成されます。PL/I エクスプローラ内に表示されていない場合は、PLIJCL を右クリックし、[リフレッシュ] を選択します。これは主キーが先頭 8 バイトの索引ファイルであり、このキー部分でデバッグ対象の PL/I プログラム名を指定します。

本チュートリアルでは構成済みの PLIDEBUG.DAT ファイルを使用します。

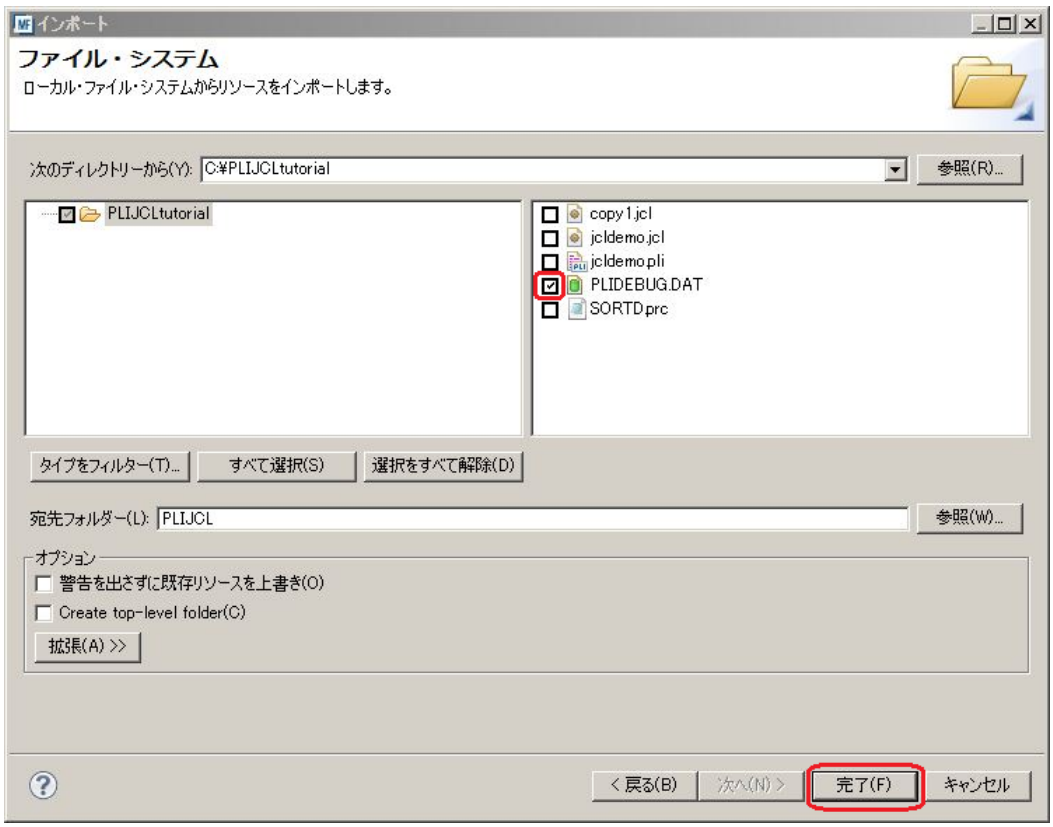
1)　PL/I エクスプローラ内で PLIJCL を右クリックして [インポート] > [インポート] を選択します。



2)　以下のダイアログで [一般] > [ファイルシステム] を選択し、[次へ] をクリックします。

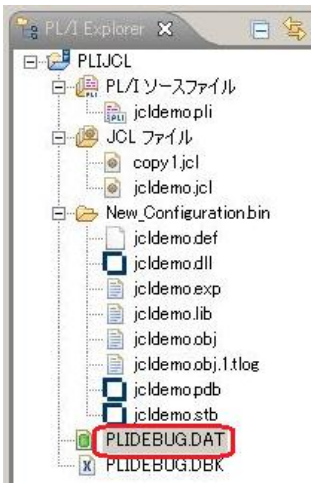


3) 以下のダイアログで [参照…] ボタンをクリックして C:¥PLIJCLtutorial を選択し、以下のように PLIDEBUG.DAT ファイルをチェックオンし、[完了] をクリックします。

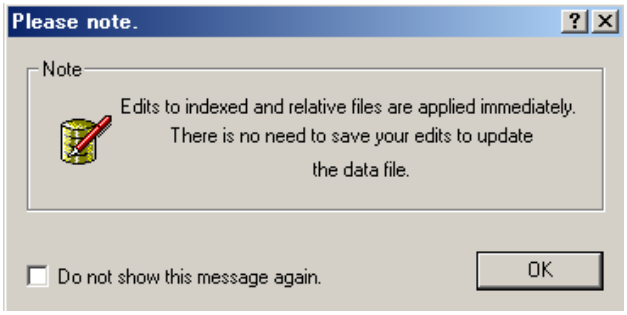


4) 以下のダイアログでは [はい] をクリックします。

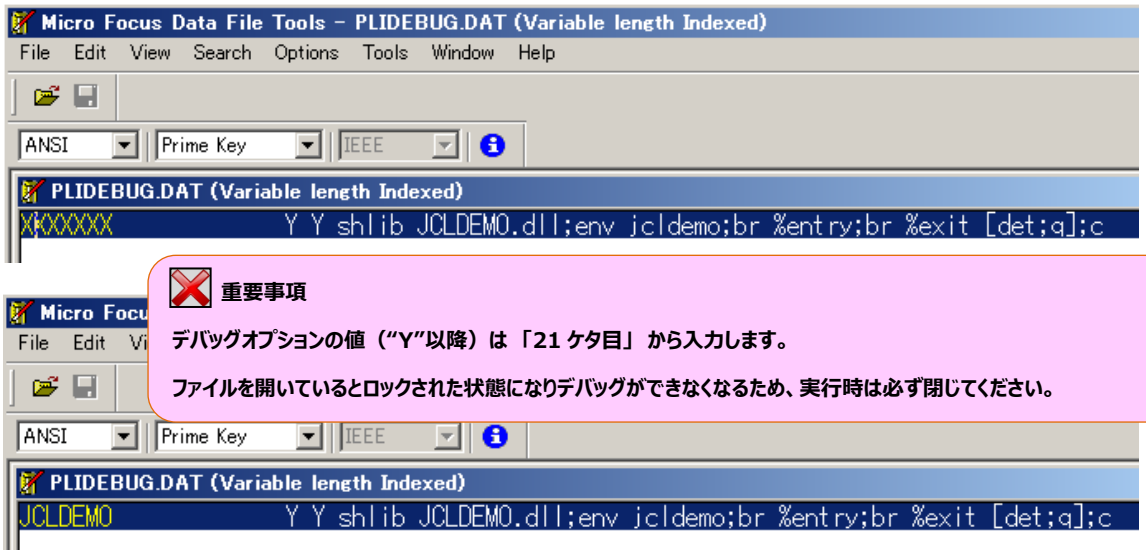
5) PL/I エクスプローラ内で PLIDEBUG.DAT をダブルクリックします。



6) 以下のダイアログに対して [OK] をクリックします。



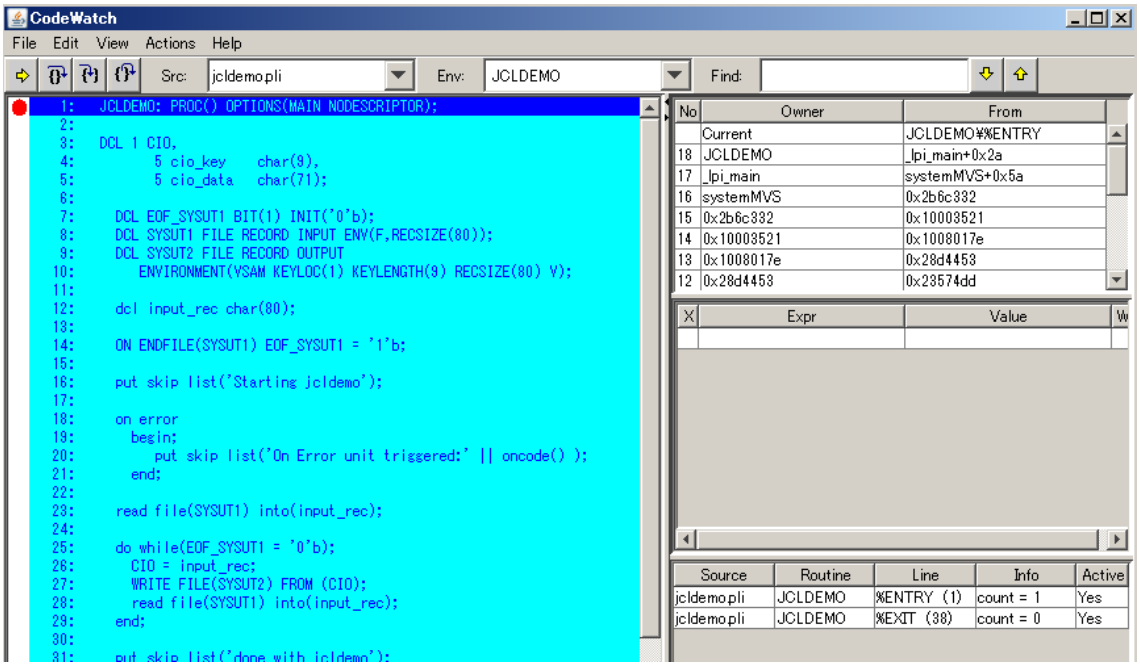
7) 以下のように 1 件のレコードが表示されます。主キーの値はデバッグ対象プログラム名を表します。”XXXXXXX” となっていますので、これを “JCLDEMO” に書き換えます。これで JCLDEMO.pli が実行されると自動的にデバッグセッションが開始するように設定されます。レコードの残りの部分はデバッグのオプションを示しています。

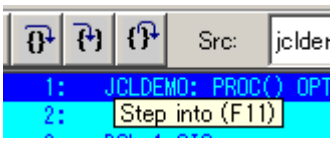


8) [File] > [Exit] でデータファイルエディタを終了します。

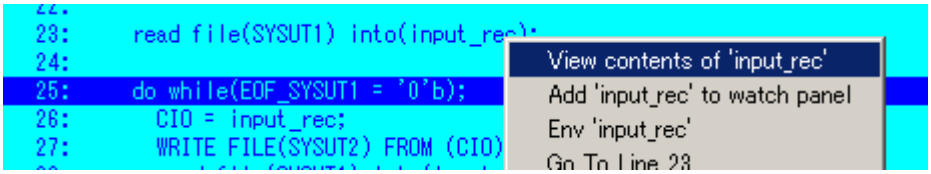
9) PL/I エクスプローラ内で jcldemo.jcl を右クリックして [Enterprise Server へのサブミット] を選択します。

10) 以下の通り PL/I デバッガである CodeWatch が起動します。



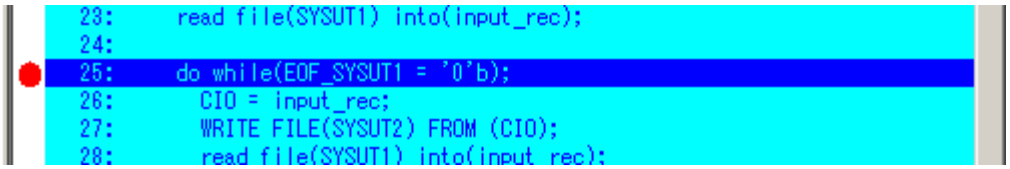
11) 左上のボタンバーの中から [Step Into] 

または F11 をクリックすると PL/I のステートメントを1ステップ実行して進めます。繰り返しクリックして以下の Read ステートメントの次の Do まで実行を進めます。SYSUT1 から読まれたレコードを格納している input_rec を以下のように右クリックし、[View contents of …] を選択します。



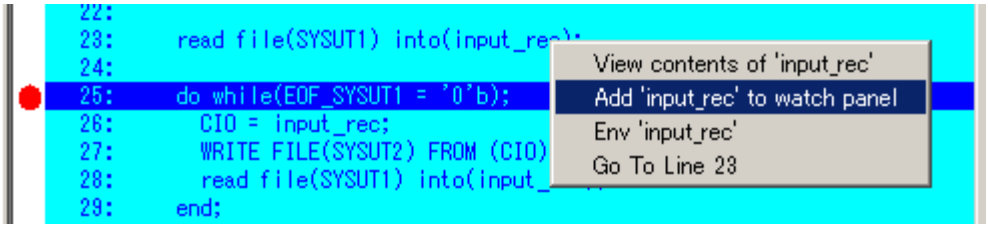
12) 以下のように読まれたレコードの内容がポップアップ表示されます。

13) プログラム中にブレークポイントを設定してみましょう。以下のように現在の実行点である Do ステートメントの行をダブルクリックすると以下のように該当行の左端に赤丸が表示されます。

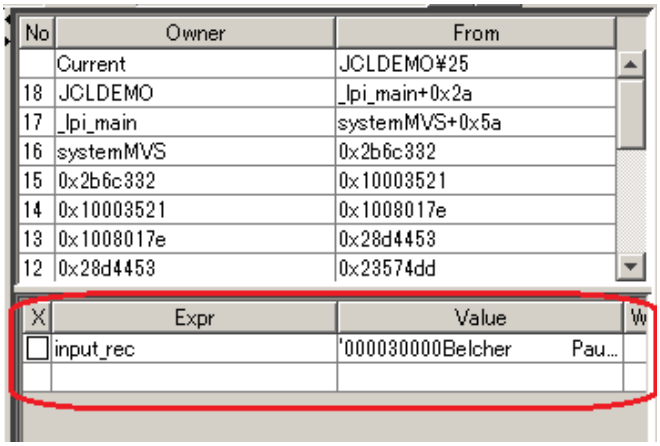


14) これで Do ステートメントにブレークポイントが設定されました。ここでボタンバー内の [Continue] ボタンまたは F5 をクリックします。

15) 以下のように 2 件目のレコードの読み込み後の状態でブレークします。再び input_rec を右クリックして今度は [Add … to watch panel]を選択します。



16) 以下のように input_rec の内容がウォッチパネルに表示されます。



17) F5を繰り返し押下すると 6 件のレコードが最後まで読まれてプログラムが終了し、それに伴ってジョブが終了し、デバッガが閉じます。

18) サーバーエクスプローラ内で PLIDEMO を右クリックして [停止] を選択します。

19) PLIDEMO リージョンが停止したら Enterprise Developer for Eclipse を終了します。

以上で PL/I バッチチュートリアルを終了します。