Micro Focus Enterprise Developer チュートリアル

メインフレーム PL/I 開発：JCL

Eclipse 編

1. 目的

本チュートリアルでは、PL/I 言語で書かれたソースをオープン環境へ移行後、Eclipse を使用したプロジェクトの作成、コンパイル、JCL の実行、デバッグまでを行い、その手順の習得を目的としています。

2. 前提

- Windows 開発環境に Enterprise Developer 4.0 for Eclipse がインストール済であること。

3. チュートリアル手順の概要

1. チュートリアルの準備
2. Eclipse の起動
3. メインフレーム PL/I プロジェクトのインポート
4. プロジェクトプロパティの確認
5. ビルドの実行
6. Enterprise Server インスタンスの設定
7. Enterprise Server インスタンス開始と確認
8. JCL の実行
9. PL/I ソースのデバッグ
10. 終了処理

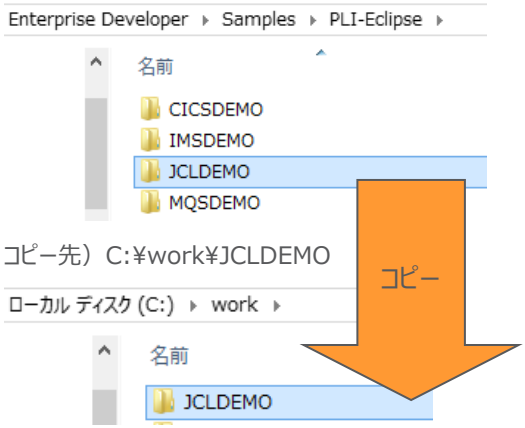
3.1 チュートリアルの準備

例題プログラムに関連するリソースを用意します。

1) Eclipse のワークスペースで使用する work フォルダを C ディレクトリ直下に作成します。

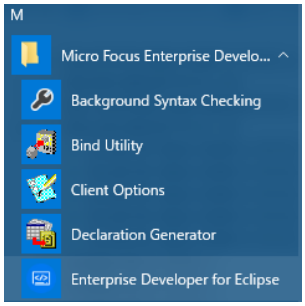
2) 製品をインストールしたフォルダ配下に含まれている例題プログラム JCLDEMO フォルダを、作成した C:¥work へコピーします。

例）C:¥Users¥Public¥Documents¥Micro Focus¥Enterprise Developer¥Samples¥PLI-Eclipse¥JCLDEMO

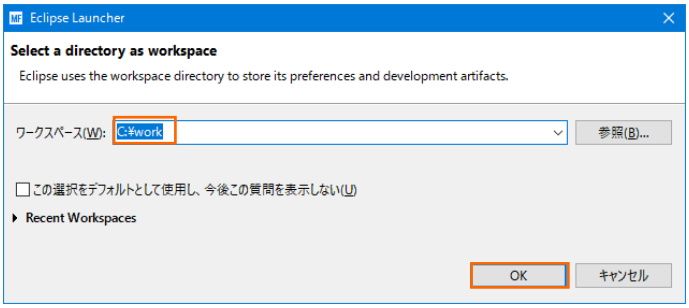


3.2 Eclipse の起動

1) Micro Focus Enterprise Developer for Eclipse を起動します。

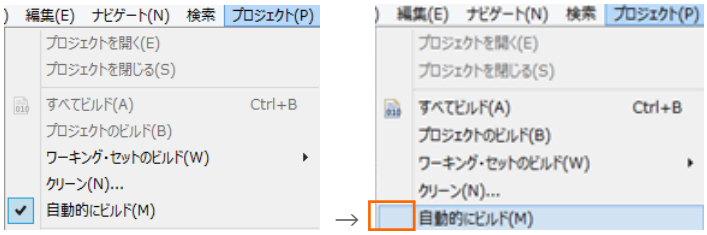


2) 前項で作成した C:¥work をワークスペースへ指定して、[OK] ボタンをクリックします。

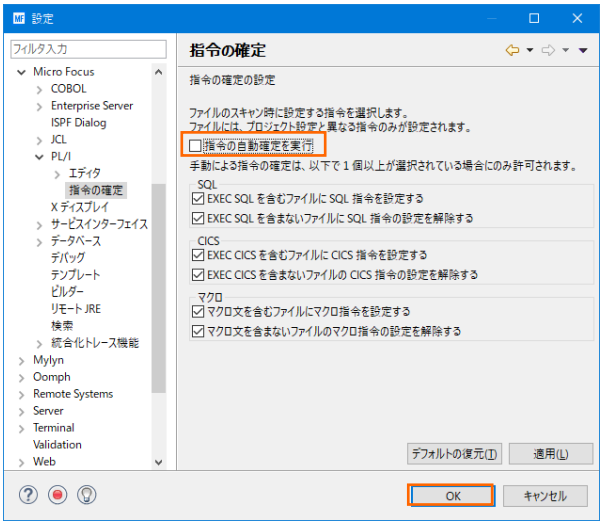


3) [ようこそ] タブが表示されますので、[Open PL/I Perspective] をクリックして、PL/I パースペクティブを開きます。

4) PL/I パースペクティブ表示後、[プロジェクト] プルダウンメニューの [自動的にビルド] を選択して、これをオフにします。

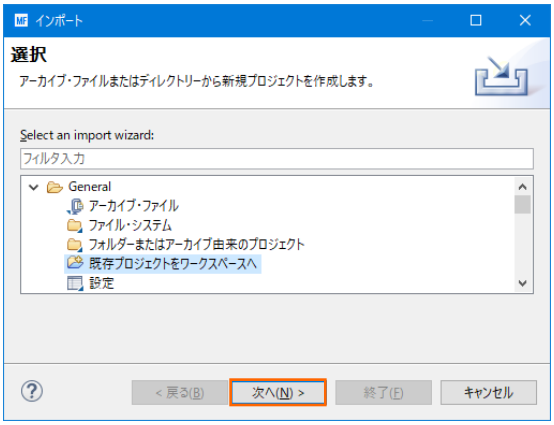


5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。 [ウィンドウ] プロダウンメニューの [設定] > [Micro Focus] > [PL/I] > [指令の確定] > [指令の確定を行う] チェックボックスをオフにして [OK] ボタンをクリックします。

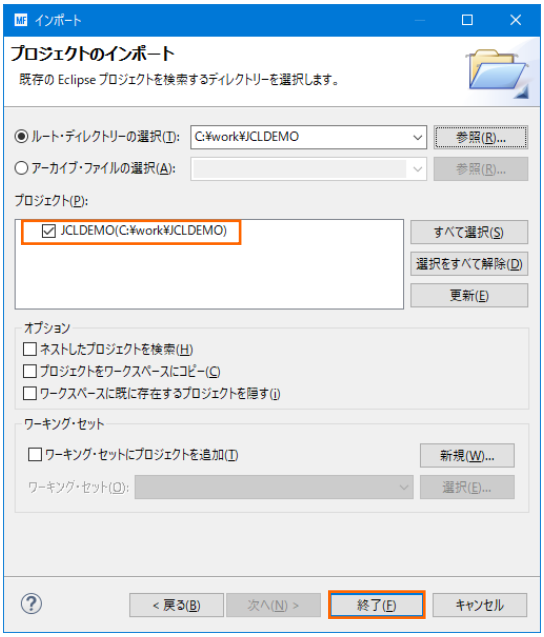


3.3 メインフレーム PL/I プロジェクトのインポート

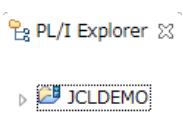
1) 用意した例題プロジェクトをインポートします。 [ファイル] プルダウンメニューから [インポート] を選択し、インポートウィンドウにて [General] > [既存プロジェクトをワークスペースへ] を選択後 [次へ] ボタンをクリックします。



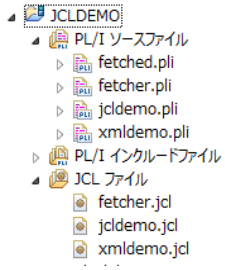
2) [ルート・ディレクトリの選択] へ C:¥work¥JCLDEMO を指定すると、このフォルダに含まれるプロジェクトが表示されます。チェックをオンにした状態で [終了] ボタンをクリックします。



3) PL/I エクスプローラーにインポートしたプロジェクトが表示されます。



4) JCLDEMO プロジェクトを展開すると PL/I ソースや JCL などが確認できます。



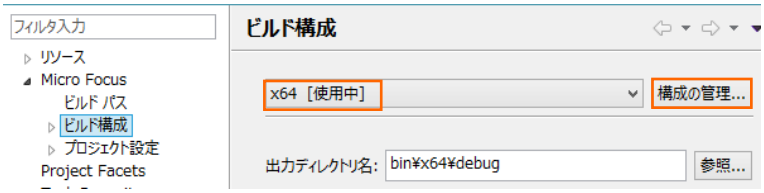
3.4 プロジェクトプロパティの確認

プロジェクトの設定値を確認していきます。

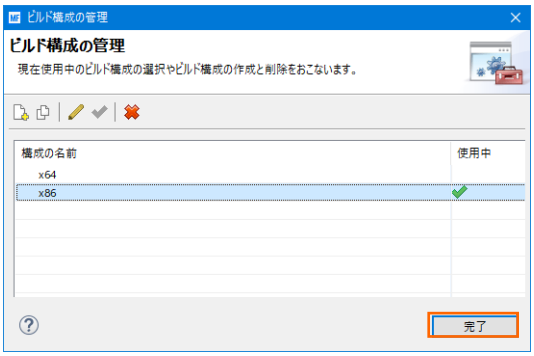
1) JCLDEMO プロジェクトを右クリックして [プロパティ] を選択するとプロパティウィンドウが表示されます。

64-bit 稼働が指定されていますが、ここでは 32-bit OS で実行することを考慮して 32-bit 稼働へ変更します。

① [Micro Focus] > [ビルド構成] で [構成の管理] ボタンをクリックして構成管理ウィンドウを表示します。

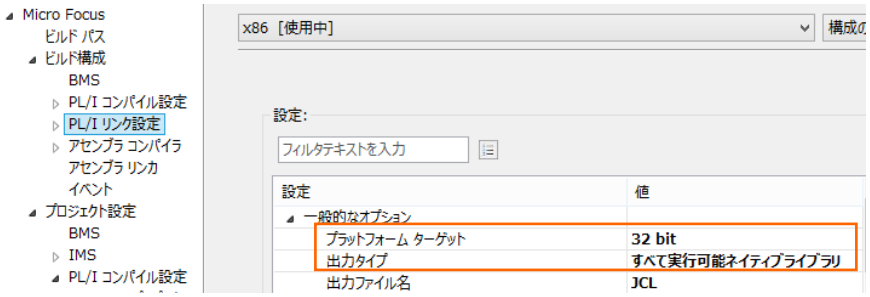


② [ビルドの構成管理] ウィンドウでは [x86] のチェックボックスをオンにして [完了] ボタンをクリックします。

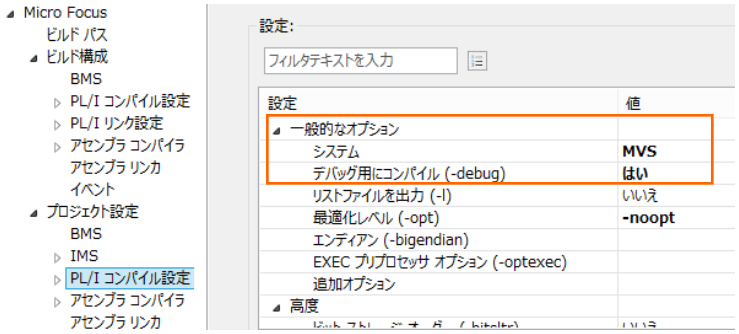


③ [Micro Focus] > [ビルド構成] ウィンドウへ戻り [x86] へ変更されたことと、プロジェクト配下の bin¥debug フォルダへ実行ファイルが出力されることを確認後 [適用] ボタンをクリックします。

④ [Micro Focus] > [ビルド構成] > [PL/I リンク設定] を選択して内容を確認すると、32 ビット稼働する実行可能ネイティブライブラリを実行ファイルタイプとして生成することがわかります。

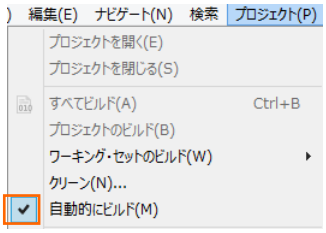


2) [Micro Focus] > [プロジェクト設定] > [PL/I コンパイル設定] を選択して内容を確認すると、例題の内容に沿って、「システム」には MVS が設定されており、デバッグ実行用ファイルを生成することがわかります。確認後、[OK] ボタンをクリックします。

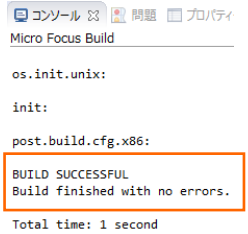


3.5 ビルドの実行

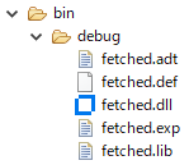
1) [プロジェクト] プルダウンメニューの [自動的にビルド] を選択して、これをオンすると自動的にビルドが実行されます。



2) コンソールタブでビルドの成功を確認します。

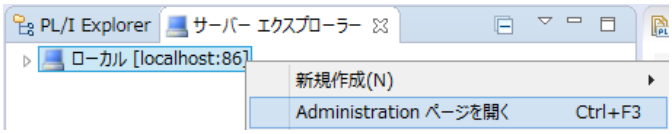


3) プロジェクトの bin¥debug フォルダ配下に目的の実行ファイルが作成されていることを確認してください。



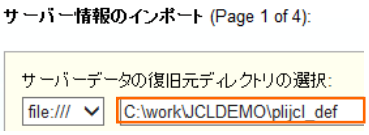
3.6 Enterprise Server インスタンスの設定

1) PL/I を実行するためのエンジンを搭載した Enterprise Server インスタンスを作成します。 サーバー エクスプローラータブの 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。

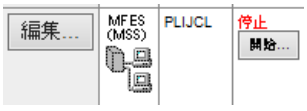


2) C:¥work¥JCLDEMO には Enterprise Server インスタンスのサンプルが含まれており、これをインポートします。PL/I アプリケーションは 32 ビット稼働を指定したため、C:¥work¥JCLDEMO¥plijcl_def がインポート対象となります。64 ビットで稼働させる場合は C:¥work¥JCLDEMO¥plijcl64_def をインポートしてください。

Enterprise Server Administration 画面左側の [インポート] をクリックして、表示される下記項目へ前述のパスを入力後、 [次へ] ボタンをクリックします。



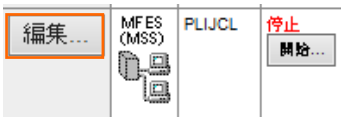
3) 画面の Page 2/4、3/4、ではそのまま [次へ] ボタンを、Page 4/4 では [OK] ボタンをクリックすると、PLIJCL という名前の 32 ビットアプリケーション稼働用 Enterprise Server インスタンスが追加されます。



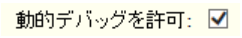
重要

アプリケーション稼働ビット数 = Enterprise Server インスタンス稼働ビット数である必要があります。

4) 設定を変更するため、[編集] ボタンをクリックします。



5) [サーバー] > [プロパティ] > [一般] タブで表示される画面の [動的デバッグを許可] 欄のチェックをオンにします。これにより、Eclipse からのデバッグが行えます。



6) 前項と同じタブの [構成情報] 欄を下記のように入力し、[適用] ボタンをクリックします。

変更前；

```
[ES-Environment]
JDEMO=C:¥Users¥Public¥Documents¥Micro Focus¥Enterprise Developer¥Samples¥"PLI-VS or PLI-Eclipse"¥JCLDEMO
CODEWATCH_NOTIF=Y
```

変更後；

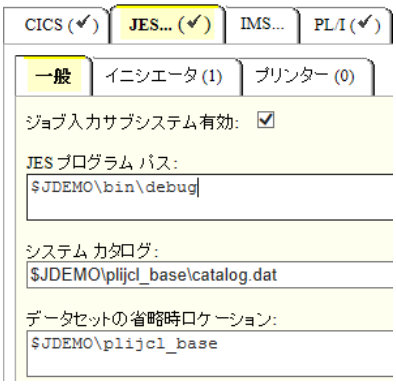
```
[ES-Environment]
JDEMO=C:¥work¥JCLDEMO
```

情報

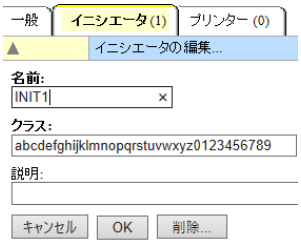
CODEWATCH_NOTIF 環境変数について：
CodeWatch デバッガを使用する際に開発用インスタンスに指定します。本番インスタンスにはパーフォーマンスの観点から排除することを推奨します。ここでは Eclipse 上のデバッガを使用するため削除します。

7) [サーバー] > [プロパティ] > [MSS] > [JES] > [一般] タブで表示される画面の各項目を確認します。

項目名	説明
メインフレーム サブシステム サポート有効	[MSS] タブ配下の設定をオン、オフ指定します。
ジョブ入力サブシステム 有効	[JES] タブ配下の設定をオン、オフ指定します。
JES プログラム パス	実行ファイルのパスを指定します。
システムカタログ	カタログファイルのパスとファイル名称を指定します。
データセットの省略時ロケーション	JCL などで指定するファイルのデフォルトパスを指定します。

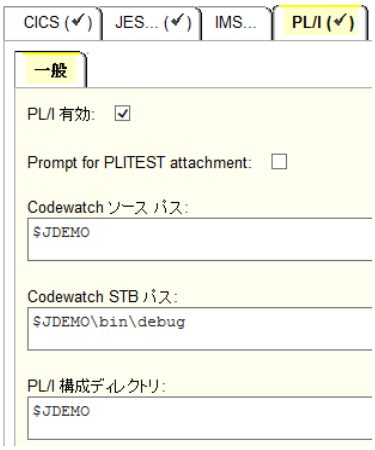


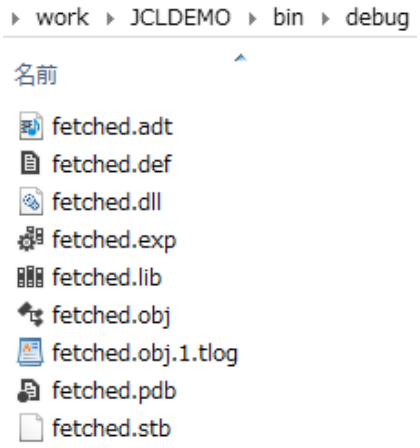
8) [サーバー] > [プロパティ] > [MSS] > [JES] > [イニシエータ] タブでイニシエータ定義を確認します。 A ～ 9 までのクラスに対するイニシエータが設定されています。



9) [サーバー] > [プロパティ] > [MSS] > [PL/I] > [一般] タブで表示される画面の各項目を確認します。

項目名	説明
PL/I 有効	[PL/I] タブ配下の設定をオン、オフ指定します。
CodeWatch ソース パス	CodeWatch デバッガで使用するソースファイルパスを指定します。
CodeWatch STB パス	CodeWatch デバッガで使用するデバッグファイルパスを指定します。
PL/I 構成ディレクトリ	プロジェクトのパスを指定します。



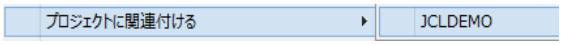


10) 画面左上の [Home] をクリックして一覧画面に戻ります。

3.7 Enterprise Server インスタンスの開始と確認

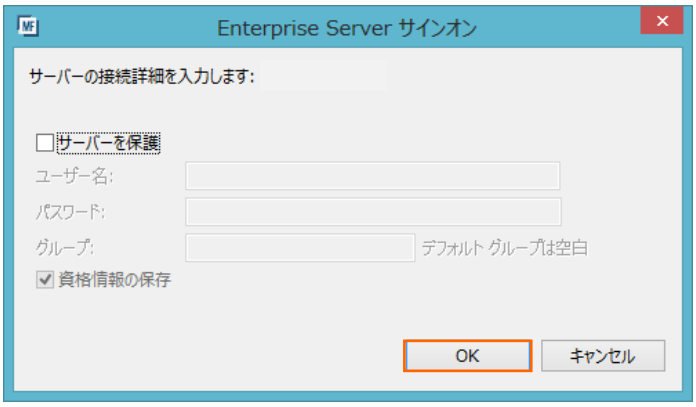
1) サーバーエクスプローラ内に PLIJCL インスタンスが表示されていることを確認します。表示されていない場合は [ローカル [localhost:86]] を右クリックし、[更新] を選択してリフレッシュしてください。

2) サーバーエクスプローラ内の PLIJCL インスタンスを右クリックし、[プロジェクトに関連付ける] > [JCLDEMO] を選択します。これにより Eclipse 内の JCLDEMO プロジェクトから実行される JCL は PLIJCL インスタンスで処理されることになります。

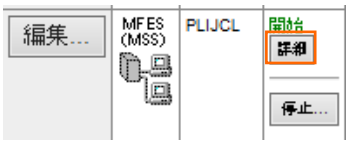


3) PLIJCL インスタンスを右クリックして [開始] を選択します。

4) 下記ウィンドウが表示された場合は、ここではユーザーによる制限を行わないため [OK] ボタンをクリックします。

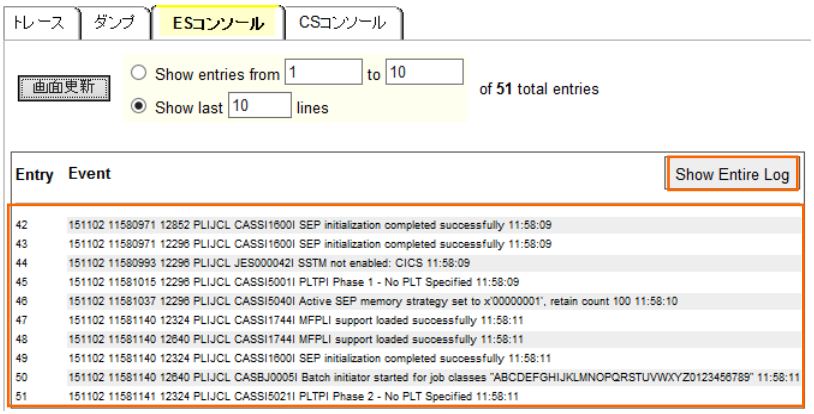


5) Enterprise Server Administration 画面へ移動して開始状態であることを確認後、[詳細] ボタンをクリックします。



6) [サーバー] > [診断] > [ES コンソール] で PLIJCL インスタンス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可能です。

正常に開始されたことを確認します。



注意

いくつかのサービス開始が失敗してもインスタンスは開始されますので、ログ内容を必ず確認してください。

3.8 JCL の実行

1) PL/I エクスプローラー内に存在する jcldemo.jcl をダブルクリックして内容を表示します。IDCAM などのユーティリティを使用してファイルを操作したのち、JCLDEMO プログラムを実行していることがわかります。

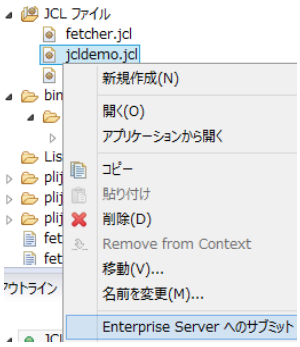
```
//*************************************************
//*  Run the JCLDEMO Program
//*************************************************
//STEP100  EXEC PGM=JCLDEMO
//SYSOUT   DD SYSOUT=*,HOLD=Y
//SYSPRINT DD SYSOUT=*,HOLD=Y,DCB=(RECFM=LSEQ)
//B1079256   DD DISP=(,CATLG),SPACE=(CYL,(5,5),RLSE),
//             DCB=(RECFM=FBA,LRECL=137,BLKSIZE=0),
//             DSN=SYSAD.STREAM.TEST
```

2) PL/I エクスプローラー内に存在する jcldemo.pli をダブルクリックして内容を表示します。CodeWatch デバッガを使用する際に利用するデバッグ文が含まれていますが、ここでは Eclipse を使用したデバッグを行うため利用しません。

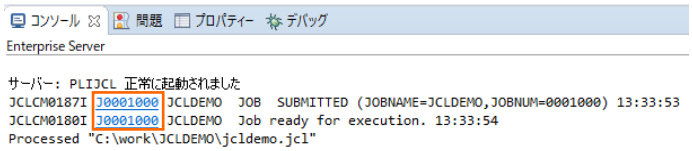
```
ON UNDEFINEDFILE (NOFILE)
   BEGIN;
     PUT SKIP LIST('ON UNIT UNDEFINED FILE - NOFILE Triggered');
     GOTO AfterUndef;
   END;

%if debugit %then
  %DO;
    CALL PLITEST('shlib jcldemo.dll;env JCLDEMO;br START_DEBUG;br AfterUndef;c','',1);
    START_DEBUG:
  %END;
```

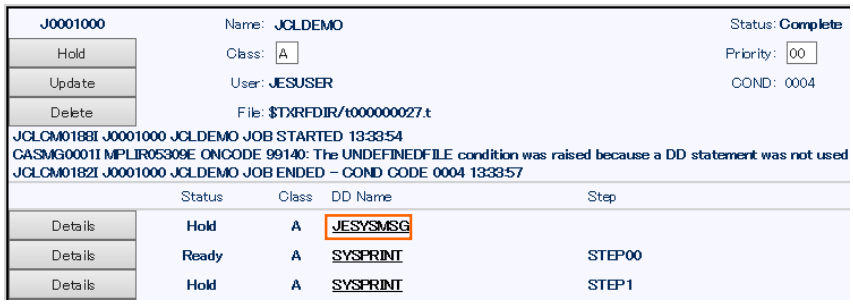
3) PL/I エクスプローラーから jcldemo.jcl を選択して右クリック後、[Enterprise Server へサブミット] を選択すると、この JCL が実行されます。



4) コンソールタブに下記が表示されますので、リンクをクリックします。



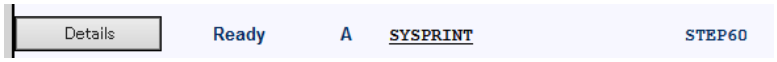
5) この JOB 番号にかかわるスプール一覧が表示されます。先頭の [JESYSMEG] をクリックしてジョブログを確認します。



6) ステップ名 STEP60 から STEP090 でリターンコードに 0004 が返却されていることがわかります。

```
---> 15:22:59 JCLCM0191I STEP ENDED     STEP60 - COND CODE 0004
```

7) STEP60 で何が発生したのか確認するために、右クリックで [前へ戻る] を選択し、スプール一覧から STEP60 の [SYSPRINT] をクリックします。



8) 最終行にワーニングが発生しており、JCL で指定した 100 件のレコードを下回ったため発生した警告と判断できます。

```
JCLAM0194W(04) - Number of records read was less than COUNT(00000100).
```

```
//SYSIN    DD *
     REPRO INFILE(IN) OUTFILE(OUT) COUNT(100)
/*
```

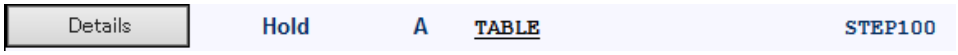
9) Jcldemo.jcl の STEP60 から STEP090 に記述されている COUNT(100) を COUNT(5) へ修正して保存すると、自動的にビルドがかかります。ビルド成功を確認後、JCL を再実行します。

```
//SYSIN    DD *
     REPRO INFILE(IN) OUTFILE(OUT) COUNT(5)
/*
```

10) 前項同様の手順で [JESYSMSG] 内容を確認すると、全てのステップが正常に終了していることがわかります。

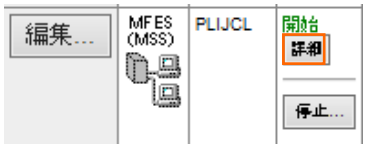
```
---> 16:31:06 JCLCM0191I STEP ENDED     STEP60 - COND CODE 0000
```

11) STEP100 では jcldemo.pli ソースから出力された内容が参照できますので、ソースコードと合わせて確認してみてください。

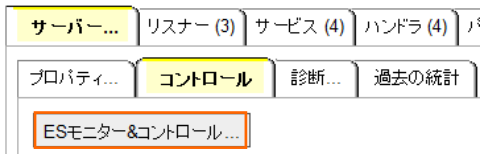


12) Enterprise Server Administration 画面へ移動後、画面左上 [Home] をクリックして一覧画面に戻ります。

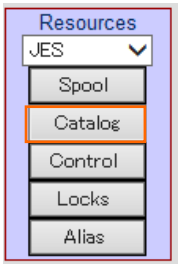
13) 実行された JCL から作成されたカタログ情報を確認します。[詳細] ボタンをクリックします。



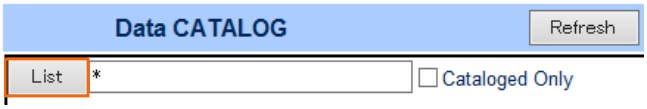
14) [サーバー] > [コントロール] > [ES モニター＆コントロール] ボタンをクリックします。



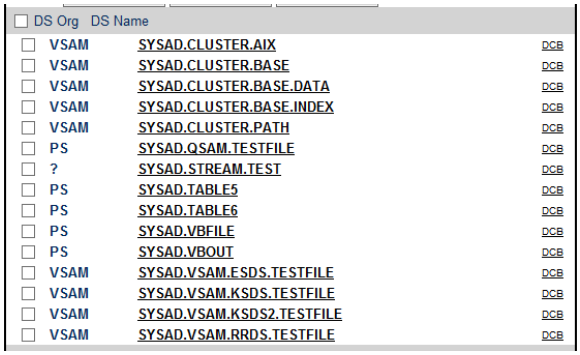
15) 画面左の中央部にある［Resources］直下のコンボボックスから［JES］を選択後、表示された［Catalog］ボタンをクリックします。前項で確認したスプールに関しても［Spool］ボタンをクリックすることにより、全てが参照可能になります。



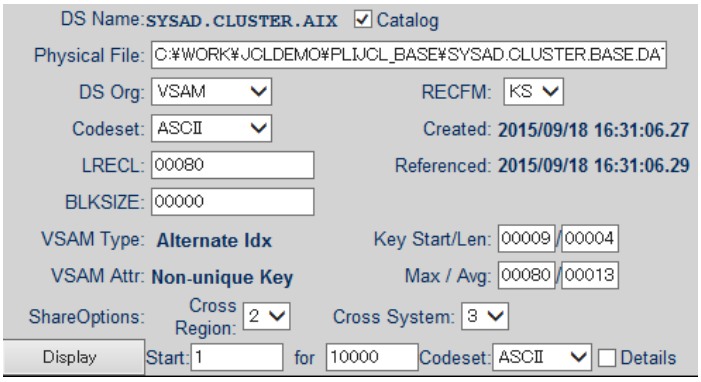
16) ［List］ボタンをクリックして、カタログ情報の一覧を表示します。



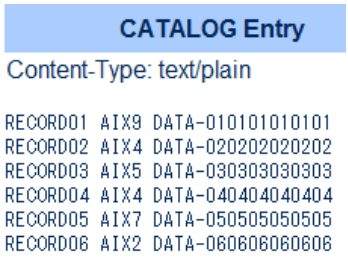
17) JCL の実行により作成されたカタログ情報が参照できます。



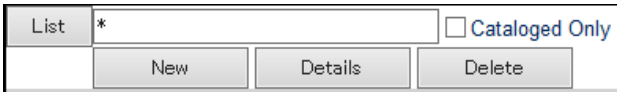
18) 画面右端の［DCB］をクリックするとカタログされたファイルの情報が表示され、変更も可能です。



19) カタログ情報一覧画面に中央部に表示されている DS Name をクリックするとデータが参照可能です。



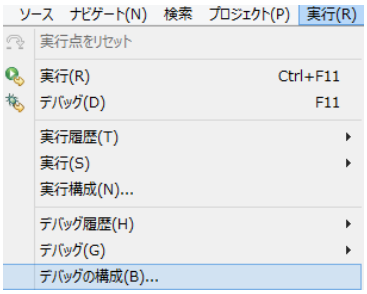
20) また、この画面からカタログの作成や削除も可能です。



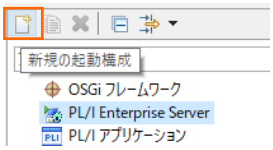
3.9 PL/I ソースのデバッグ

JCL から実行される PL/I プログラムをデバッグします。

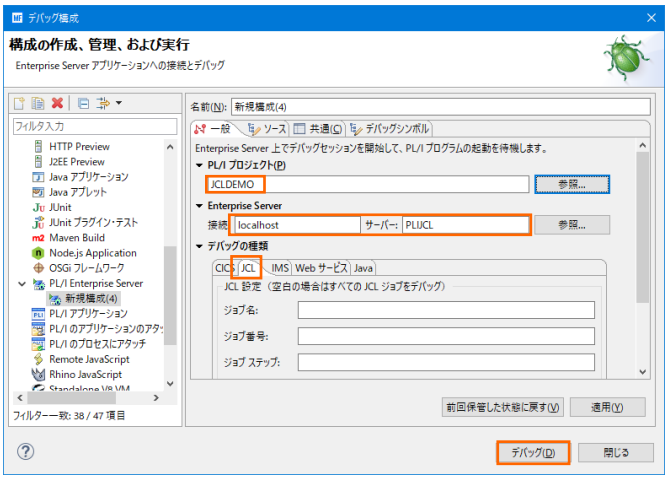
1) [実行] プルダウンメニューの [デバッグの構成] を選択します。



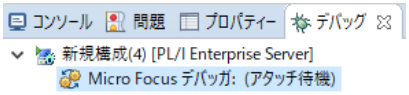
2) 左側のツリービューから [PL/I Enterprise Server] を選択して、左上の [新規の起動構成] アイコンをクリックします。



3) [PL/I プロジェクト] へ対象となる JCLDEMO プロジェクトを入力し、[Enterprise Server] へ実行させる PLIJCL インスタンスを指定します。[デバッグの種類] は「JCL」タブを選択した状態で、[デバッグ] ボタンをクリックします。

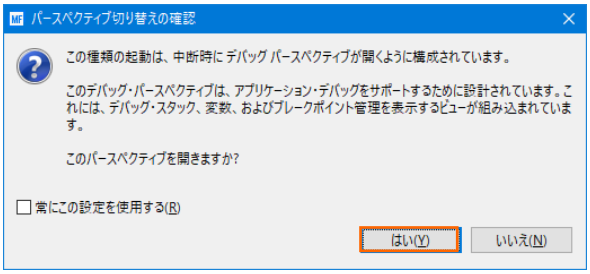


4) デバッグタブで [アタッチ待機] 状態になったことを確認します。

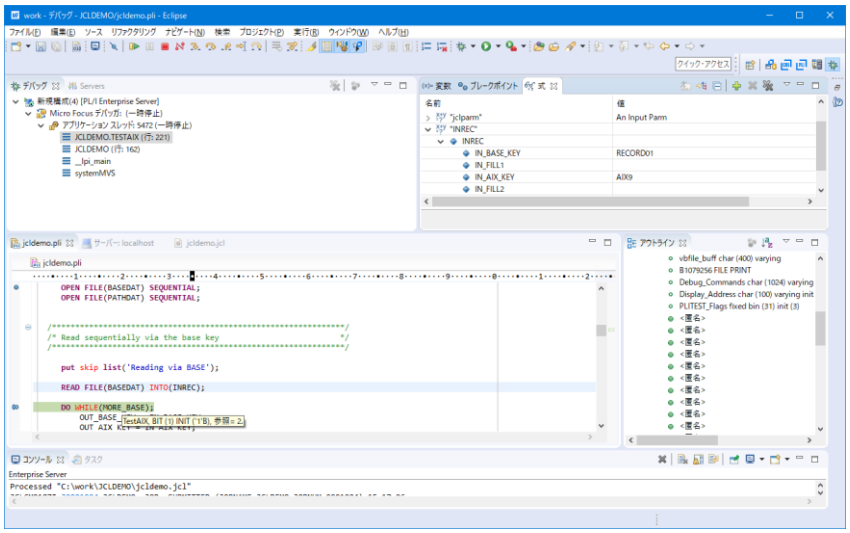


5) PL/I エクスプローラー内の jcldemo.jcl を右クリックして [Enterprise Server へのサブミット] を選択し、 JCL を実行します。

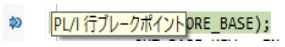
6) 再度、パースペクティブの切り替え確認ウィンドウが表示されますので、[はい] ボタンをクリックし、デバッグ用のパースペクティブを開きます。



7) 少し待つとデバッグセッションが開始して、プログラムのステップ実行が可能になります。[F5] キーもしくは [実行] プルダウンメニューから [ステップイン] を選択してステップを進めることができ、 [式] タブでは使用している変数の値が確認できます。

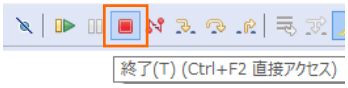


8) 希望のステップの左端をダブルクリックすることにより、ブレークポイントを設定することも可能です。

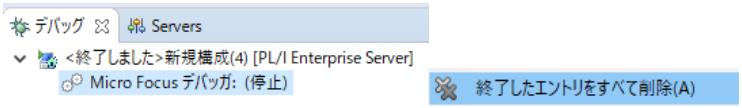


9) 先に進める場合は画面上部の再開アイコンをクリックします。

10) デバッグを終了させるため、画面上部の終了アイコンをクリックします。



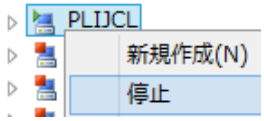
11) デバッガが停止状態になったのを確認後、右クリックして［終了したエントリをすべて削除］を選択し、これを削除します。



12) PL/I パースペクティブへ戻るには画面右上の PL/I アイコンをクリックします。

3.10 終了処理

1) サーバーエクスプローラ内で PLIJCL インスタンスを右クリックして [停止] を選択し、開始中のインスタンスを停止します。



2) PLIJCL インスタンスの停止状態を確認後に、Eclipse を終了します。

WHAT'S NEXT

- メインフレーム PL/I 開発：CICS Eclipse 編
- 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。