

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

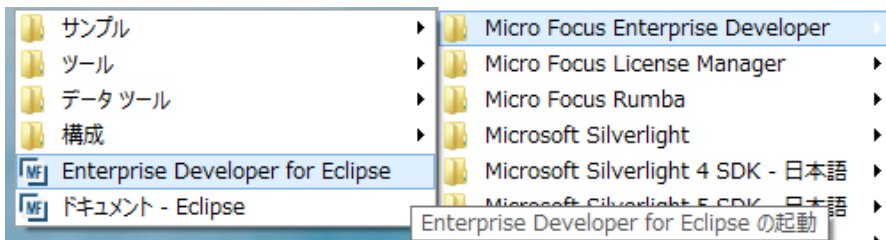
本チュートリアルで使用する例題プログラムは、キットに添付されている PLIJCLtutorial.zip に圧縮されています。これを C:¥ の直下に解凍しておきます。

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。

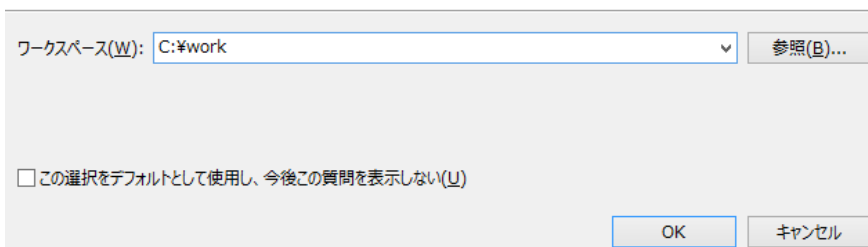
1) Windows スタートメニューから Enterprise Developer for Eclipse を選択して起動します。



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。

ワークスペースの選択

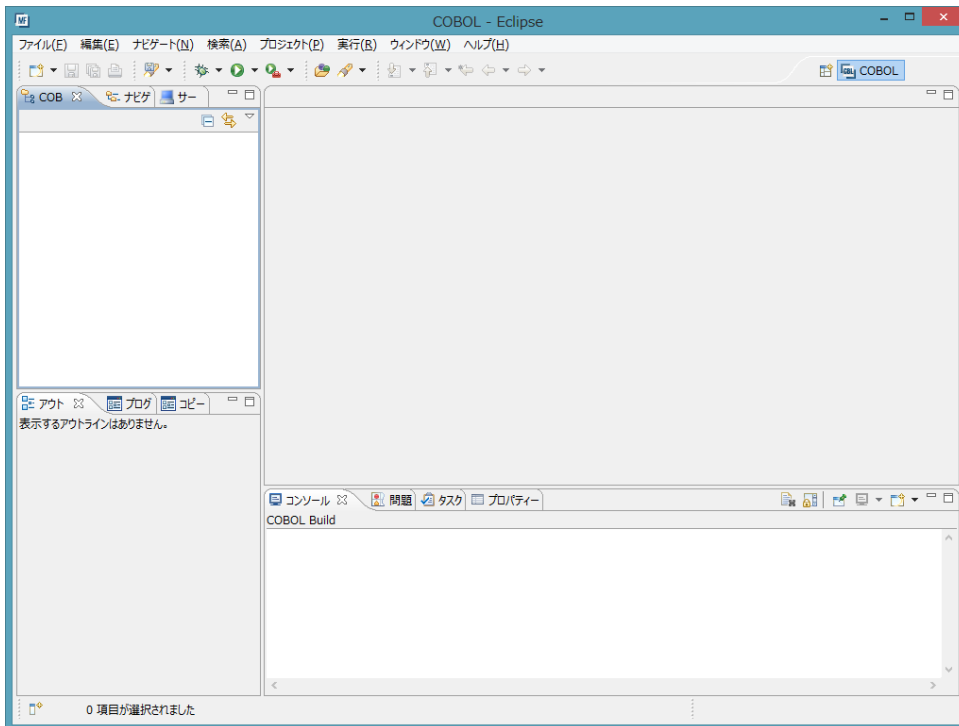
Eclipse は、ワークスペースと呼ばれるフォルダーにプロジェクトを保管します。
このセッションに使用するワークスペース・フォルダーを選択してください。



3) 以下のバナー画面が現れます。



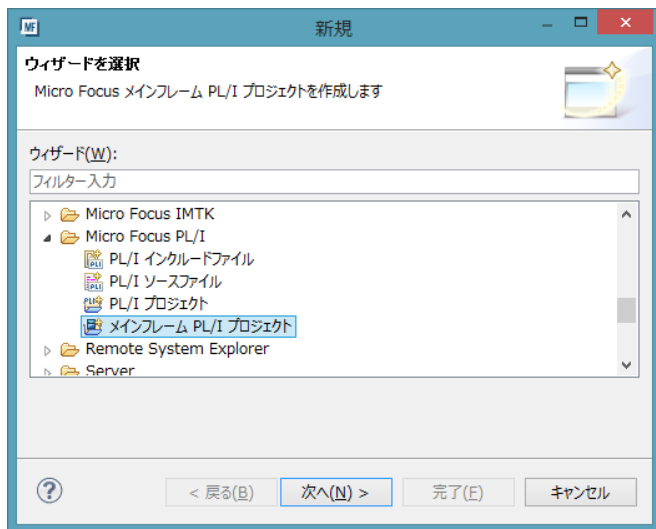
- 4) 左上の「ようこそ」の右の X をクリックしてバナーを閉じます。以下のように Eclipse の COBOL パースペクティブが開きます。



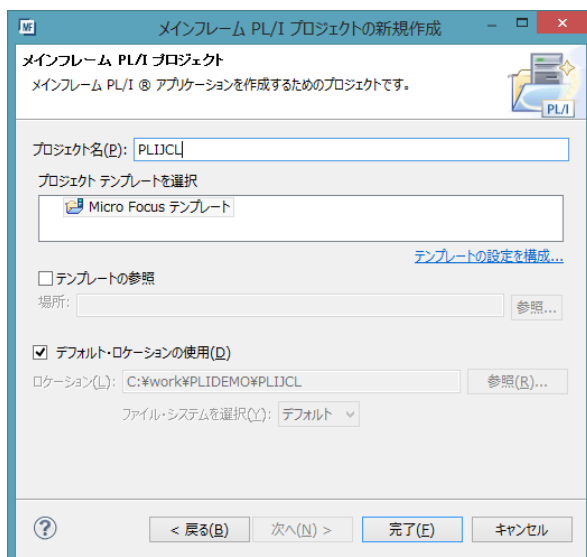
9.3 プロジェクトの新規作成

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。

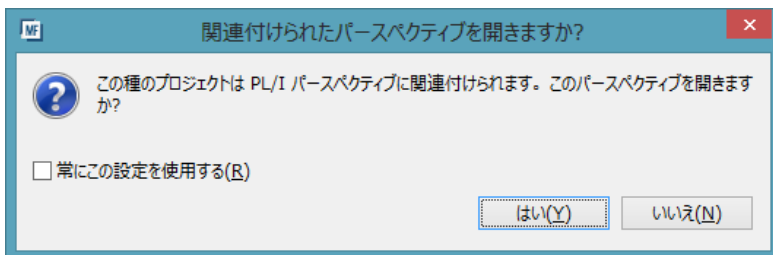
- 1) [ファイル] > [新規] > [その他...] を選択します。
- 2) 以下のダイアログで [Micro Focus PL/I] > [メインフレーム PL/I プロジェクト] を選択し [次へ] をクリックします。



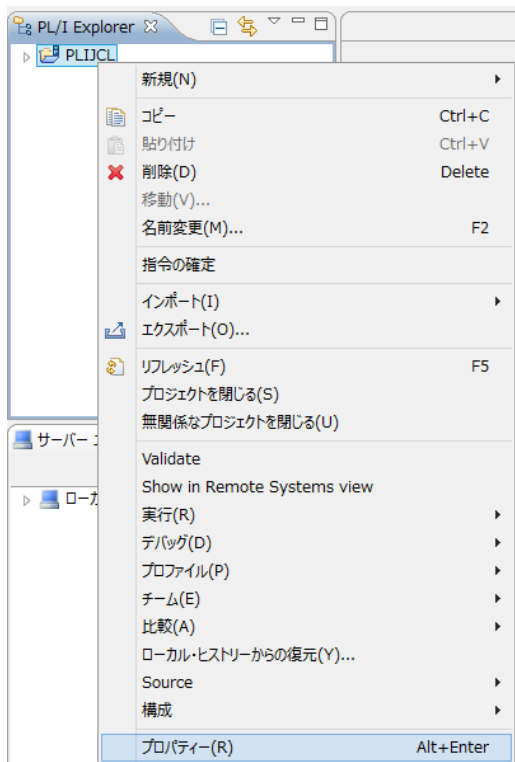
- 3) 以下のダイアログでプロジェクト名を指定します。ここでは“PLIJCL”と命名します。[完了] をクリックします。



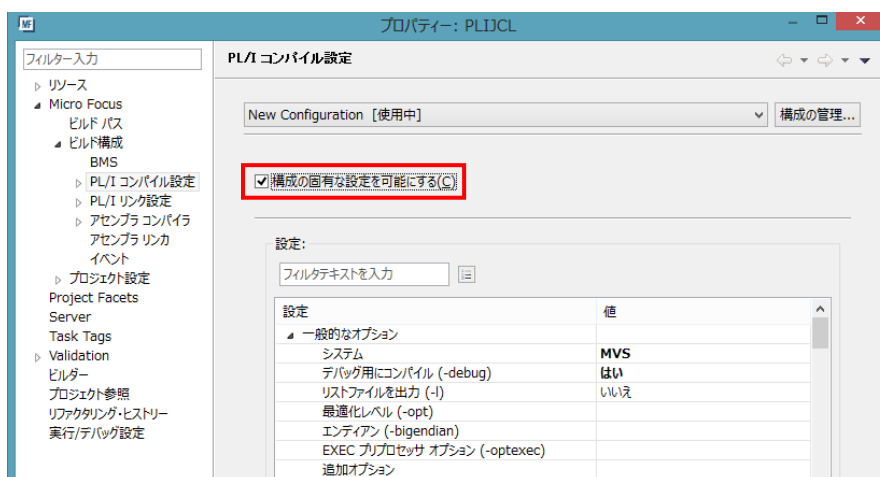
- 4) 以下のダイアログに対して [はい] をクリックします。



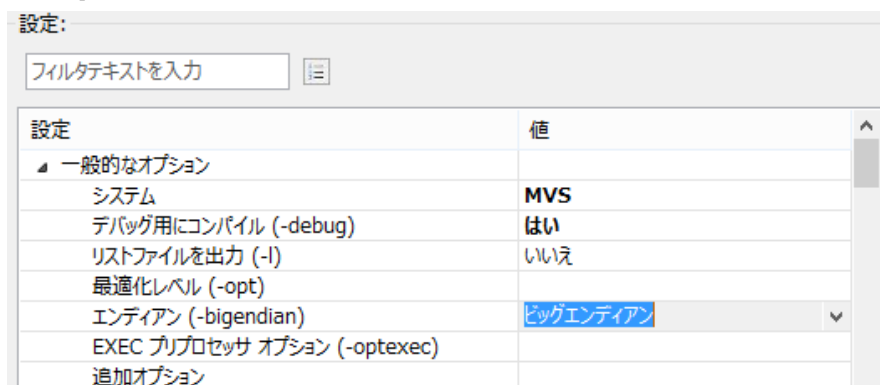
- 5) 空の PL/I プロジェクトが作成され PL/I エクスプローラ内に以下のようなツリー構造 PLIJCL のノードが作成されます。作成されたプロジェクトに必要なプロパティの設定を行います。PL/I エクスプローラ内で PLIJCL を右クリックして [プロパティ] を選択します。



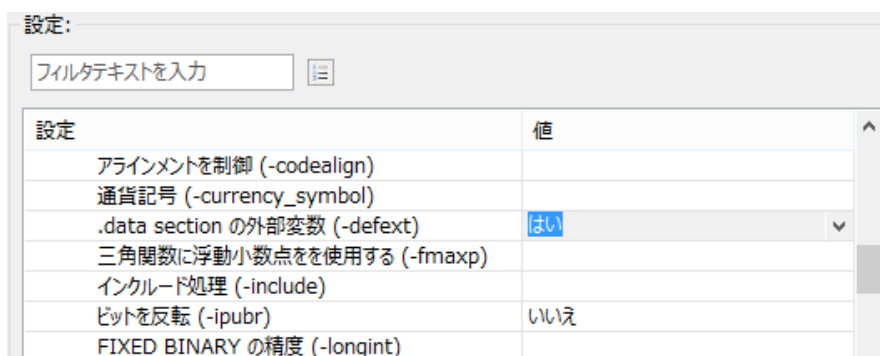
- 6) 以下のようにプロジェクトのプロパティダイアログが開きます。ここでプロジェクトの各種設定を行うことができます。左側ペインのツリービューにて [Micro Focus] > [ビルド構成] > [PL/I コンパイル設定] を開き、以下のように [構成の固有な設定を可能にする] をチェックオンしてください。



- 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。

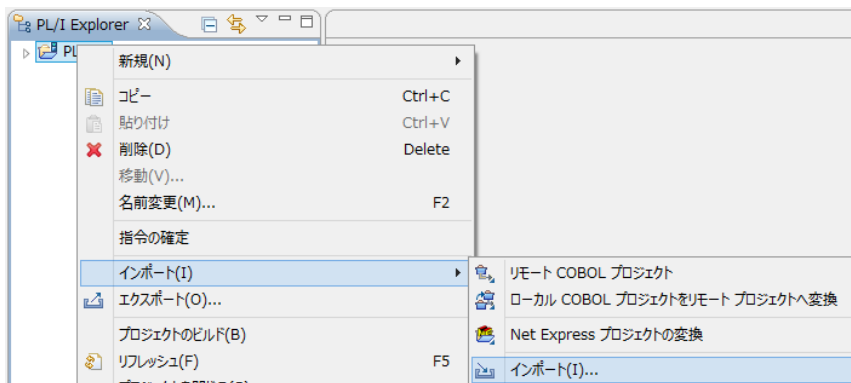


- 9) [OK] をクリックします。

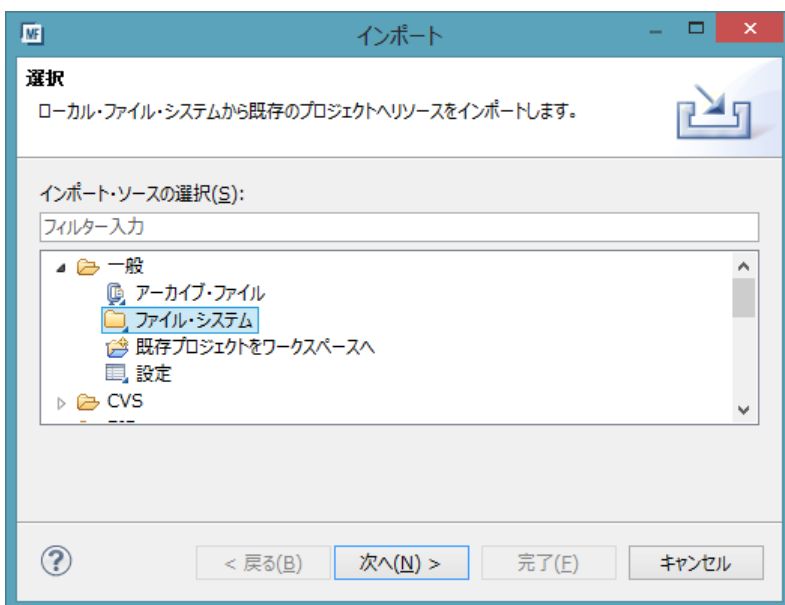
9.4 例題プログラムのインポート

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。

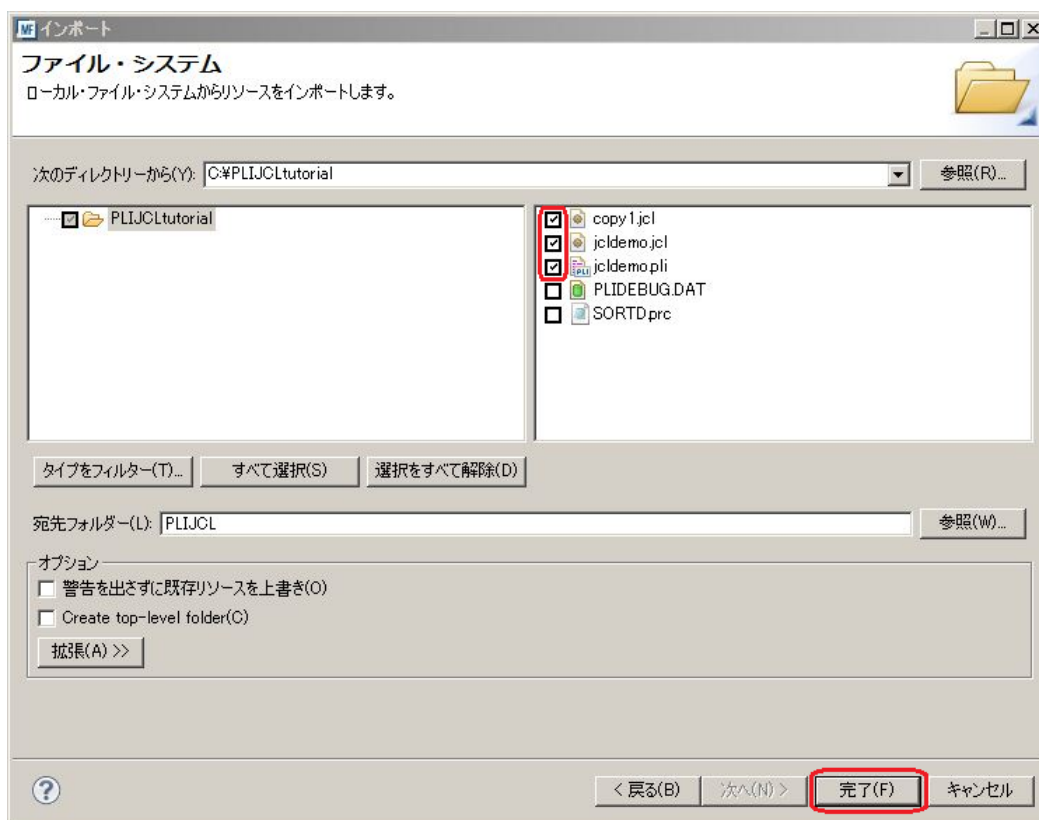
- 1) PL/I エクスプローラ内で PLIJCL を右クリックして [インポート] > [インポート] を選択します。



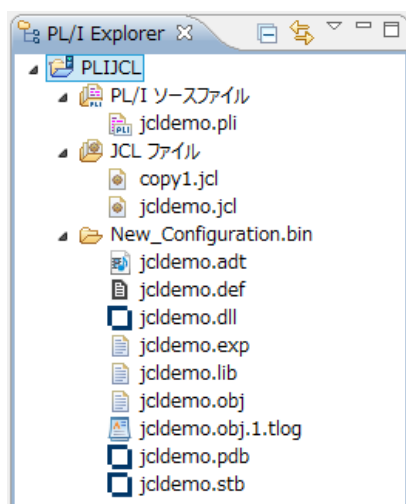
- 2) 以下のダイアログで [一般] > [ファイル・システム] を選択し、[次へ] をクリックします。



- 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。

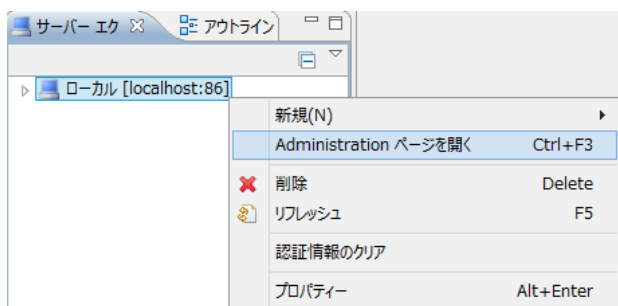


- 5) 右下のコンソールにエラーなくコンパイルが完了した旨が表示されればプロジェクトは完成となります。

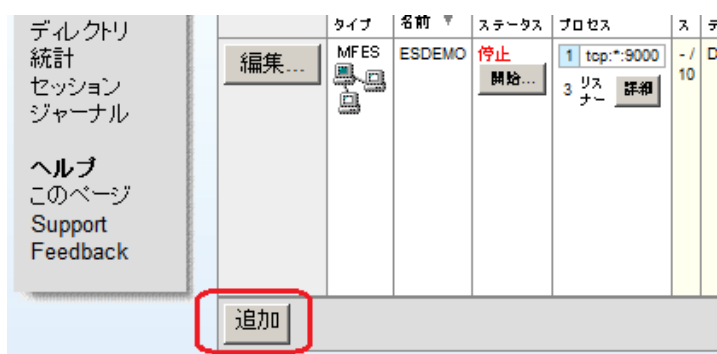
9.5 JES リージョンの作成

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。

Home

アクション
アドレス更新
エクスポート
インポート
すべて削除

構成
オプション
セキュリティ

表示
ディレクトリ
統計
セッション
ジャーナル

ヘルプ
このページ
Support
Feedback

ステータス
MDS0000I OK

Enterprise Server Administration > サーバー追加
WIN-NDAONJ6FAHV.microfocus.com (86)

サーバー追加 (Page 1 of 3):

サーバー名: PLIDEMO

動作モード:
 32-bit 64-bit

You cannot change your choice of working mode once a server is created, although importing a server.

キャンセル 次へ >>

- 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。

サーバー追加 (Page 2 of 3):

サーバー名: PLIDEMO

サーバータイプ:

MFES **Micro Focus Enterprise Server**
An enterprise server that provides an execution environment for COBOL application programs running as services in a service orientated architecture.

MFES (MSS) **Micro Focus Enterprise Server with Mainframe Subsystem Support**
An enterprise server that also provides an execution environment for CICS applications that have been migrated from the mainframe.

You can change your choice of server type later.

<< 戻る 次へ >>

- 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。

サーバー追加 (Page 3 of 3):

サーバー名:

System Directory:

開始オプション:

共有メモリページ数: <input type="text" value="512"/>	サービス実行プロセス: <input type="text" value="2"/>
共有メモリクッション: <input type="text" value="32"/>	トレーステーブルサイズ: <input type="text" value="341"/>
ローカルトレースサイズ: <input type="text" value="341"/>	診断ファイル最大サイズ: <input type="text" value="0"/>
要求ライセンス: <input type="text" value="10"/>	
コールドスタート診断ファイル: <input checked="" type="checkbox"/>	システムアベント時ダンプ: <input checked="" type="checkbox"/>
補助トレースアクティブ: <input type="checkbox"/>	ローカルコンソールを表示: <input checked="" type="checkbox"/>
Mainframe Subsystem Support: <input checked="" type="checkbox"/>	64-Bit Working Mode: <input type="checkbox"/>

トレースフラグ:

タスク管理 <input type="checkbox"/>	ストレージ管理 <input type="checkbox"/>	テーブル管理 <input type="checkbox"/>
アプリケーションコンテナ <input type="checkbox"/>	要求ハンドラ <input type="checkbox"/>	RMインタフェース <input type="checkbox"/>
通信 <input type="checkbox"/>	アプリケーション <input type="checkbox"/>	終了 <input type="checkbox"/>

生成オプション:

TN3270リスナーの作成 using port

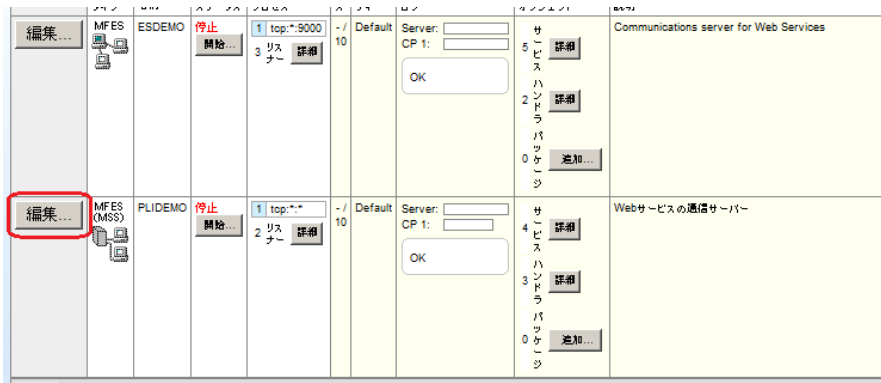
- 6) [追加] をクリックします。

構成情報

説明

Webサービスの通信サーバー

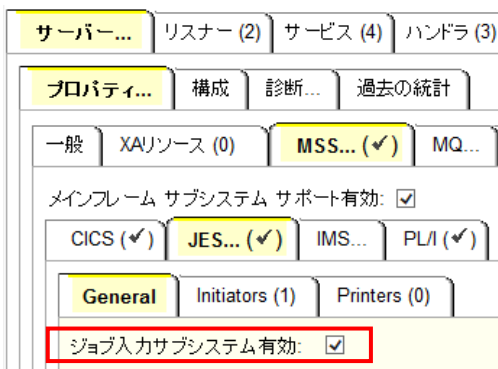
7) 以下のように PLIDEMO が新規に追加されました。左端の [編集...] ボタンをクリックします。



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。

General	Initiators (0)	Printers (0)
ジョブ入力サブシステム有効: <input checked="" type="checkbox"/>		
JESプログラムパス: C:\work\PLIJCL\New_Configuration.bin		
システムカタログ: C:\work\PLIJCL\DATASETS\CATALOG.DAT		
データセットの省略時ロケーション: C:\work\PLIJCL\DATASETS		
システムプロシージャライブラリ: SYS1.PROCLIB		
Fileshare Configuration Location: <input type="text"/>		
<input type="button" value="Apply"/>		



重要事項

パスは全て半角英数字で指定してください。
以降も同様です。

[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。C:\work\PLIDEMO\DATASETS 配下に SYS1.PROCLIB フォルダを作成しておきます。

- 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。

一般 Initiators (0) Printers (0)

▲ Add Initiator...

名前:
INITABC

Class:
ABC

説明:
クラス ABCのイニシエータ

キャンセル 追加

- 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。

CICS (✓) JES... (✓) IMS...

一般 Initiators (1) Printers (0)

名前	クラス	説明
INITABC	ABC	クラス ABCのイニシエータ

編集... 追加

- 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。

CICS (✓) JES... (✓) IMS... PL/I

一般

PL/I 有効:

Prompt for PLITEST attachment:

Codewatch ソースパス:
C:\work\PLIJCL

Codewatch STB パス:
C:\work\PLIJCL\New_Configuration.bin

PL/I 構成ディレクトリ:
C:\work\PLIJCL

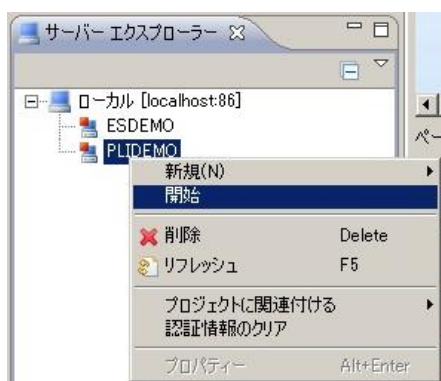
Apply

- 16) [Apply] をクリックし、JES リージョンが作成されました。

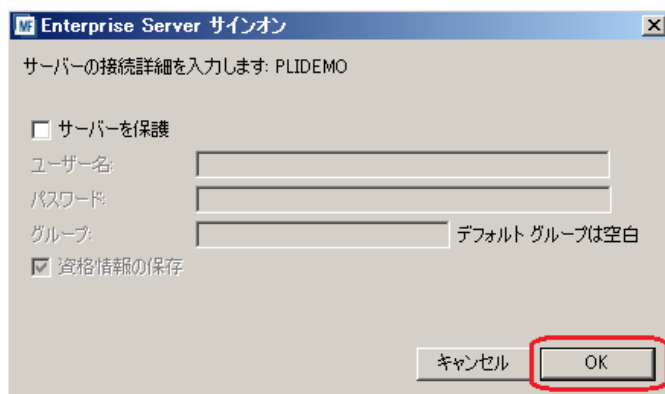
9.6 JES リージョンの起動

作成された JES リージョンを起動します。

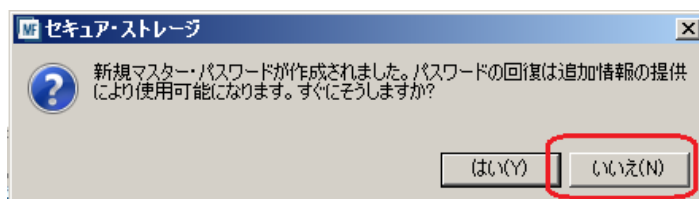
- 1) サーバーエクスプローラ内に新規作成された PLIDEMO が表示されていることを確認します。もし表示されていなければ [ローカル] を右クリックして [リフレッシュ] を選択してください。
- 2) PLIDEMO を右クリックし [プロジェクトに関連付ける] > [PLIJCL] を選択します。
- 3) PLIDEMO を右クリックし [開始] を選択します。



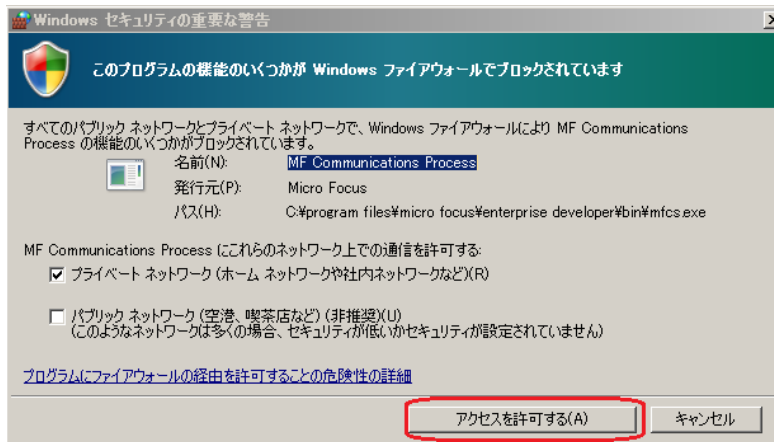
- 4) このチュートリアルではセキュリティプロテクトはかけないで進めます。以下のダイアログが現れたらそのまま [OK] をクリックします。



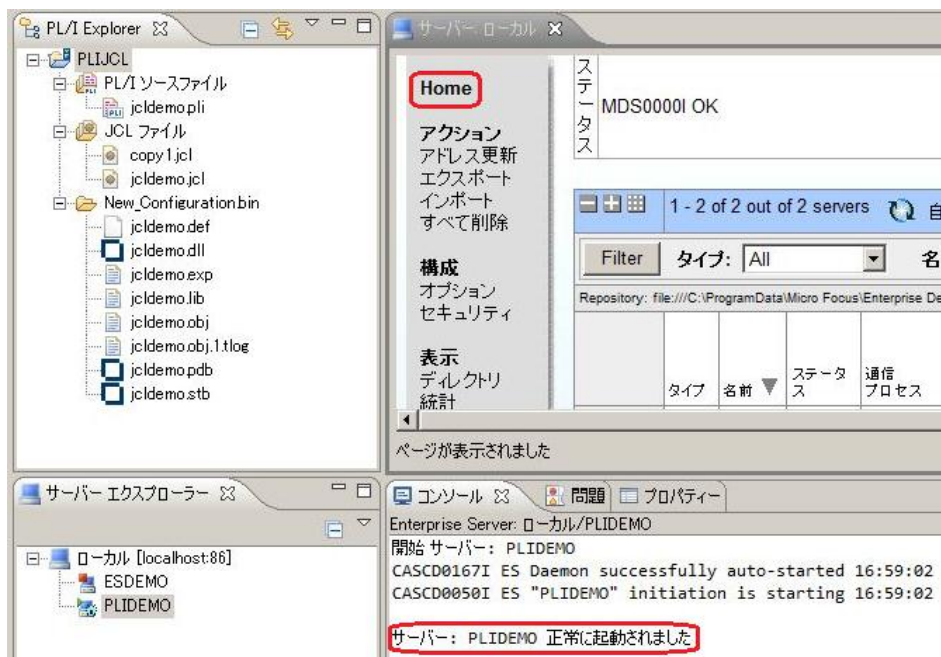
- 5) 以下のダイアログでは [いいえ] をクリックします。



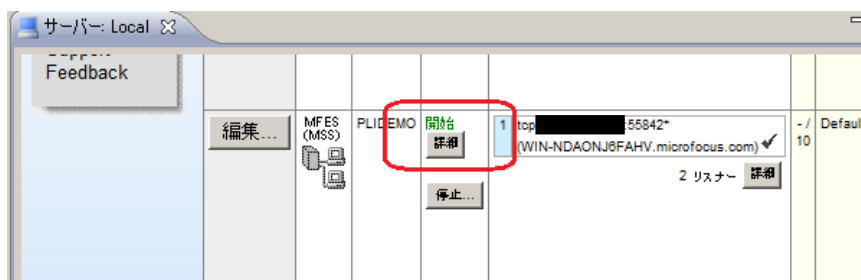
6) Firewall の警告が出る場合には [アクセスを許可する] をクリックします。



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。

The image shows two screenshots. The top screenshot is the PLIDEMO ES Console interface. It has a navigation bar with 'サーバー...' selected. Below it are tabs for 'プロパティ...', '構成', '診断...' (highlighted), and '過去の統計'. Under '診断...', there are sub-tabs for 'トレース', 'ダンプ', 'ESコンソール' (highlighted), and 'CSコンソール'. A '画面更新' button is on the left. On the right, there are radio buttons for 'Show entries from 1 to 10 of 44 total entries' and 'Show last 10 lines'. A 'Show Entire Log' button is highlighted with a red box. The main area shows a list of log entries with columns for 'Entry' and 'Event'.

Entry	Event
35	130719 15341008 3304 PLIDEMO CASSI1600I SEP initialization completed successfully 15:34:09
36	130719 15341028 3304 PLIDEMO JES000042I SSTM not enabled: CICS 15:34:10
37	130719 15341048 3304 PLIDEMO CASSI5001I PLTPI - No PLT Specified 15:34:10
38	130719 15341088 3304 PLIDEMO CASSI5040I Active SEP memory strategy set to x'00000001', retain count 100 15:34:10
39	130719 15341107 3508 PLIDEMO CASSI1600I SEP initialization completed successfully 15:34:11
40	130719 15341127 2598 PLIDEMO CASBJ0005I Batch initiator started for job classes 'INITABC' 15:34:11
41	130719 15341127 3508 PLIDEMO CASSI5021I PLTPI Phase 2 - No PLT Specified 15:34:11
42	130719 15341147 2598 PLIDEMO JCLCM0188I JOB01000 COPY1 JOB STARTED 15:34:11
43	130719 15341167 4080 PLIDEMO CASSI1600I SEP initialization completed successfully 15:34:11
44	130719 15341167 2598 PLIDEMO JCLCM0182I JOB01000 COPY1 JOB ENDED - COND CODE 0000 15:34:11

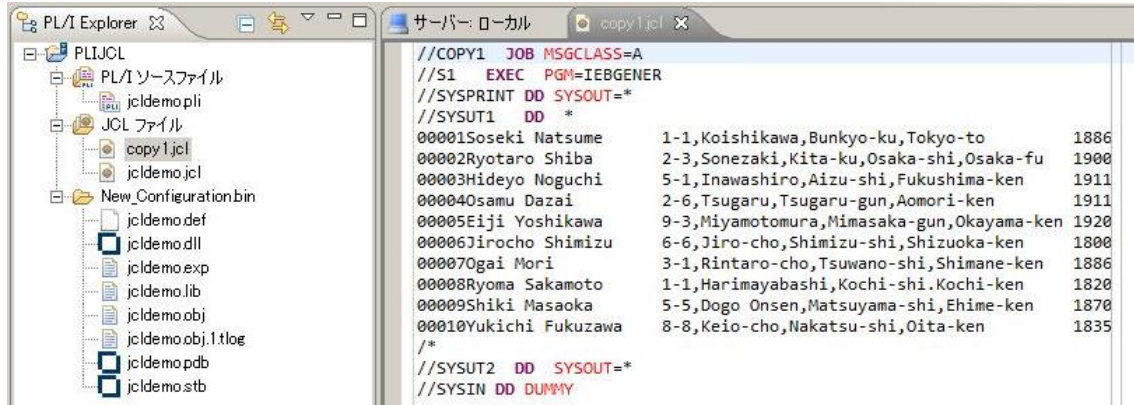
The bottom screenshot is a Windows Internet Explorer browser window showing the log output from the ES console. The address bar shows 'http://MF_GETCASLOG'. The page content is a list of log entries with columns for time, process ID, and event details.

Time	Process ID	Event
120523 17521899	CAS000099I	ES Build Tag: ED2-0/20120412a 17:52:18
120523 17521939	CAS00120I	Server manager created for ES MSSDEMO, process-id = 2444 17:52:19
120523 17521960	CASS10000I	Server manager initialization started 17:52:19
120523 17522166	CASS14005I	Retrieving ES configuration from WFDS (127.0.0.1:86) 17:52:21
120523 17522369	2444 MSSDEMO	CASS10012I Region running in 32 bit mode 17:52:22
120523 17522369	2444 MSSDEMO	CASSE0011I Region is running with Resource security disabled. 17:52:23
120523 17522375	2444 MSSDEMO	CASS10001I Region will use SYSID \$1VP 17:52:23
120523 17522397	CAS01075I	ES TSC Service Process created for Server MSSDEMO, process-id = 3872 17:52:23
120523 17522417	3872 MSSDEMO	CAS00007I ES TSC Service Process initialization started 17:52:24
120523 17522419	CAS00127I	SEP 00001 created for ES MSSDEMO, process-id = 3582 17:52:24
120523 17522419	3872 MSSDEMO	CAS05015I ES TSC Log and Intra-Partition TS/TD: "C:\Users\Administrator\Documents\Micro Focus\Enterprise D
120523 17522439	3592 MSSDEMO	CASS11500I SEP initialization started 17:52:24
120523 17522441	CAS00127I	SEP 00002 created for ES MSSDEMO, process-id = 3012 17:52:24
120523 17522441	3872 MSSDEMO	CAS05016I ES TSC Log and Intra-Partition TS/TD File size: 32 blocks (2,016,724 bytes) 17:52:24
120523 17522461	3012 MSSDEMO	CASS11500I SEP initialization started 17:52:24
120523 17522463	CAS01039I	ES Communications Server created, ES MSSDEMO, process-id = 1716 17:52:24
120523 17522463	3872 MSSDEMO	CAS05017I ES TSC Log and Intra-Partition TS/TD running with 1 buffers of 63,020 bytes each 17:52:24
120523 17522483	1716 MSSDEMO	CAS05001I Communications interface 01 initialization started 17:52:24
120523 17522485	2444 MSSDEMO	CASK01000I ES concurrent request limit: 0000000010 17:52:24
120523 17522485	3872 MSSDEMO	CAS00002I ES TSC Service Process initialization complete 17:52:24
120523 17522485	2444 MSSDEMO	CASS11000I Server Manager initialization completed successfully 17:52:24
120523 17522485	CAS01071I	Administration SEP created for Server MSSDEMO, process-id = 1060 17:52:24
120523 17522506	1060 MSSDEMO	CASS11500I SEP initialization started 17:52:24
120523 17522508	CAS05003I	Communications interface 01 initialization complete 17:52:24
120523 17522713	1716 MSSDEMO	CAS05100I Communications Process instance 01 is ready to accept requests 17:52:26
120523 17522713	3592 MSSDEMO	CASS11425I TS non-recoverable cold started 17:52:26
120523 17522713	3592 MSSDEMO	CASS11425I TS recoverable warm started 17:52:27
120523 17522713	3592 MSSDEMO	CASS11426I TD non-recoverable cold started 17:52:27
120523 17522713	3592 MSSDEMO	CASS11426I TD recoverable warm started 17:52:27
120523 17522713	3592 MSSDEMO	CASS11600I SEP initialization completed successfully 17:52:27
120523 17522714	3592 MSSDEMO	CASS15001I PLTPI - No PLT Specified 17:52:27
120523 17522714	3592 MSSDEMO	CASS15040I Active SEP memory strategy set to x'00000001', retain count 100 17:52:27
120523 17522725	3012 MSSDEMO	CASS11600I SEP initialization completed successfully 17:52:27
120523 17522777	3012 MSSDEMO	CASS15021I PLTPI Phase 2 - No PLT Specified 17:52:27
120523 17522817	1060 MSSDEMO	CASS11600I SEP initialization completed successfully 17:52:28

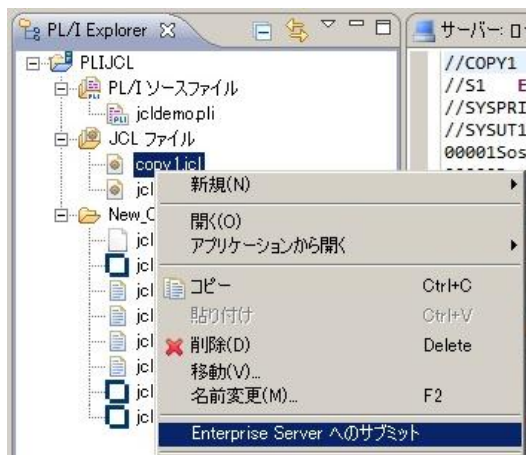
9.7 簡単な JCL の実行

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。

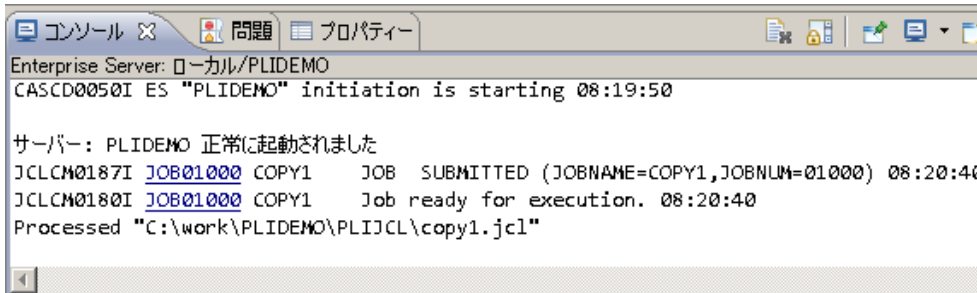
- 1) プロジェクト内の PL/I エクスプローラで copy1.jcl をダブルクリックし、エディタでその内容を確認します。



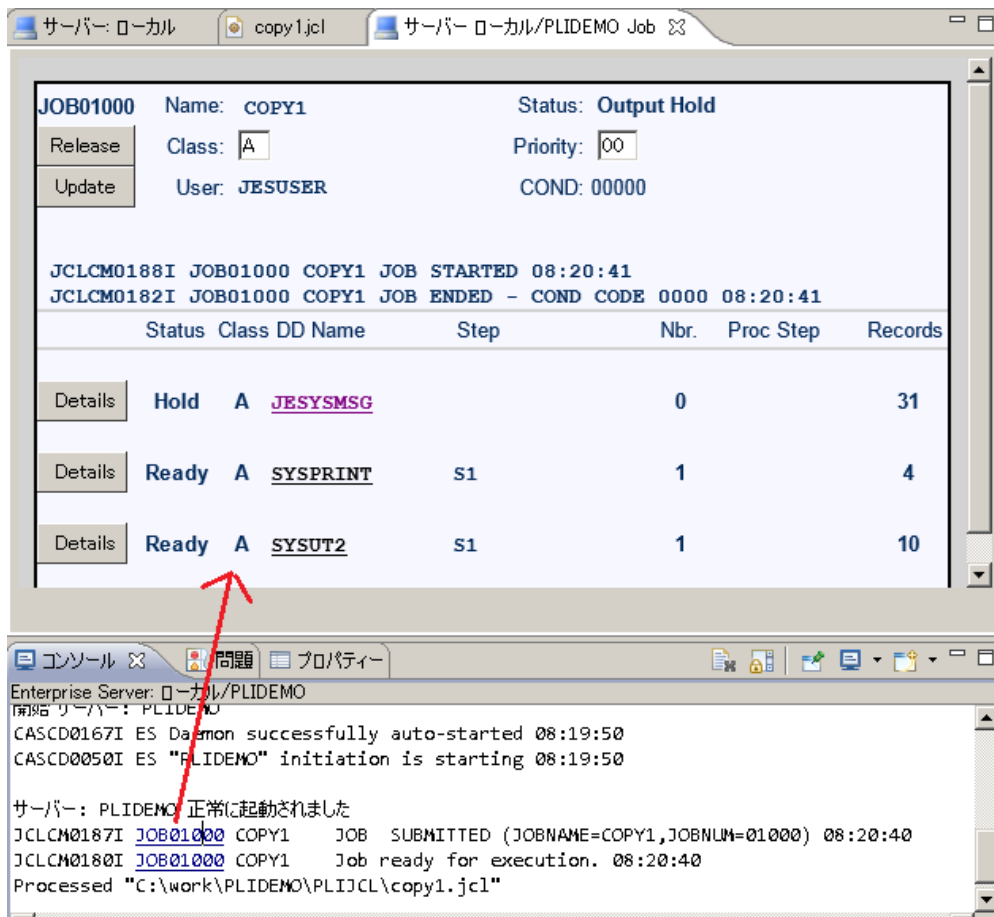
- 2) このジョブは IEBGENER ユーティリティを起動して JCL 内に書かれたインラインデータを SYSOUT に書き出しているだけのものです。
- 3) PL/I エクスプローラ内で copy1.jcl を右クリックし「Enterprise Server へのサブミット」を選択します。



- 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 ジョブの出力カスプールが表示されます。





以上で簡単なジョブの実行が確認できました。

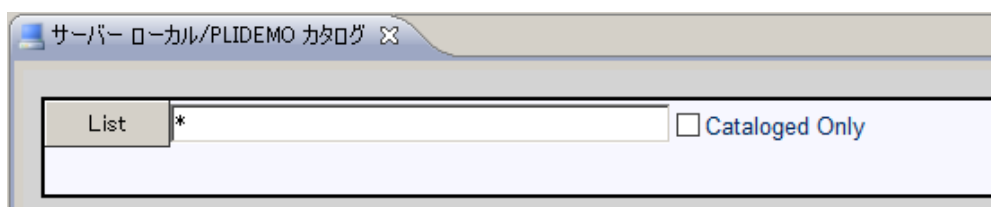
9.8 プロシージャライブラリの作成

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。

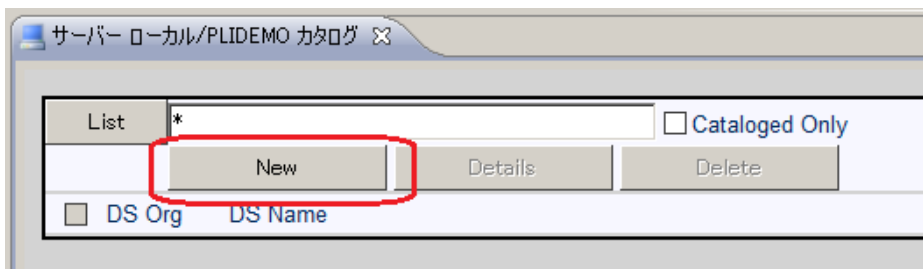
- 1) サーバーエクスプローラ内で PLIDEMO を右クリックし [カタログ表示] を選択します。



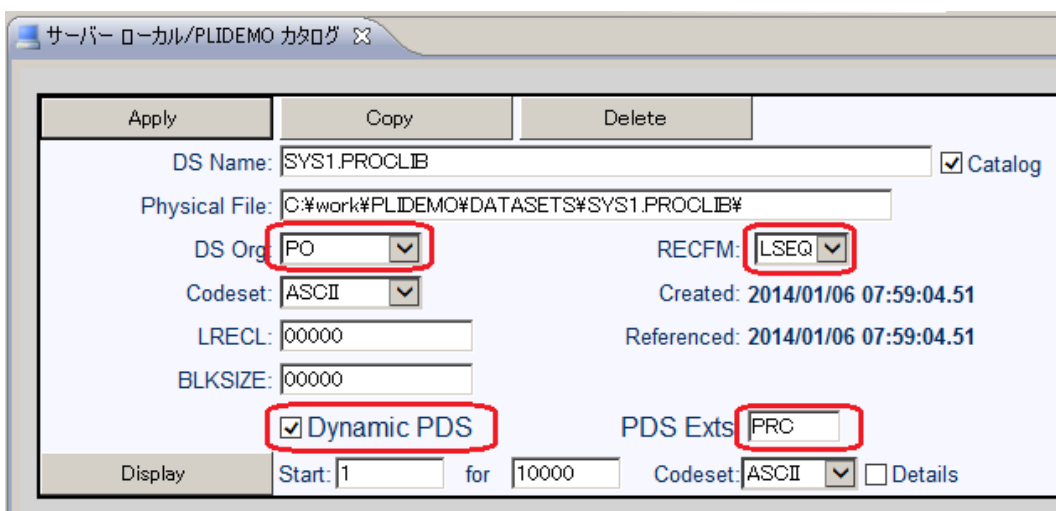
- 2) 以下のようにカタログビューが表示されます。ここで [List]ボタンをクリックします。



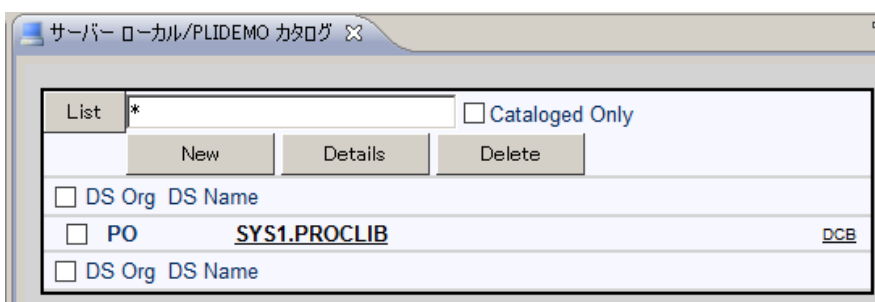
- 3) 現在カタログされているデータセットは何もありません。そこで [New] ボタンをクリックします。



- 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。



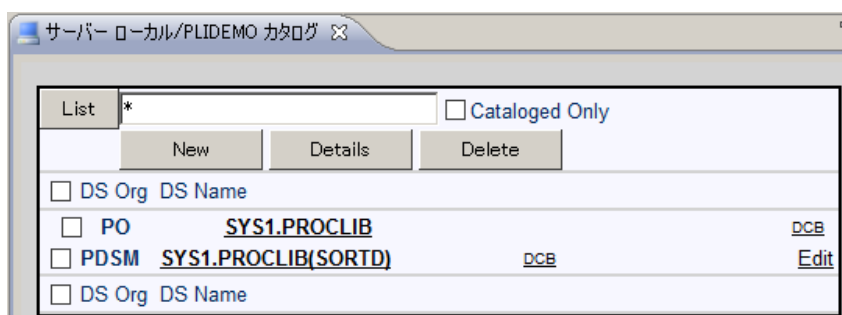
- 5) カタログビューを再表示すると、以下のようにカタログエントリ SYS1.PROCLIB が作成されています。



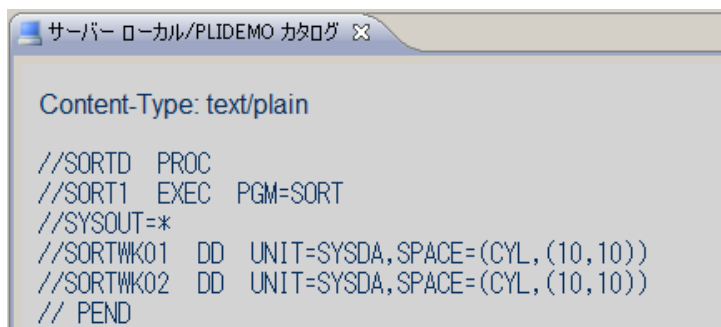
- 6) 前項で作成した C:\work\PLIDEMO\DATASETS\SYS1.PROCLIB に C:\PLIJCLtutorial\SORTD.prc をコピーします。



- 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。



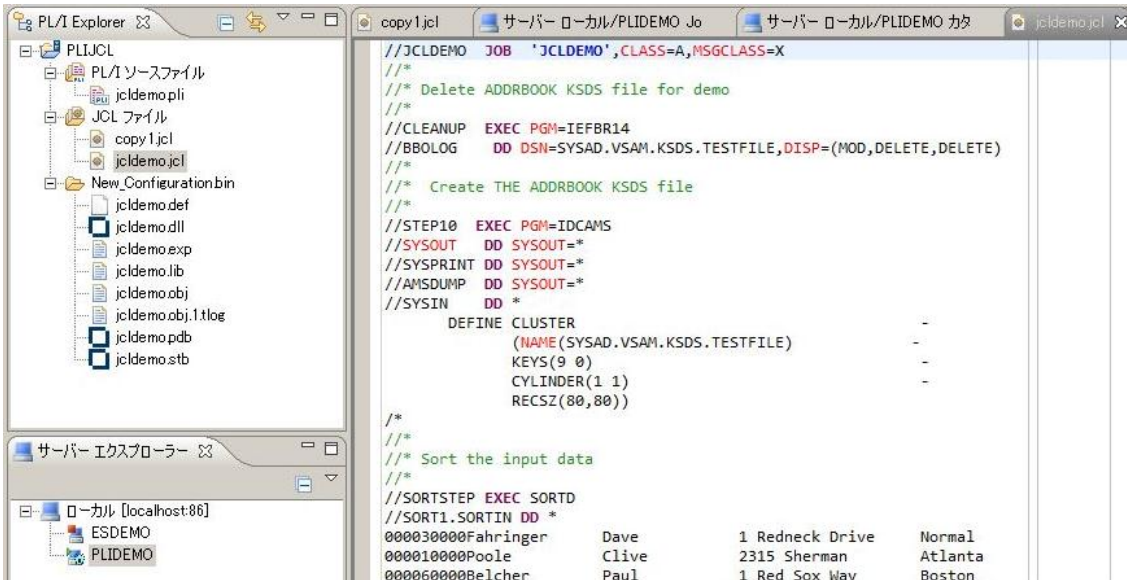
- 8) SYS1.PROCLIB(SORTD) をクリックすると以下のようにその内容を表示させることができます。



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。



```
//JCLDEMO JOB 'JCLDEMO',CLASS=A,MSGCLASS=X
/*
/* Delete ADDRBOOK KSDS file for demo
/*
//CLEANUP EXEC PGM=IEFBR14
//BBOLOG DD DSN=SYSAD.VSAM.KSDS.TESTFILE,DISP=(MOD,DELETE,DELETE)
/*
/* Create THE ADDRBOOK KSDS file
/*
//STEP10 EXEC PGM=IDCAMS
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//SYSIN DD *
        DEFINE CLUSTER
                (NAME(SYSAD.VSAM.KSDS.TESTFILE)
                KEYS(9 0)
                CYLINDER(1 1)
                RECSZ(80,80))
/*
/*
/* Sort the input data
/*
//SORTSTEP EXEC SORTD
//SORT1.SORTIN DD *
000030000Fahringer      Dave      1 Redneck Drive      Normal
000010000Poole         Clive    2315 Sherman         Atlanta
000060000Belcher       Paul     1 Red Sox Way        Boston
```

このジョブは4つのステップから構成され下記のよう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。

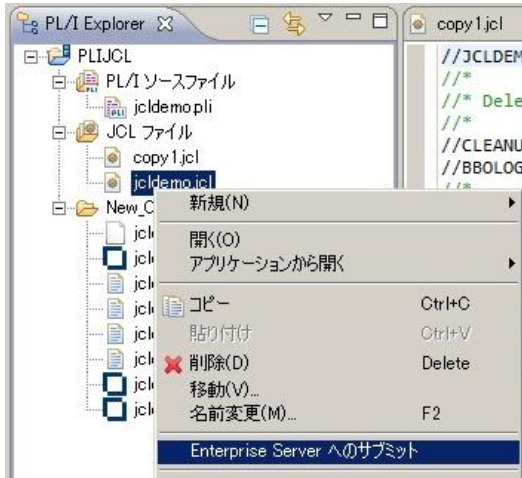
- STEP3: SORTSTEP

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。

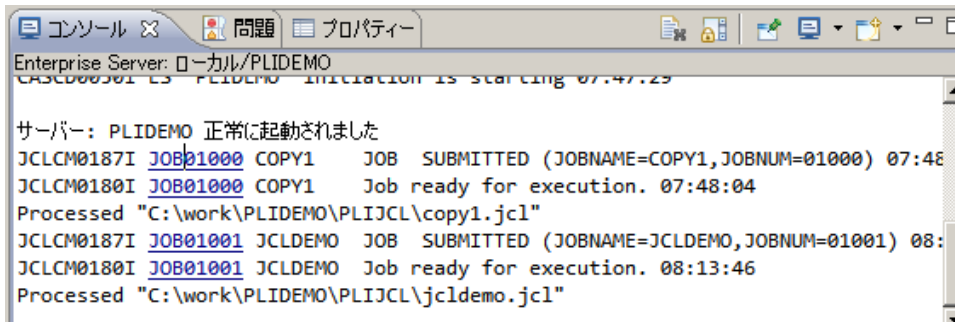
- STEP4: STEP20

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 jcdemo.jcl を右クリックして [Enterprise Server へのサブミット] を選択します。



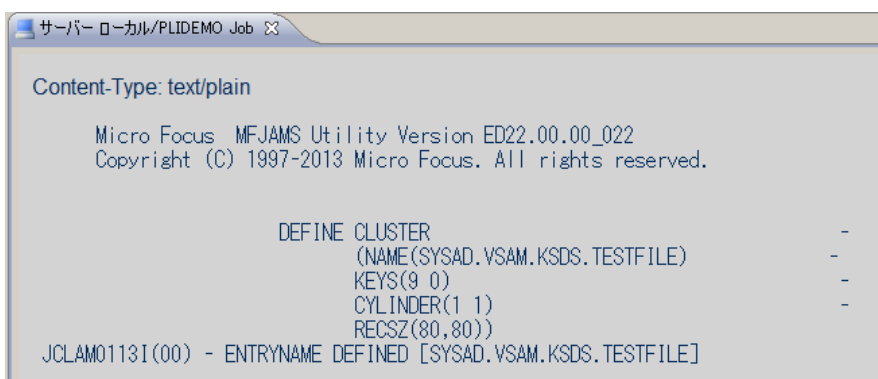
- 3) 以下のように JCLDEMO ジョブが実行された旨のメッセージがコンソールに表示されます。



- 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。

JOB01001	Name: JCLDEMO	Status: Output Hold					
Release	Class: A	Priority: 00					
Update	User: JESUSER	COND: 00000					
<p>JCLCM0188I JOB01001 JCLDEMO JOB STARTED 13:44:24 JCLCM0182I JOB01001 JCLDEMO JOB ENDED - COND CODE 0000 13:44:24</p>							
	Status	Class	DD Name	Step	Nbr.	Proc Step	Records
Details	Hold	X	JESYSMSG		0		87
Details	Ready	X	SYSPRINT	STEP10	2		11
Details	Ready	X	SYSOUT	SORTSTEP	3	SORT1	11
Details	Hold	X	SYSPRINT	STEP20	4		2

- 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。

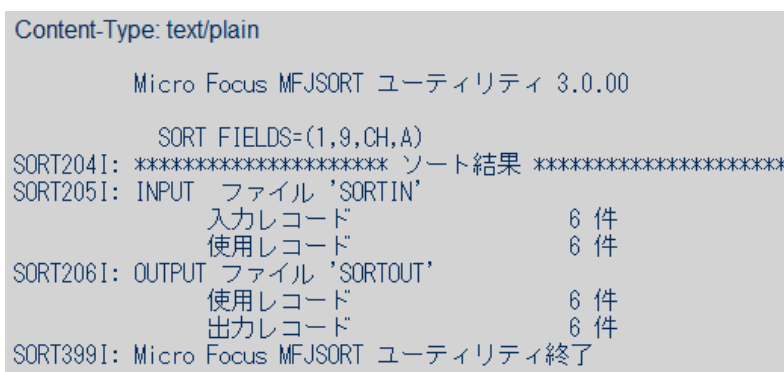


The screenshot shows a browser window titled 'サーバー ローカル/PLIDEMO Job' with a 'Content-Type: text/plain' header. The text content is as follows:

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

          DEFINE CLUSTER                -
            (NAME(SYSAD.VSAM.KSDS.TESTFILE) -
            KEYS(9 0)                    -
            CYLINDER(1 1)                -
            RECSZ(80,80))                -
JCLAM0113I(00) - ENTRYNAME DEFINED [SYSAD.VSAM.KSDS.TESTFILE]
```

- 8) 4) の画面に戻り SORTSTEP の SYSOUT をクリックすると、以下のように SORT ユーティリティの結果を確認することができます。

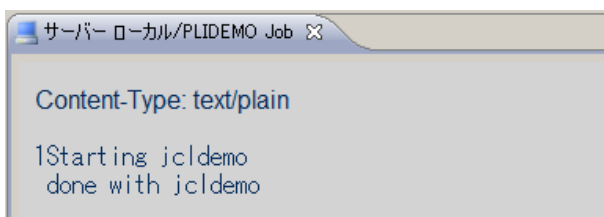


The screenshot shows a browser window titled 'サーバー ローカル/PLIDEMO Job' with a 'Content-Type: text/plain' header. The text content is as follows:

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

          SORT FIELDS=(1,9,CH,A)
SORT204I: ***** ソート結果 *****
SORT205I: INPUT   ファイル 'SORTIN'
              入力レコード           6 件
              使用レコード           6 件
SORT206I: OUTPUT  ファイル 'SORTOUT'
              使用レコード           6 件
              出力レコード           6 件
SORT399I: Micro Focus MFJSORT ユーティリティ終了
```

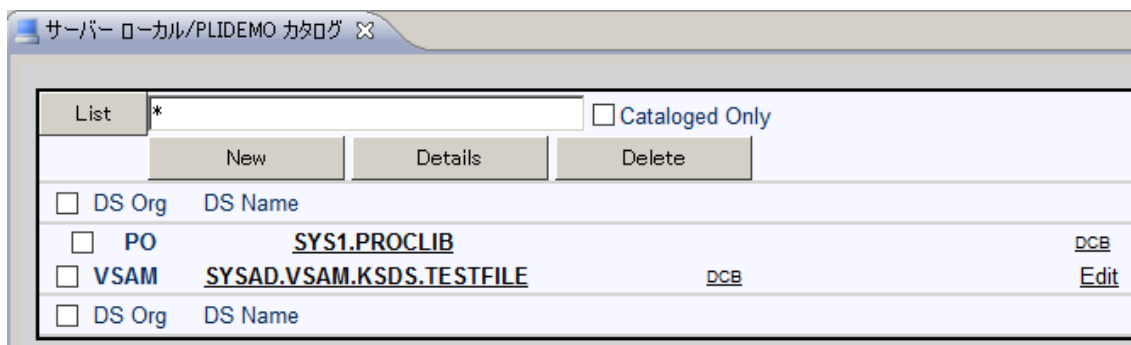
- 9) 4) の画面に戻り STEP20 の SYSPRINT をクリックすると、以下のように PL/I プログラムの PUT 文で書かれた内容を確認することができます。



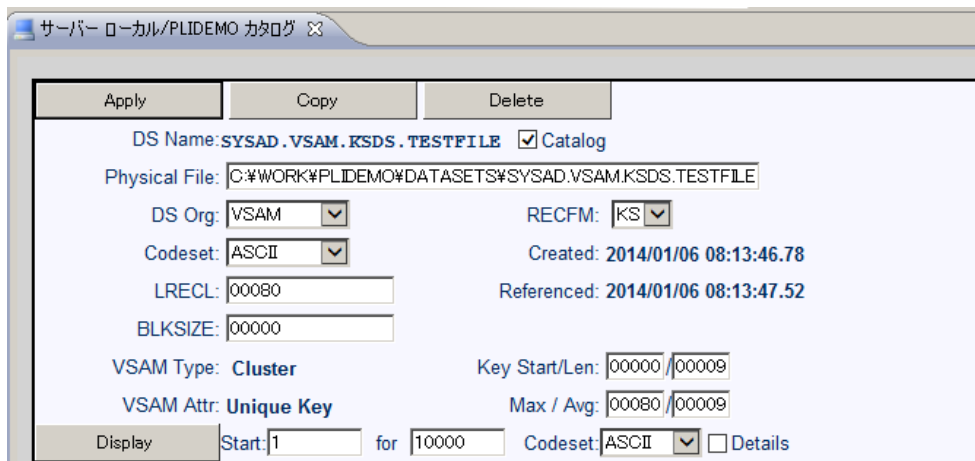
The screenshot shows a browser window titled 'サーバー ローカル/PLIDEMO Job' with a 'Content-Type: text/plain' header. The text content is as follows:

```
1Starting jcldemo
done with jcldemo
```

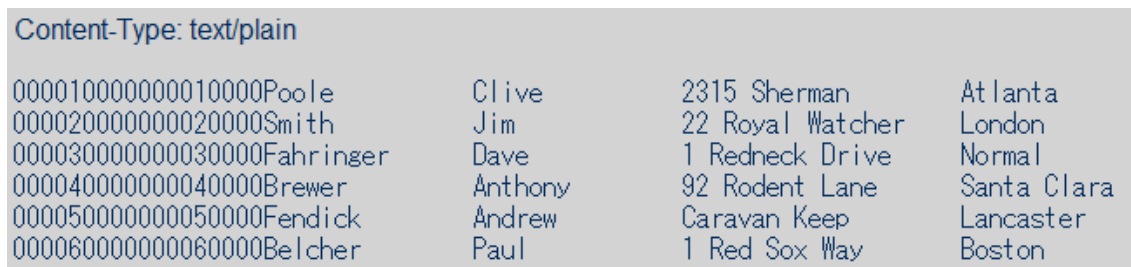
- 10) 続いてこのジョブの実行によってカタログされたデータセットを見てみます。サーバーエクスプローラ内で PLIDEMO を右クリックし [カタログ表示] を選択します。[List] ボタンをクリックすると VSAM ファイル SYSAD.VSAM.KSDS.TESTFILE がカタログされていることがわかります。



- 11) 名前の右側の [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。



- 12) [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。



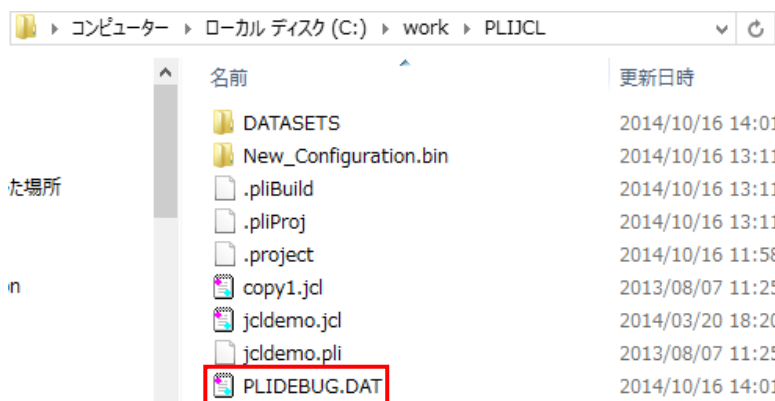
9.10 PL/I バッチプログラムのデバッグ

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。

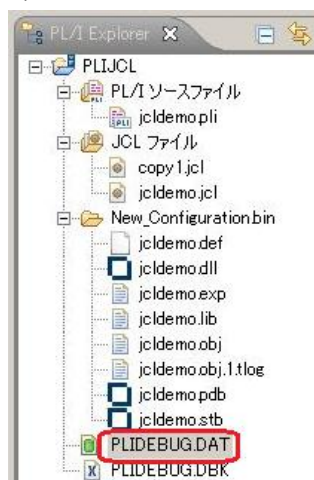
JES リージョン配下で実行される PL/I プログラムのデバッグは、PLIDEBUG.DAT という構成ファイルで指示します。インストール直後は存在しませんが PL/I プログラムを最初に実行した際に作成されます。この例題では JCLDEMO.jcl が実行された後に空のファイルが作成されています。

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。このファイルは主キーが先頭 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。

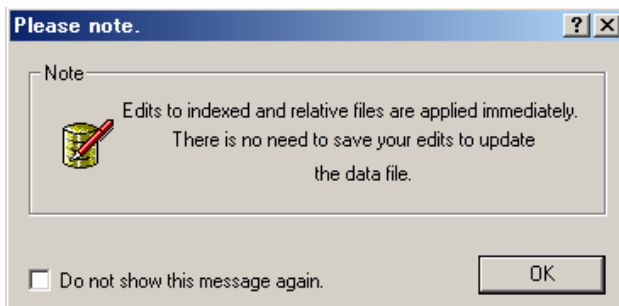
本チュートリアルでは構成済みの PLIDEBUG.DAT ファイルを使用するため、C:¥PLIJCLtutorial からプロジェクトへ、ソース同様の手順でインポートし書きしてください。



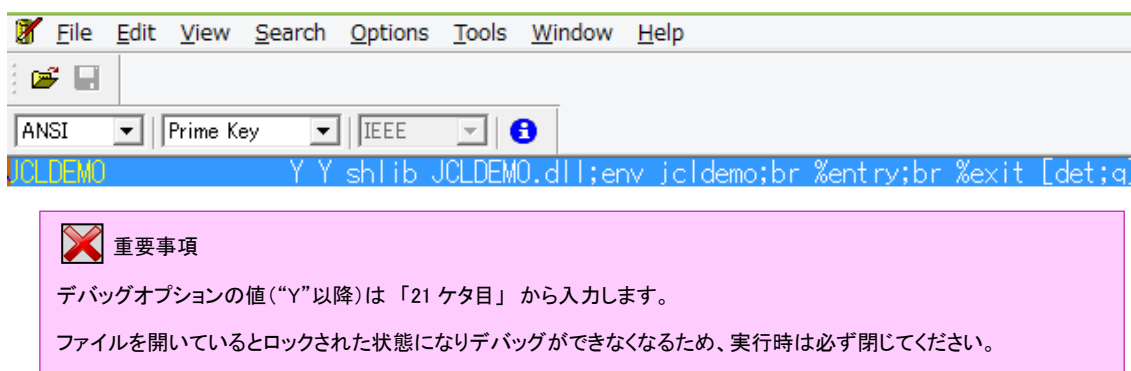
1) PL/I エクスプローラ内で PLIDEBUG.DAT をダブルクリックします。



2) 以下のダイアログに対して [OK] をクリックします。



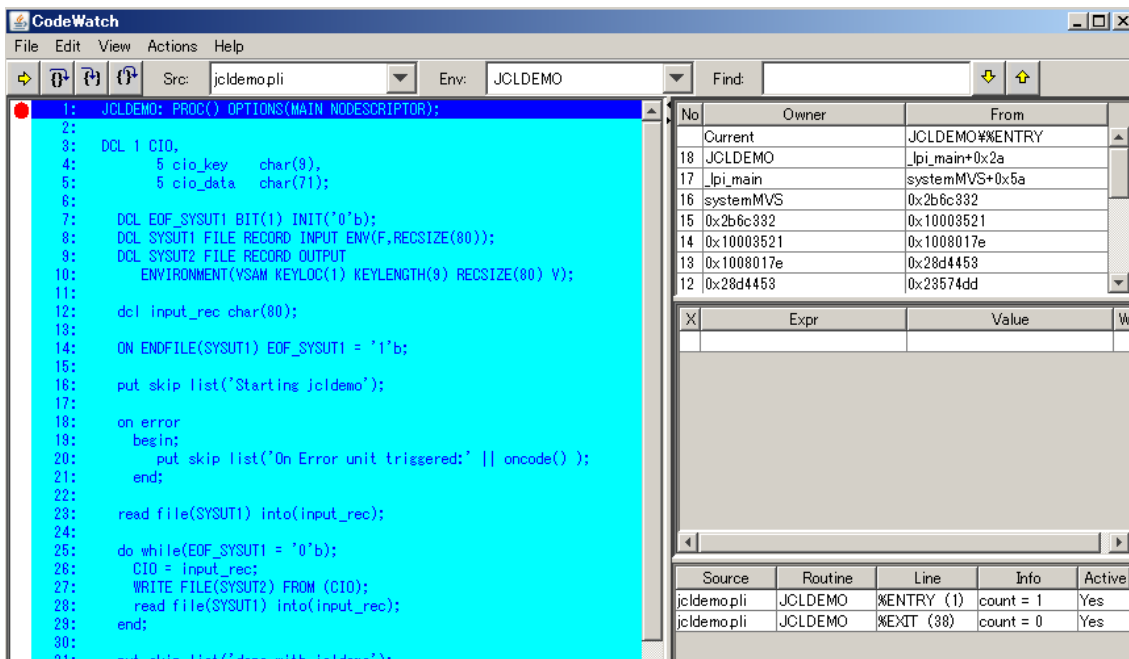
3) 以下のように 1 件のレコードが表示されます。主キーの値はデバッグ対象プログラム名を表します。"JCLDEMO" となっていますので、JCLDEMO.pli が実行されると自動的にデバッグセッションが開始するように設定されます。レコードの残りの部分はデバッグのオプションを示しています。



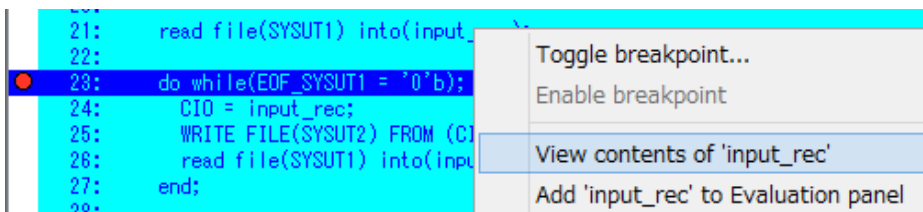
4) [File] > [Exit] でデータファイルエディタを終了します。

5) 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。

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



7) 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。



8) 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。



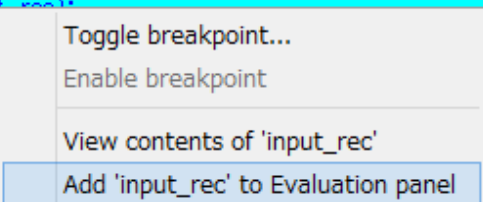
- 9) プログラム中にブレークポイントを設定してみましょう。以下のように現在の実行点である Do ステートメントの行をダブルクリックすると以下のように該当行の左端に赤丸が表示されます。

```
23: read file(SYSUT1) into(input_rec);
24:
25: do while(EOF_SYSUT1 = '0'b);
26:   CIO = input_rec;
27:   WRITE FILE(SYSUT2) FROM (CIO);
28:   read file(SYSUT1) into(input_rec);
```

- 10) 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。

- 11) 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 Evaluation panel]を選択します。

```
21: read file(SYSUT1) into(input_rec);
22:
23: do while(EOF_SYSUT1 = '0'b);
24:   CIO = input_rec;
25:   WRITE FILE(SYSUT2) FROM (CIO);
26:   read file(SYSUT1) into(input_rec);
27: end;
28:
```



A context menu is displayed over the code editor, listing several actions: 'Toggle breakpoint...', 'Enable breakpoint', 'View contents of 'input_rec'', and 'Add 'input_rec' to Evaluation panel'. The 'Add 'input_rec' to Evaluation panel' option is highlighted with a blue selection bar.

- 12) 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。

No	Owner	From
	Current	JCLDEMO#25
18	JCLDEMO	_lpi_main+0x2a
17	_lpi_main	systemMVS+0x5a
16	systemMVS	0x2b6c332
15	0x2b6c332	0x10003521
14	0x10003521	0x1008017e
13	0x1008017e	0x28d4453
12	0x28d4453	0x23574dd

X	Expr	Value	Ww
<input type="checkbox"/>	input_rec	'000030000Belcher	Pau...

- 13) 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。

14) 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。



15) 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。