

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 目的

当チュートリアルでは PL/I ソースにおけるバッチジョブ実行の下記手順の習得を目的としています。

- 1) プロジェクトの作成方法
- 2) プロジェクトプロパティの設定方法
- 3) 既存プログラムの取り込みとビルド方法
- 4) Enterprise Server 設定方法
- 5) Enterprise Server 起動と確認方法
- 6) JCL の実行と確認方法
- 7) プロシージャライブラリの作成方法
- 8) PL/I バッチプログラムの実行方法
- 9) PL/I バッチプログラムのデバッグ方法

9.2 チュートリアルの前提と準備

本チュートリアル実施に伴う例題プログラムに関連する資源を用意します。

- 1) キットに添付されている PLIJCLtutorial.zip を C:¥ 直下へ解凍し、C:¥ PLIJCLtutorial を作成します。
- 2) 作業用に C:¥work というフォルダを作成しておきます。

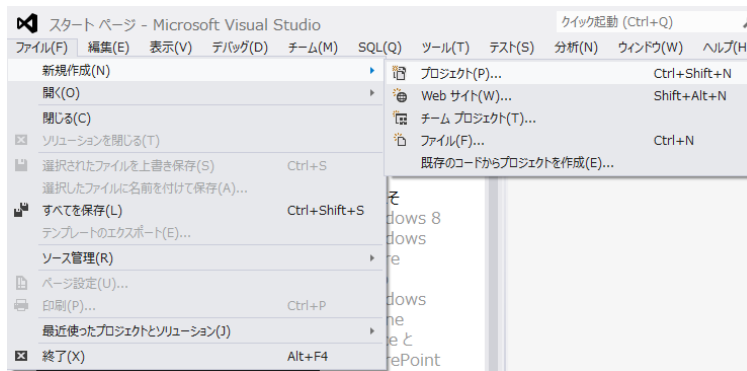
9.3 プロジェクトの作成

1) スタートアイコンから Enterprise Developer for Visual Studio 2012 を起動します。



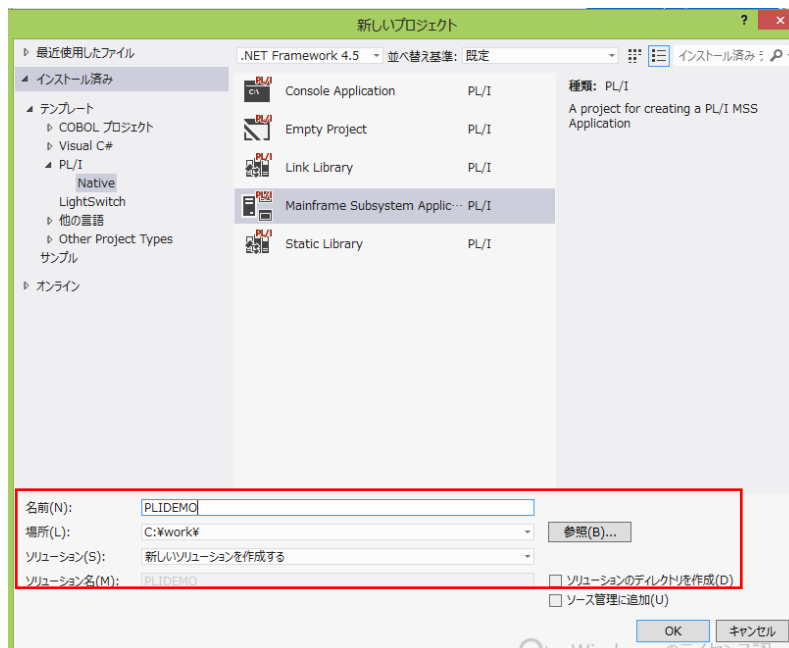
2) 新しいプロジェクトを作成します。

プログラウメニューの [ファイル]> [新規作成]> [プロジェクト] を選択します。



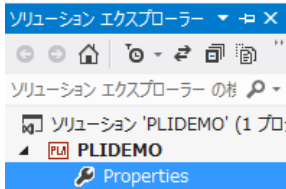
3) プロジェクトの種類と名前、場所を指定して作成します。

- ① ペインツリーの [テンプレート]> [PL/I]> [Native] 内から “Mainframe Subsystem Application” を選択します。
- ② [名前] は “PLIDEMO” を入力します。
- ③ [場所] は 作成した C:\%work フォルダを指定します。
- ④ [ソリューションのディレクトリを作成] のチェックを外して [OK] をクリックします。



9.4 プロジェクトプロパティの設定

- 1) ソリューション エクスプローラで作成したプロジェクト内の [Properties] をダブルクリックし、プロパティ設定ウィンドウを表示します。



- 2) ツリービューの [PL/I] を選択して下記項目の確認と設定を行います。

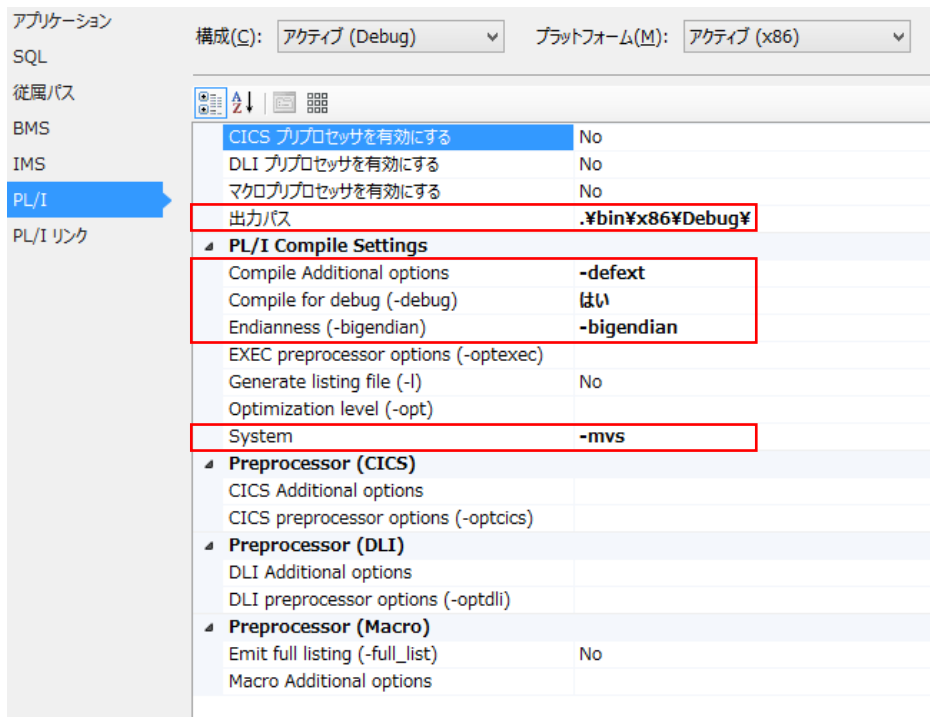
- ① [General]> [Output Path] (出力パス) を確認します。
- ② [PL/I Compile Setting]> [Compile for debug] が “はい” であることを確認します。



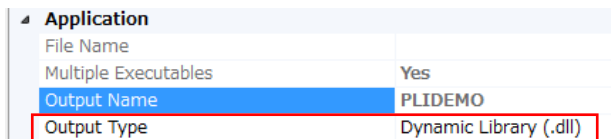
重要事項

PL/I プログラムの実行にはデバッグ用オブジェクトが必要になるため、この指定は必須です。

- ③ [PL/I Compile Setting]> [Compile Addition options] へ “-defext” を入力します。
- ④ [PL/I Compile Setting]> [Endianness] へ “-bigendian” を設定します。
- ⑤ [PL/I Compile Setting]> [System] へ “-mvs” を設定します。



- 3) ツリービューの [Application] を選択して [Output Type] が “.dll” であることを確認します。

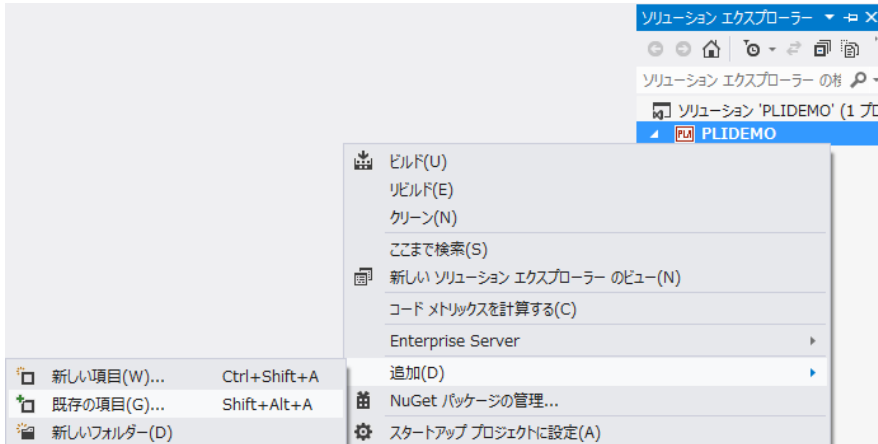


- 4) プロジェクトのプロパティを保存して、設定ウィンドウを閉じます。

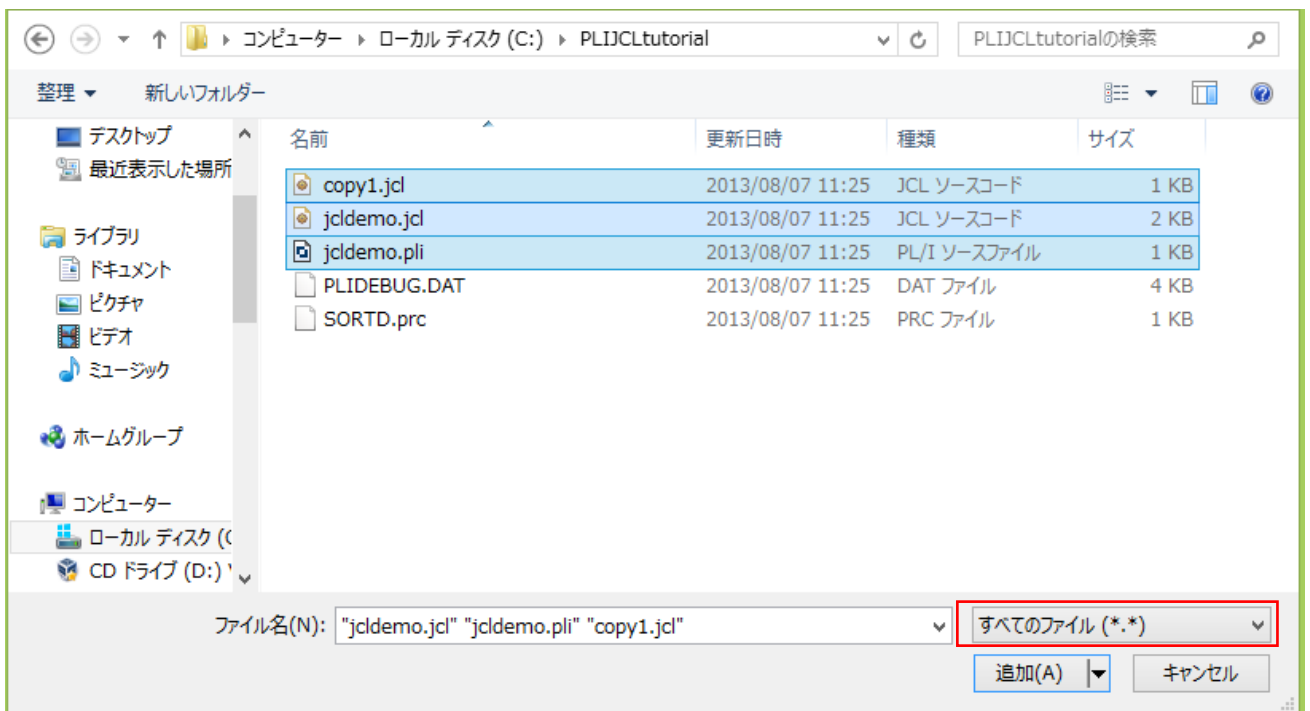


9.5 既存プログラムの取り込みとビルド

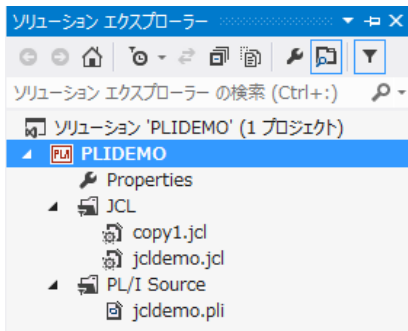
- 1) ソリューション エクスプローラでプロジェクトを右クリックし、[追加]>[既存の項目] を選択すると、ファイル 選択のためのエクスプローラが表示されます。



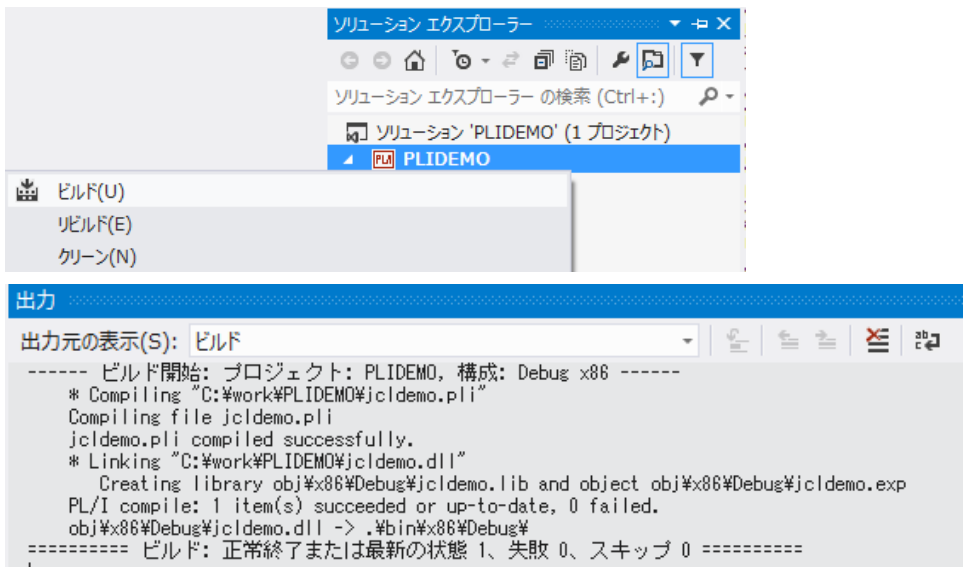
- 2) [すべてのファイル]を選択して、展開した C:¥PLIJCLtutorial 直下に存在する “.jcl” と “.pli” 拡張子を持つファイルを選択し [追加] をクリックします。



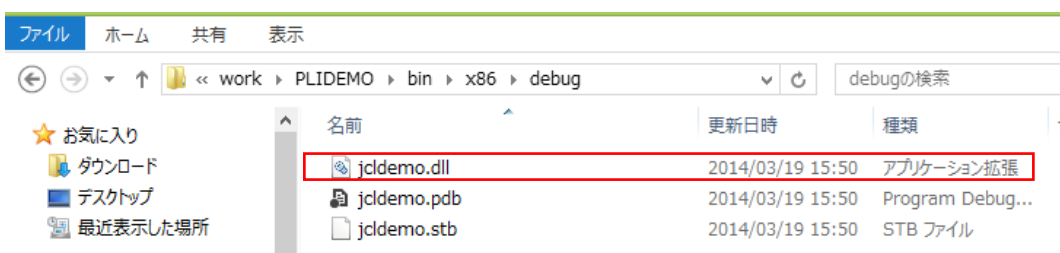
3) ソリューション エクスプローラへ追加されたファイルが表示されます。



4) ソリューション エクスプローラでプロジェクトを右クリックし [ビルド] を実行すると、正常終了となったコンパイル結果が出力ウィンドウへ表示されます。



5) プロパティで設定した出力パスには DLL ファイルが作成されています。

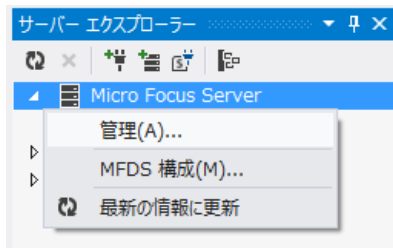


9.6 Enterprise Server の設定

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(Enterprise Server)を作成します。
Enterprise Developer には開発用の Enterprise Server と呼ばれるミドルウェアが内蔵されており、これが持つメインフレームランタイム環境を使用してメインフレームアプリケーションのテスト・デバッグを行います。
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) 管理コンソールを表示します。

[サーバー エクスプローラ]> [Micro Focus Server] を右クリックして [管理] を選択します。



- 2) 新しい Enterprise Server を作成するために、管理コンソール左下の [追加] ボタンをクリックします。

A screenshot of the 'Enterprise Server Administration' web console. The page title is 'Enterprise Server Administration' with the URL 'TOK-kt-W8v1.microfocus.com'. The main content area shows a table of servers. The table has columns for 'タイプ' (Type), '名前' (Name), 'ステータス' (Status), and '説明' (Description). Two servers are listed: 'DB2DEMO' and 'DBDEMO64', both with status '停止' (Stopped). The '追加' (Add) button is highlighted with a red box in the bottom left corner of the table area. The left sidebar contains navigation options like 'Home', 'アクション', '構成', '表示', and 'ヘルプ'.

3) 追加ウィンドウの 1/3 ページには下記内容を設定します。

- ① サーバー名 : "PLIDEMO" を入力します。
- ② 動作モード : 32-bit を選択します。
- ③ [次へ] ボタンをクリックします。

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

ステータス
MDS00001 OK
Wed Mar 19 17:18:09 2014
[mfuser] [Page id: a001]

Enterprise Server Administration > サーバー追加
TOK-kt-W8v1.microfocus.com
Version 1.16.42
Enterprise Server (MSS LICENSED)

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 you can change it when copying or importing a server.

キャンセル 次へ >>

4) 追加ウィンドウの 2/3 ページには下記内容を設定します。

- ① サーバータイプ : "Micro Focus Enterprise Server with Mainframe Subsystem Support" を選択します。
- ② [次へ] ボタンをクリックします。

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
MDS00001 OK
Wed Mar 19 17:23:00 2014
[mfuser] [Page id: a101]

Enterprise Server Administration > サーバー追加
TOK-kt-W8v1.microfocus.com
Version 1.16.42
Enterprise Server (MSS LICENSED)

サーバー追加 (Page 2 of 3):

サーバー名: PLIDEMO

サーバータイプ:
 Micro Focus Enterprise Server
An enterprise server that provides an execution environment for COBOL application programs running as services in a service orientated architecture.
 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) 追加ウィンドウの 3/3 ページには下記内容を設定します。

- ① ローカルコンソールを表示：チェックをして有効にします。
- ② TN3270 リスナーの作成：チェックをはずして無効にします。
- ③ [追加] ボタンをクリックします。

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"/>	システムアバート時ダンプ: <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

構成情報

説明

Micro Focus Enterprise Server

キャンセル << 戻る **追加**

6) "PLIDEMO" が追加されました。次に設定を行うため [編集] ボタンをクリックします。

1 - 1 of 1 out of 9 servers

Filter タイプ: All 名前: PLI*

Repository: file:///C:/ProgramData/Micro Focus/Enterprise Developer/MFDS/

タイプ	名前	ステータス	説明
MFES (MSS)	PLIDEMO	停止 開始...	Micro Focus Enterprise Server

編集... 追加

7) [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て、下記項目を設定後 [Apply] ボタンをクリックします。

入力内容には改行を入れないように注意してください。

- ① [JES プログラムパス] : 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。開発プロジェクトの bin ディレクトリを指定します。
- ② [システムカタログ] : 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とファイル名を指定します。
- ③ [データセットの省略時刻ロケーション] : 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を指定します。
- ④ [システムプロシージャライブラリ] : 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を指定します。

プロパティ... 構成 診断... 過去の統計

一般 XAリソース (0) MSS... (✓) MQ... スクリプト アクセス権 セキュリティ

メインフレーム サブシステム サポート有効:

CICS (✓) JES... (✓) IMS... PL/I (✓)

General Initiators (1) Printers (0)

ジョブ入力サブシステム有効:

JESプログラムパス:
C:\work\PLIDEMO\bin\x86\debug

システムカタログ:
C:\work\PLIDEMO\DATASET\CATALOG.DAT

データセットの省略時刻ロケーション:
C:\work\PLIDEMO\DATASET

システムプロシージャライブラリ:
SYS1.PROCLIB

Fileshare Configuration Location:

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。

8) 上記 "JES" にて指定した "DATASET" フォルダを指定したパスに作成します。

ファイル ホーム 共有 表示

ローカル ディスク (C:) > work > PLIDEMO > DATASET

名前	更新日時	種類
bin	2014/03/19 14:56	ファイル フォルダー
obj	2014/03/19 15:50	ファイル フォルダー
Properties	2014/03/19 14:43	ファイル フォルダー
DATASET	2014/03/20 11:13	ファイル フォルダー

9) ジョブ実行クラスを定義するため、[JES...]>[Initiators] へ下記項目を設定後 [追加] をクリックします。

- ① [名前]: 任意の名前を入力します。
- ② [Class]: 実行受け入れクラスを指定します。
- ③ [説明]: 任意の説明を入力します。

CICS (✓) JES... (✓) IMS... PL/I

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

! 注意事項

1. サーバ開始時に起動する指定です。実行した JOB CLASS がこの CLASS に含まれていない場合は、ジョブは正常に実行されません。
2. サーバ開始後 JOB CLASS がイニシエータの CLASS に含まれていない場合は、ESMAC 画面から[Resources]>[JES]>[Control] で動的に作成かつ反映することが可能ですが一時的な追加となるため、サーバ再起動時には無効となります。

10) 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... PL/I

一般 Initiators (1) Printers (0)

	名前	クラス	説明
編集...	INITABC	ABC	クラスABCのイニシエータ

追加

11) [サーバー]>[プロパティ]>[MSS...]>[PL/I] タブの下記項目を設定後 [Apply] をクリックします。

- ① [PL/I 有効] : チェックをして有効にします。
- ② [Codewatch ソースパス] : CodeWatch デバッガ呼び出し時に指定されたディレクトリ内で何も見つからなかった場合に使用するデバッガ用のソース・ファイルの場所を指定します。プロジェクトディレクトリを指定します。
- ③ [Codewatch STB パス] : CodeWatch デバッガ呼び出し時に指定されたディレクトリ内で何も見つからなかった場合に使用するデバッガ用の STB ファイルの場所を指定します。出力ファイルディレクトリを指定します。
- ④ PL/I 構成ディレクトリ : プロジェクトディレクトリを指定します。
- ⑤ [Apply] ボタンをクリックします。

一般 XAリソース (0) MSS... (✓) MQ... スクリプト アクセス権 セキュリ:

メインフレーム サブシステム サポート有効:

CICS (✓) JES... (✓) IMS... PL/I (✓)

一般

PL/I 有効:

Prompt for PLITEST attachment:

Codewatch ソース パス:
C:\work\PLIDEMO

Codewatch STB パス:
C:\work\PLIDEMO\bin\x86\debug

PL/I 構成ディレクトリ:
C:\work\PLIDEMO

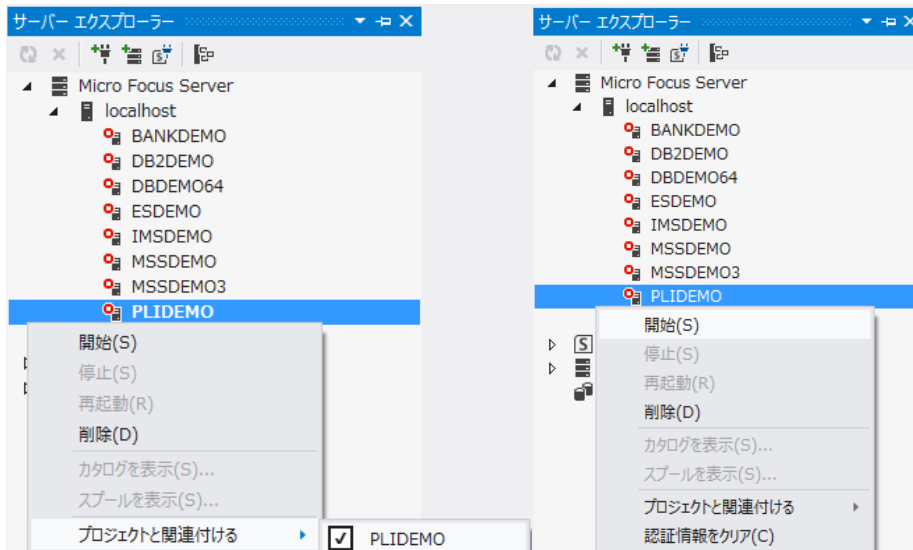
Apply

9.7 Enterprise Server の起動と確認

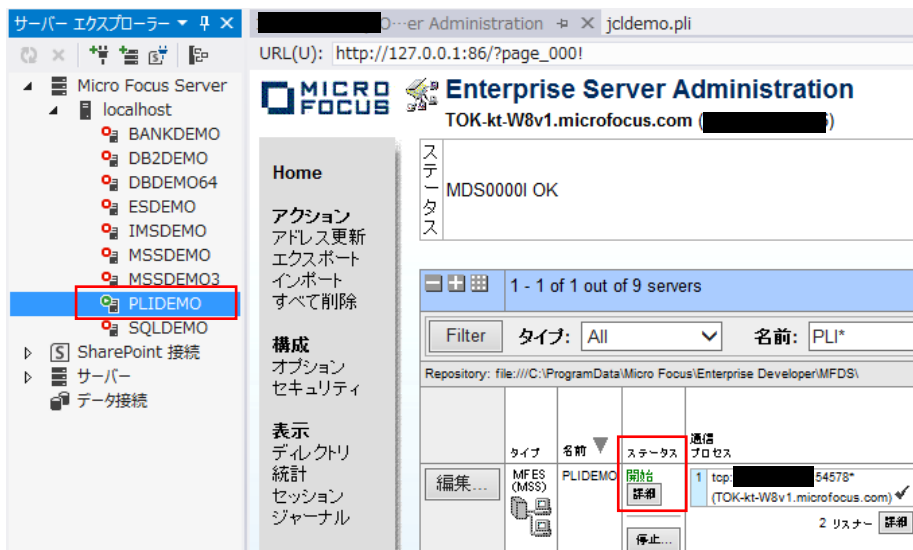
- 1) サーバーエクスプローラ内に新規作成された PLIDEMO が表示されていることを確認します。表示されていない場合は [Micro Focus Server] を右クリックして [最新の情報に更新] を選択、または左上アイコンをクリックしてください。



- 2) PLIDEMO を右クリックし [プロジェクトと関連付ける] を選択して PLIDEMO プロジェクトを指定後、再度右クリックを行い [開始] を選択します。



- 3) PLIDEMO が開始状態になっていることを確認後 [詳細] ボタンをクリックします。



- 4) [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可能です。

Server PLIDEMO [開始 ✓]

サーバー… リスナー (2) サービス (4) ハンドラ (3) パッケージ (0)

プロパティ… 構成 診断… 過去の統計

トレース ダンプ ESコンソール CSコンソール

画面更新 Show entries from 1 to 10 of 47 total entries
 Show last 10 lines

Entry Event Show Entire Log

38	140320 13324459 16532 PLIDEMO JES000042I SSTM not enabled: CICS 13:32:44
39	140320 13324481 16532 PLIDEMO CASSI5001I PLTPI Phase 1 - No PLT Specified 13:32:44
40	140320 13324503 16532 PLIDEMO CASSI5040I Active SEP memory strategy set to x'00000001', retain count 100 13:32:44
41	140320 13324525 17264 PLIDEMO CASSI1744I MFPLI support loaded successfully 13:32:45
42	140320 13324547 16880 PLIDEMO CASSI1744I MFPLI support loaded successfully 13:32:45
43	140320 13324547 17264 PLIDEMO CASSI1600I SEP initialization completed successfully 13:32:45
44	140320 13324568 11260 PLIDEMO CASSI1744I MFPLI support loaded successfully 13:32:45
45	140320 13324568 16880 PLIDEMO CASSI1600I SEP initialization completed successfully 13:32:45
46	140320 13324590 11260 PLIDEMO CASBJ0005I Batch initiator started for job classes "ABC" 13:32:45
47	140320 13324590 16880 PLIDEMO CASSI5021I PLTPI Phase 2 - No PLT Specified 13:32:45

ログ全体を表示します。

http://10.18.1...78/MF_GETCASLOG... Administration jdemo.pli

URL(U): http://10.18.11.153:54578/MF_GETCASLOG

```

140320 13323572 17248 PLIDEMO CASSI0000I Server manager initialization started 13:32:35
140320 13323776 17248 PLIDEMO CASSI4005I Retrieving ES configuration from WF06 (127.0.0.1:86) 13:32:36
140320 13323961 17248 PLIDEMO CASSI0012I Region running in 32 bit mode 13:32:38
140320 13324003 17248 PLIDEMO CASSE0011I Region is running with Resource security disabled. 13:32:38
140320 13324025 17248 PLIDEMO CASSI0001I Region will use SYSID $IWP 13:32:40
140320 13324047 CASD01075I ES TSC Service Process created for Server PLIDEMO, process-id = 17176 13:32:40
140320 13324059 CASD00127I SEP 0001 created for ES PLIDEMO, process-id = 16532 13:32:40
140320 13324080 17176 PLIDEMO CASTS0007I ES TSC Service Process initialization started 13:32:40
140320 13324090 CASD00127I SEP 0002 created for ES PLIDEMO, process-id = 16880 13:32:40
140320 13324112 16532 PLIDEMO CASSI1500I SEP initialization started 13:32:40
140320 13324112 17176 PLIDEMO CASTS0015I ES TSC Log and Intra-Partition IS/TD: "C:\Users\Marlot\Documents\Micro Focus User\Enterprise Developer for Visual Studio 2012\WORKAREA\PLIDEMO\LOGSTTD0.dat" 13:32:40
140320 13324112 CASD01038I ES Communications Server created, ES PLIDEMO, process-id = 16808 13:32:41
140320 13324115 16880 PLIDEMO CASSI1500I SEP initialization started 13:32:40
140320 13324117 17176 PLIDEMO CASTS0013I ES TSC Log and Intra-Partition IS/TD cold started 13:32:41
140320 13324117 17248 PLIDEMO CASG01000I ES concurrent request limit: 0000000010 13:32:41
140320 13324139 17176 PLIDEMO CASTS0161I ES TSC Log and Intra-Partition IS/TD File size: 32 blocks ( 2,016,740 bytes) 13:32:41
140320 13324139 17248 PLIDEMO CASSI1000I Server Manager initialization completed successfully 13:32:41
140320 13324161 16880 PLIDEMO CASG05001I Communications interface 01 initialization started 13:32:41
140320 13324161 17176 PLIDEMO CASTS0171I ES TSC Log and Intra-Partition IS/TD running with 1 buffers of 63,020 bytes each 13:32:41
140320 13324163 17176 PLIDEMO CASTS0002I ES TSC Service Process initialization complete 13:32:41
140320 13324163 16880 PLIDEMO CASG05003I Communications interface 01 initialization complete 13:32:41
140320 13324163 17248 PLIDEMO JES000051I Job Entry Subsystem (JES) services initialized 13:32:41
140320 13324204 CASD01060I JES Initiator created for Server PLIDEMO, process-id = 11260 13:32:42
140320 13324226 17248 PLIDEMO CASBJ0023I Batch initiator INITABC: classes) "ABC" 13:32:42
140320 13324248 11260 PLIDEMO CASBJ0006I Batch initiator initialization started 13:32:42
140320 13324248 CASD01071I Administration SEP created for Server PLIDEMO, process-id = 17264 13:32:42
140320 13324304 CASSI1500I SEP initialization started 13:32:42
140320 13324306 16880 PLIDEMO CASG05100I Communications Process instance 01 is ready to accept requests 13:32:43
140320 13324326 16532 PLIDEMO CASSI1425I TS non-recoverable cold started 13:32:43
140320 13324350 16532 PLIDEMO CASSI1426I TS recoverable warm started 13:32:43
140320 13324373 16532 PLIDEMO CASSI1426I TD non-recoverable cold started 13:32:43
140320 13324395 16532 PLIDEMO CASSI1426I TD recoverable warm started 13:32:43
140320 13324417 16532 PLIDEMO CASSI1744I MFPLI support loaded successfully 13:32:43
140320 13324457 16532 PLIDEMO CASSI1600I SEP initialization completed successfully 13:32:44
140320 13324459 16532 PLIDEMO JES000042I SSTM not enabled: CICS 13:32:44
140320 13324461 16532 PLIDEMO CASSI5001I PLTPI Phase 1 - No PLT Specified 13:32:44
140320 13324503 16532 PLIDEMO CASSI5040I Active SEP memory strategy set to "00000001", retain count 100 13:32:44
140320 13324525 17264 PLIDEMO CASSI1744I MFPLI support loaded successfully 13:32:45
140320 13324547 16880 PLIDEMO CASSI1744I MFPLI support loaded successfully 13:32:45
140320 13324547 17264 PLIDEMO CASSI1600I SEP initialization completed successfully 13:32:45
140320 13324568 11260 PLIDEMO CASSI1744I MFPLI support loaded successfully 13:32:45
140320 13324568 16880 PLIDEMO CASSI1600I SEP initialization completed successfully 13:32:45
140320 13324590 11260 PLIDEMO CASBJ0005I Batch initiator started for job classes "ABC" 13:32:45
140320 13324590 16880 PLIDEMO CASSI5021I PLTPI Phase 2 - No PLT Specified 13:32:45

```

出力ウィンドウへもメッセージが表示されます。

出力

出力元の表示(S): 全般

```

CASCD0167I ES Daemon successfully auto-started 13:45:55
CASCD0050I ES "PLIDEMO" initiation is starting 13:45:55
プロジェクト PLIDEMO に関連付けられたサーバーが更新されました。
PLIDEMO 開始の成功

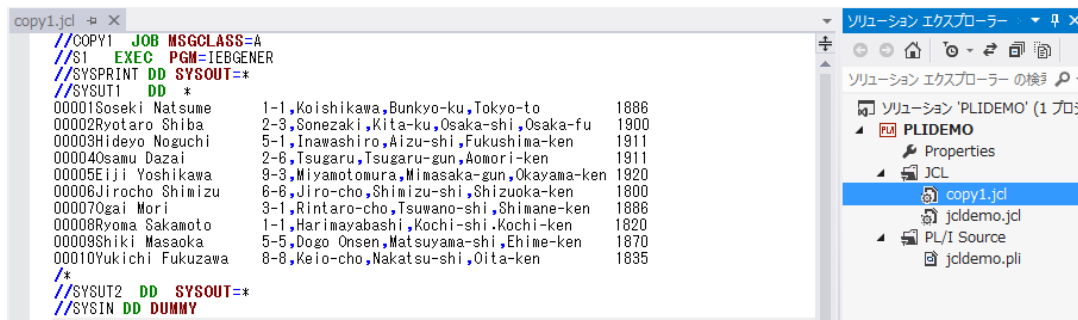
```

注意事項
 いくつかのサービス開始が失敗してもサーバーは開始されますので、ログ内容を確認してください。

9.8 JCLの実行と確認

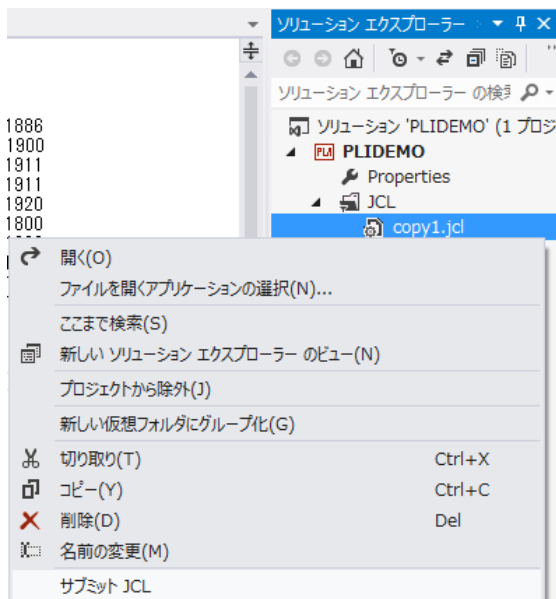
簡単なJCLを PLIDEMO サーバーへ実行してみます。

- 1) ソリューションエクスプローラの 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。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。

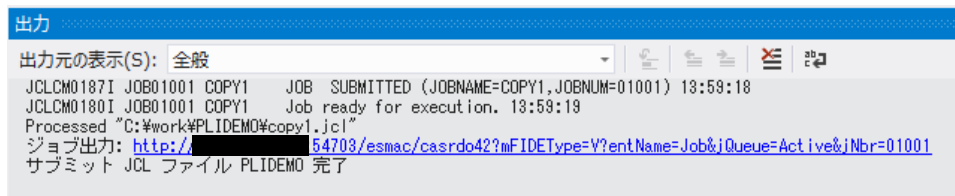


```
//COPY1 JOB MSGCLASS=A
//S1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD *
00001Soseki Natsume 1-1,Koishikawa,Bunkyo-ku,Tokyo-to 1886
00002Ryotaro Shiba 2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 1900
00003Hideoyo Noguchi 5-1,Inawashiro,Aizu-shi,Fukushima-ken 1911
00004Osamu Dazai 2-8,Tsugaru,Tsugaru-gun,Aomori-ken 1911
00005Eiji Yoshikawa 9-3,Miyatomura,Mimasaka-gun,Okayama-ken 1920
00006Jirocho Shimizu 8-8,Jiro-cho,Shimizu-shi,Shizuoka-ken 1800
00007Gai Mori 3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken 1886
00008Ryoma Sakamoto 1-1,Harimayabashi,Kochi-shi,Kochi-ken 1820
00009Shiki Masaoka 5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken 1870
00010Yukichi Fukuzawa 8-8,Keio-cho,Nakatsu-shi,Oita-ken 1835
/*
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
```

- 2) ソリューションエクスプローラの copy1.jcl を右クリックし「サブミット JCL」を選択し、実行します。



- 3) 出カウインドウに結果が表示されます。[ジョブ出力] のリンクをクリックして出カスプールを表示します。



- 4) 出カスプール一覧が表示されます。 JESYSMSG をクリックして実行結果ログを表示します。

JOB01001	Name: COPY1	Status: Output Hold				
<input type="button" value="Release"/>	Class: <input type="text" value="A"/>	Priority: <input type="text" value="00"/>				
<input type="button" value="Update"/>	User: JESUSER	COND: 00000				
JCLCM0188I JOB01001 COPY1 JOB STARTED 13:59:19 JCLCM0182I JOB01001 COPY1 JOB ENDED - COND CODE 0000 13:59:19						
Status	Class	DD Name	Step	Nbr.	Proc. Step	Records
<input type="button" value="Details"/>	Hold	A	JESYSMSG	0		31
<input type="button" value="Details"/>	Ready	A	<u>SYSPRINT</u>	1	S1	4
<input type="button" value="Details"/>	Ready	A	<u>SYSUT2</u>	1	S1	10

```

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: COPY1 User: JESUSER Date: 03/20/14 Time: 13:59:18 *-*
*-* File: $TXRFDIR/T000000010.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 17796

13:59:19 JCLCM0188I JOB STARTED

13:59:19 JCLCM0190I STEP STARTED S1
13:59:19 JCLCM0191I Program MFJGENER is COBOL ASCII Big-Endian NOAMODE.
MFE2014.S0320.S135918.J01001.D00001.SYSPRINT SYSPRINT
C:\WORK\PLIDEMO\DATASET\MFE2014.*5918.J01001.D00001.SYSPRINT.DAT SPOOLED
MFE2014.S0320.S135918.J01001.D00002.SYSUT1 SYSUT1
C:\WORK\PLIDEMO\DATASET\MFE2014.*135918.J01001.D00002.SYSUT1.DAT DELETED
MFE2014.S0320.S135918.J01001.D00003.SYSUT2 SYSUT2
C:\WORK\PLIDEMO\DATASET\MFE2014.*135918.J01001.D00003.SYSUT2.DAT SPOOLED
DUMMY SYSIN
--> 13:59:19 JCLCM0191I STEP ENDED S1 - COND CODE 0000
--> 13:59:19 JCLCM0182I JOB ENDED - COND CODE 0000
  
```

- 5) SYSPRINT をクリックして、IEBGENER ユーティリティの実行ログを確認します。

```

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
  
```

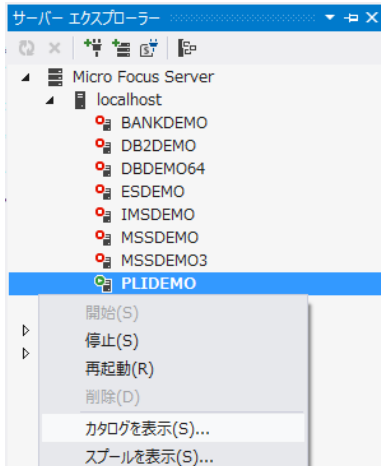
- 6) SYSUT2 をクリックして、出力されたスプールの内容を確認します。

Content-Type: text/plain		
00001Soseki Natsume	1-1,Koishikawa,Bunkyo-ku,Tokyo-to	1886
00002Ryotaro Shiba	2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu	1900
00003Hideyo Noguchi	5-1,Inawashiro,Aizu-shi,Fukushima-ken	1911
00004Osamu Dazai	2-6,Tsugaru,Tsugaru-gun,Aomori-ken	1911
00005Eiji Yoshikawa	9-3,Miyatomura,Mimasaka-gun,Okayama-ken	1920
00006Jirocho Shimizu	6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken	1800
00007Ogai Mori	3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken	1886
00008Ryoma Sakamoto	1-1,Harimayabashi,Kochi-shi,Kochi-ken	1820
00009Shiki Masaoka	5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken	1870
00010Yukichi Fukuzawa	8-8,Keio-cho,Nakatsu-shi,Oita-ken	1835

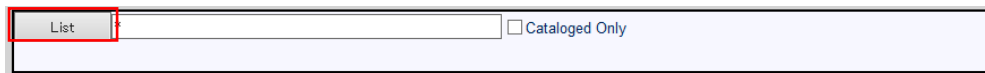
9.9 プロシージャライブラリの作成

本チュートリアル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するため、プロシージャライブラリを作成してプロシージャを配備しておきます。

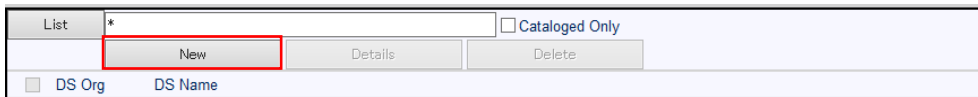
- 1) サーバーエクスプローラから PLIDEMO を右クリックし [カタログの表示] を選択します。



- 2) カタログビューが表示されましたら [List] ボタンをクリックして現状を確認します。

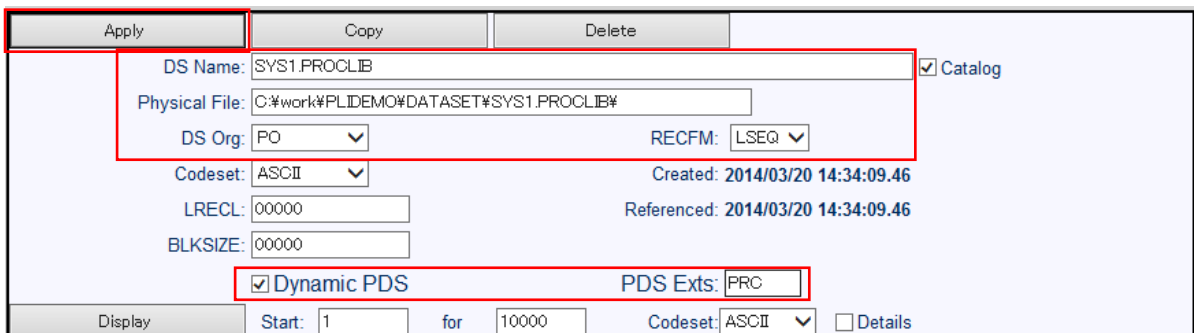


- 3) カタログされているデータセットは何もありません。[New] ボタンをクリックし、新規登録を行います。



- 4) 新規作成ダイアログが現れます。下記項目を設定して [Apply] をクリックします。

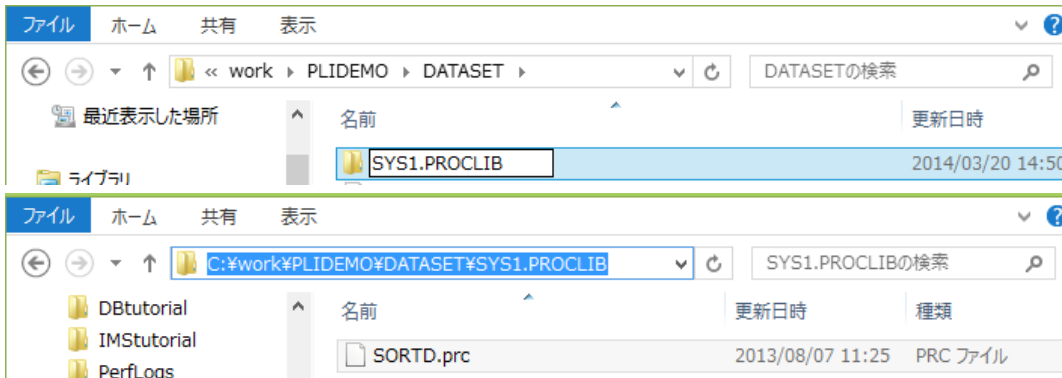
- ① [DS Name] : "SYS1.PROCLIB" を入力します。
- ② [Physical File] : " C:¥work¥PLIDEMO¥DATASET¥SYS1.PROCLIB¥" を入力します。
- ③ [DS Org] : "PO" (パーティションドデータセット) を選択します。
- ④ [RECFM] : "LSEQ" を選択して、行順ファイル形式を指定します。
- ⑤ [Dynamic PDS] : チェックして、動的 PDS を指定します。
- ⑥ [PDS Exts] : "PRC" を入力して、PO が拡張子 .PRC のテキストファイルをフォルダ配下に保持することを指定します。



5) カタログビューを再表示すると、カタログエントリ SYS1.PROCLIB が作成されていることが確認できます。

List	*	<input type="checkbox"/> Cataloged Only	
	New	Details	Delete
<input type="checkbox"/> DS Org	DS Name		
<input type="checkbox"/> PO	SYS1.PROCLIB		DCB
<input type="checkbox"/> DS Org	DS Name		

6) 物理ファイルに指定したフォルダを作成し、直下にパスに C:\\$work\PLIDEMO\DATASET.prc をコピーします。



7) カタログビューで SYS1.PROCLIB をクリックすると、メンバー SYS1.PROCLIB(SORTD) が登録されていることが確認でき、SYS1.PROCLIB(SORTD) をクリックすると内容を表示確認できます。

List	*	<input type="checkbox"/> Cataloged Only	
	New	Details	Delete
<input type="checkbox"/> DS Org	DS Name		
<input type="checkbox"/> PO	SYS1.PROCLIB		DCB
<input type="checkbox"/> PDSM	SYS1.PROCLIB(SORTD)	DCB	Edit
<input type="checkbox"/> DS Org	DS Name		

```
Content-Type: text/plain

//SORTD PROC
//SORT1 EXEC PGM=SORT
//SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
// PEND
```

9.10 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) ソリューションエクスプローラの jcldemo.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。

```

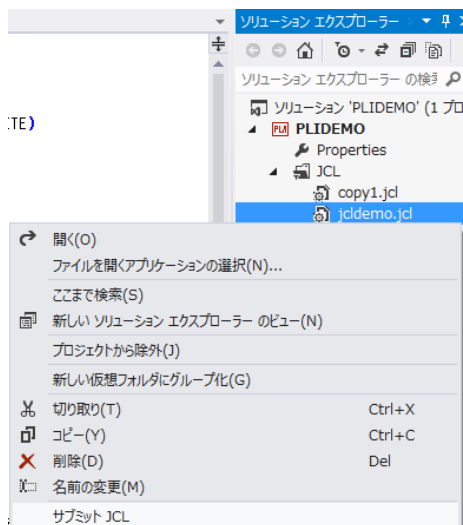
jcldemo.jcl
//JCLDEMO JOB *'JCLDEMO',CLASS=A,MSGCLASS=X
//
// * Delete ADDRBOOK KSDS file for demo
//
//CLEANUP EXEC PGM=IEFBR14
//BBOLOG DD DSN=SYSAD.VSAM.KSDS.TESTFILE,DISP=(MOD,DELETE,DELETE)
//
// * Create THE ADDRBOOK KSDS file
//
//STEP10 EXEC PGM=IDCAMS
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//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 *
00003000DFahringer Dave 1 Redneck Drive Normal
00001000DFooler Clive 2315 Sherman Atlanta
00008000DBelcher Paul 1 Red Sox Way Boston
00002000DSmith Jim 22 Royal Watcher London
00004000DBrewer Anthony 92 Rodent Lane Santa Clara
00005000DFendick Andrew Caravan Keep Lancaster
//
//SORT1.SORTOUT DD DSN=&ADDRDAT,DISP=(NEW,PASS),
//SPACE=(800,(10,10)),DCB=(RECFM=FB,LRECL=80,DSORG=PS),UNIT=SYSDA
//SORT1.SYSIN DD *
SORT FIELDS=(1,9,CH,A)
//
// * Load the ADDRBOOK KSDS using a PLI Program
//
//STEP20 EXEC PGM=JCLDEMO
//SYSOUT DD SYSOUT=*,HOLD=Y
//SYSPRINT DD SYSOUT=*,HOLD=Y,DCB=(RECFM=LSEQ)
//SYSUT1 DD DSN=&ADDRDAT,DISP=OLD
//SYSUT2 DD DSN=SYSAD.VSAM.KSDS.TESTFILE,DISP=OLD
//

```

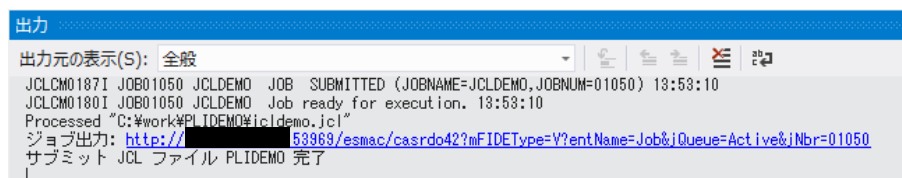
このジョブは4つのステップから構成され下記の様に連携されています。

- 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ッサ 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) "jcldemo.jcl" を実行します。



- 3) 出力ウィンドウに結果が表示されます。[ジョブ出力] のリンクをクリックして出力スプールを表示します。



- 4) 出力スプルー一覧が表示されます。 JESYSMSG をクリックして実行結果ログを表示します。

JOB01050		Name: JCLDEMO	Status: Output Hold				
Release	Class: A		Priority: 00				
Update	User: JESUSER		COND: 00000				
JCLCM0188I JOB01050 JCLDEMO JOB STARTED 13:53:10 JCLCM0182I JOB01050 JCLDEMO JOB ENDED - COND CODE 0000 13:53:10							
	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

- 5) 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。

```
Content-Type: text/plain
*****
Micro Focus ESJCL  ASCII  JES2 Version ED22.00.00_022
Copyright (C) 1997-2013 Micro Focus. All rights reserved.
Job: 01050 Name: JCLDEMO User: JESUSER Date: 03/24/14 Time: 13:53:10
File: $TXRFDIR/T000000062.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 /** Sort the input data
23 /**
24 //SORTSTEP EXEC SORTD
  XXSORTD PROC
  XXSORT1 EXEC PGM=SORT
25 //SORT1.SORTIN DD *
33 //SORT1.SORTOUT DD DSN=AMPADDRDAT,DISP=(NEW,PASS),
34 // SPACE=(800,(10,10)),DCB=(RECFM=FB,LRECL=80,DSORG=PS),UNIT=SYSDA
35 //SORT1.SYSIN DD *
37 /**
38 /** Load the ADDRBOOK KSDS using a PLI Program
39 /**
```

```
*** Execution on Server PLIDEMO Process 16340

13:53:10 JCLCM0188I JOB STARTED

13:53:10 JCLCM0190I STEP STARTED CLEANUP
SYSAD.VSAM.KSDS.TESTFILE BBOLOG
C:\WORK\PLIDEMO\DATASET\SYSAD.VSAM.KSDS.TESTFILE.DAT DELETED
---> 13:53:10 JCLCM0191I STEP ENDED CLEANUP - COND CODE 0000

13:53:10 JCLCM0190I STEP STARTED STEP10
13:53:10 JCLCM0199I Program MFJAMS is COBOL ASCII Big-Endian NOAMODE.
MFE2014.S0324.S135310.J01050.D00001.SYSOUT SYSOUT
C:\WORK\PLIDEMO\DATASET\MFE2014.*135310.J01050.D00001.SYSOUT.DAT REMOVED
MFE2014.S0324.S135310.J01050.D00002.SYSPRINT SYSPRINT
C:\WORK\PLIDEMO\DATASET\MFE2014.*5310.J01050.D00002.SYSPRINT.DAT SPOOLED
MFE2014.S0324.S135310.J01050.D00003.AMSDUMP AMSDUMP
C:\WORK\PLIDEMO\DATASET\MFE2014.*35310.J01050.D00003.AMSDUMP.DAT REMOVED
MFE2014.S0324.S135310.J01050.D00004.SYSIN SYSIN
C:\WORK\PLIDEMO\DATASET\MFE2014.*S135310.J01050.D00004.SYSIN.DAT DELETED
---> 13:53:10 JCLCM0191I STEP ENDED STEP10 - COND CODE 0000

13:53:10 JCLCM0190I STEP STARTED SORTSTEP.SORT1
13:53:10 JCLCM0199I Program MFJSORT is COBOL ANS85 ASCII Big-Endian NOAMODE.
MFE2014.S0324.S135310.J01050.D00005.SORTIN SORTIN
C:\WORK\PLIDEMO\DATASET\MFE2014.*135310.J01050.D00005.SORTIN.DAT DELETED
MFE2014.S0324.S135310.J01050.ANDAND.ADDRDAT SORTOUT
C:\WORK\PLIDEMO\DATASET\MFE2014.*35310.J01050.ANDAND.ADDRDAT.DAT PASSED
MFE2014.S0324.S135310.J01050.D00006.SYSIN SYSIN
C:\WORK\PLIDEMO\DATASET\MFE2014.*S135310.J01050.D00006.SYSIN.DAT DELETED
MFE2014.S0324.S135310.J01050.D00007.SYSOUT SYSOUT
C:\WORK\PLIDEMO\DATASET\MFE2014.*135310.J01050.D00007.SYSOUT.DAT SPOOLED
---> 13:53:10 JCLCM0191I STEP ENDED SORTSTEP.SORT1 - COND CODE 0000
```

```
13:53:10 JCLCM0190I STEP STARTED STEP20
13:53:10 JCLCM0220I Job restart will not be permitted at this step.
13:53:10 JCLCM0199I Program JCLDEMO is Open PL/I ASCII Big-Endian NOAMODE.
MFE2014.S0324.S135310.J01050.D00008.SYSPRINT SYSPRINT
C:\WORK\PLIDEMO\DATASET\MFE2014.*5310.J01050.D00008.SYSPRINT.DAT SPOOLED
MFE2014.S0324.S135310.J01050.ANDAND.ADDRDAT SYSUT1
C:\WORK\PLIDEMO\DATASET\MFE2014.*35310.J01050.ANDAND.ADDRDAT.DAT DELETED
SYSAD.VSAM.KSDS.TESTFILE SYSUT2
C:\WORK\PLIDEMO\DATASET\SYSAD.VSAM.KSDS.TESTFILE.DAT RETAINED
---> 13:53:10 JCLCM0191I STEP ENDED STEP20 - COND CODE 0000
---> 13:53:10 JCLCM0182I JOB ENDED - COND CODE 0000
```

- 6) 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。

```
Content-Type: text/plain

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))                  -
JCLAM01131(00) - ENTRYNAME DEFINED [SYSAD.VSAM.KSDS.TESTFILE]
```

- 7) SORTSTEP の SYSOUT をクリックすると、以下のように ソート実行結果を確認することができます。

```
Content-Type: text/plain

Micro Focus MFJSORT ユーティリティ 3.0.00

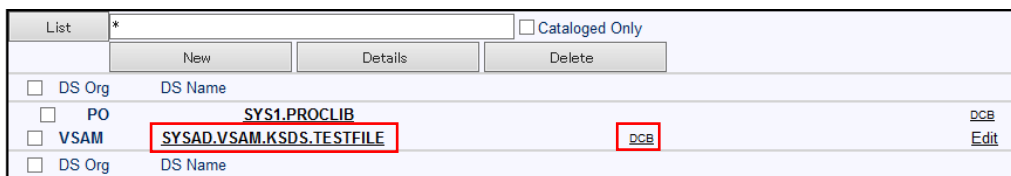
      SORT FIELDS=(1,9,CH,A)
SORT204I: ***** ソート結果 *****
SORT205I: INPUT   ファイル 'SORTIN'
              入力レコード              6 件
              使用レコード              6 件
SORT206I: OUTPUT  ファイル 'SORTOUT'
              使用レコード              6 件
              出力レコード              6 件
SORT399I: Micro Focus MFJSORT ユー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

- 8) 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。

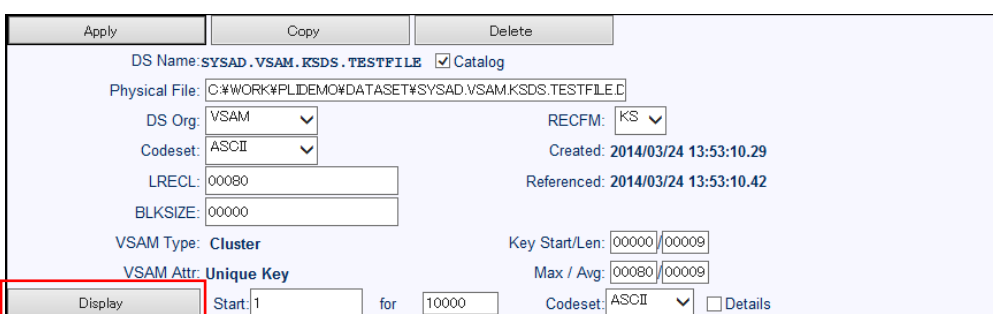
```
Content-Type: text/plain

1Starting jcldemo
done with jcldemo
```

- 9) ジョブ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。 [DCB] をクリックすると DCB 情報が確認できます。



- 10) DCB 情報の [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。



[レコード内容]

Content-Type: text/plain				
000010000000010000	Poole	Clive	2315 Sherman	Atlanta
000020000000020000	Smith	Jim	22 Royal Watcher	London
000030000000030000	Fahringer	Dave	1 Redneck Drive	Normal
000040000000040000	Brewer	Anthony	92 Rodent Lane	Santa Clara
000050000000050000	Fendick	Andrew	Caravan Keep	Lancaster
000060000000060000	Belcher	Paul	1 Red Sox Way	Boston

9.11 PL/I バッチプログラムのデバッグ

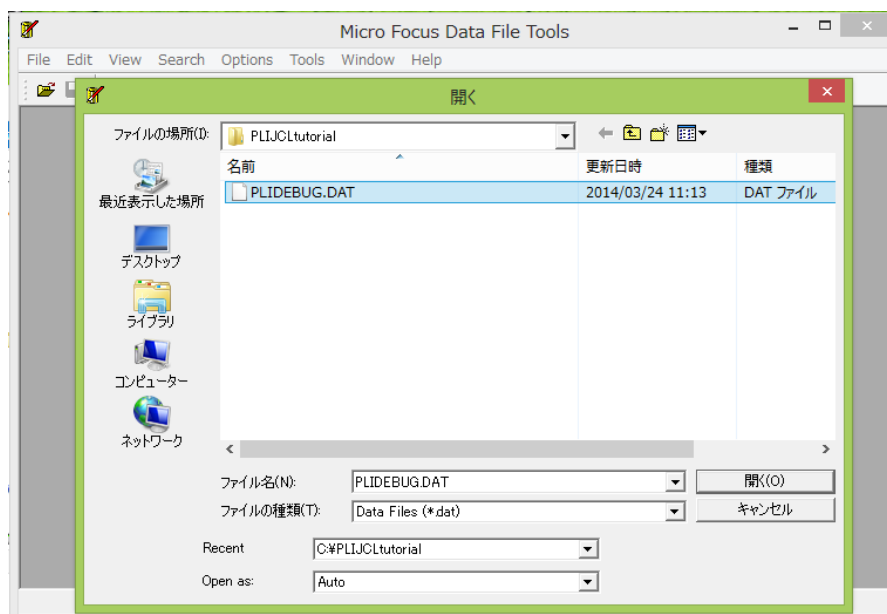
JCL から起動される PL/I プログラムをステップインデバッグします。

- 1) PL/I プログラムを実行するために必要なファイルを用意します。スタートアイコンから 「データファイルツール」 を起動します。

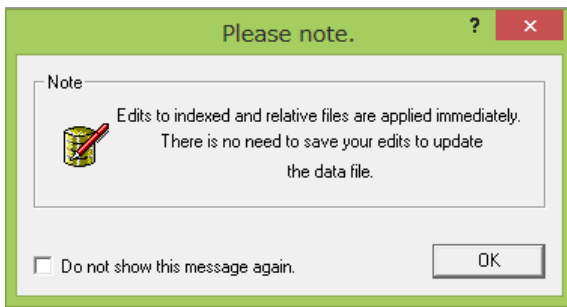


- 2) JES リージョン配下で実行される PL/I プログラムのデバッグは、PLIDEBUG.DAT という構成ファイルにより指示されます。このファイルは、インストール直後は存在しませんが PL/I プログラムを最初に実行した際に作成されます。当チュートリアルでは構成済みの PLIDEBUG.DAT ファイルを使用します。

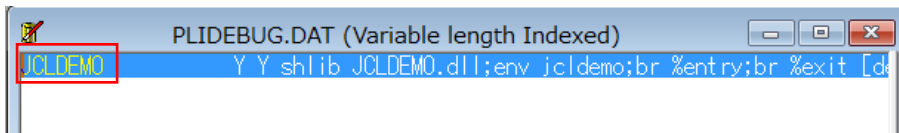
[FILE]> [OPEN] を開き C:\%PLIJCLtutorial に存在する “PLIDEBUG.DAT” を選択してください。



- 3) 下記警告には [OK] をクリックします。変更時は確認を行わずに反映されますので注意してください。



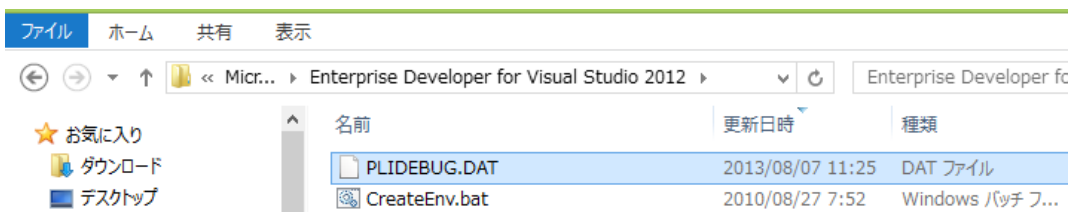
- 4) 以下のように 1 件のレコードが表示されます。先頭にはデバッグ対象プログラム名を指定するため今回は "JCLDEMO" と記述します。これにより "JCLDEMO.pli" 実行時にデバッグセッションが自動的に開始するよう設定されます。



重要事項

このファイルは主キーを先頭を持つ索引ファイルで、キーにはデバッグ対象プログラム名を指定します。デバッグオプションの値 ("Y"以降) は 「21 ケタ目」 から入力します。ファイルを開いているとロックされた状態になりデバッグができなくなるため、実行時は必ず閉じてください。

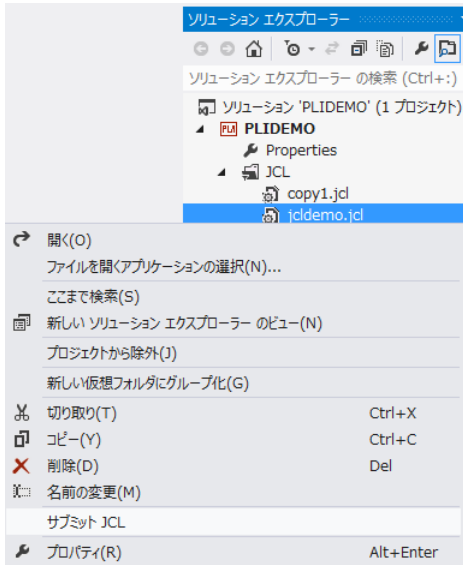
- 5) [File]>[Exit] でデータファイルツールを終了します。
- 6) 上記で確認した "C:¥PLIJCLtutorial¥PLIDEBUG.DAT" ファイルを製品インストール先フォルダへ上書きコピーします。デフォルト位置は "C:¥Program Files (x86)¥Micro Focus¥Enterprise Developer for Visual Studio 2012" です。



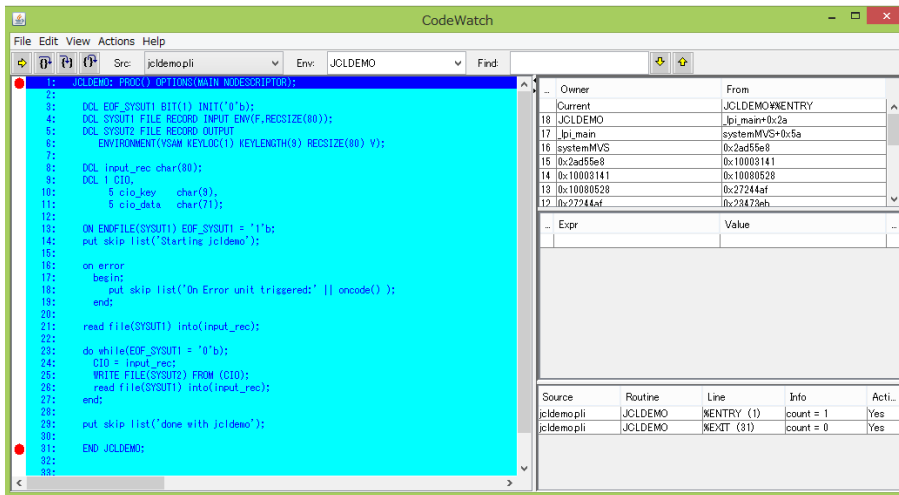
重要事項

正当な場所に "PLIDEBUG.DAT" が存在しないと、PL/I プログラム実行時にエラーとなります。

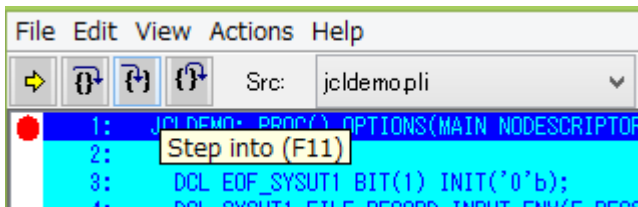
7) ソリューションエクスプローラの jcldemo.jcl を右クリックし 「サブミット JCL」 を選択し、実行します。



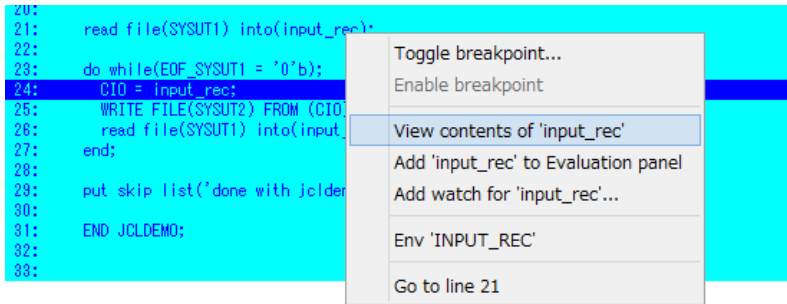
8) PL/I デバッガである CodeWatch が起動します。



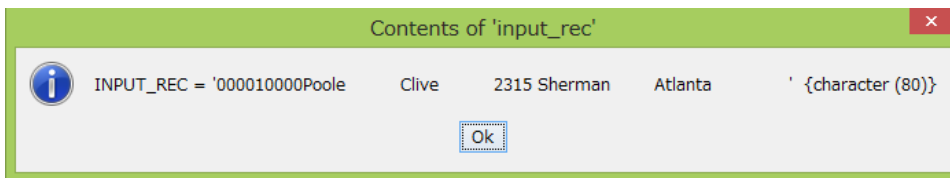
9) 左上のボタンバーの中から [Step Into] または F11 をクリックすると PL/I のステートメントを1ステップ実行することが可能です。



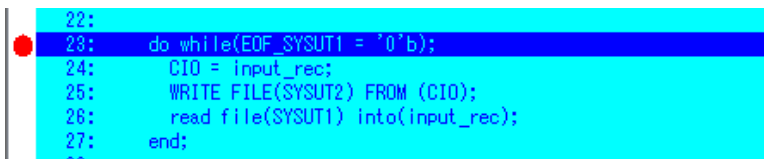
- 10) Read ステートメントの次の Do まで実行を進めます。SYSUT1 から読まれたレコードを格納している input_rec を右クリックし、[View contents of ...] を選択します。



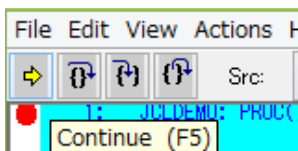
- 11) input_rec の内容がポップアップ表示され、読み込んだレコード内容の確認が可能です。



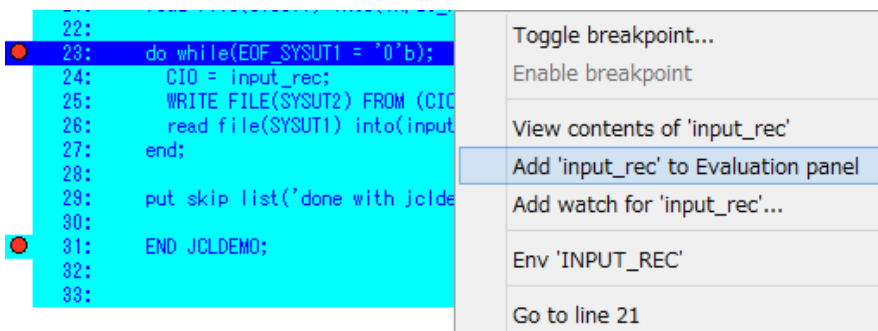
- 12) Do ステートメントの行をダブルクリックすることによりブレークポイントが設定され、該当行の左端に赤丸が表示されます。



- 13) ボタンバー内の [Continue] ボタンまたは F5 をクリックします。



- 14) 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。

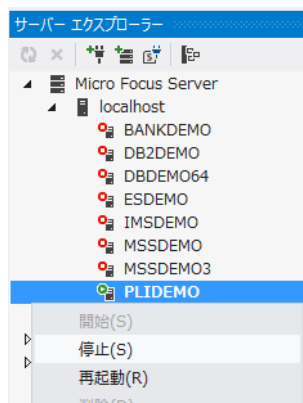


15) input_rec の内容がウォッチパネルに表示され、内容の確認が容易になります。

...	Owner	From	
	Current	JCLDEMO#23	^
18	JCLDEMO	_lpi_main+0x2a	
17	_lpi_main	systemMVS+0x5a	
16	systemMVS	0x2ad55e8	
15	0x2ad55e8	0x10003141	
14	0x10003141	0x10080528	
13	0x10080528	0x27244af	
12	0x27244af	0x23473eb	v
...	Expr	Value	...
<input type="checkbox"/>	input_rec	'00001000Poole Clive	...

16) 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。

17) 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。



18) PLIDEMO が停止したら Visual Studio 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。