

# 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

### 5. Enterprise Developer for Visual Studio 2012 on Windows8 : JCL チュートリアル

#### 5.1 目的

当チュートリアルでは、JCL を使用したアプリケーションの実行と構築手順習得を目的としています。

- 1) ソリューションとプロジェクトの作成方法
- 2) プロジェクトへ既存プログラムの取り込み方法
- 3) ソースプロパティの設定とソースのビルド方法
- 4) Enterprise Server の構築方法
- 5) 簡易 JCL の実行と結果の確認方法
- 6) プロシージャライブラリの作成方法
- 7) バッチプログラムの実行方法
- 8) バッチプログラムの動的デバッグ方法

## 5.2 チュートリアルの準備

本チュートリアルの例題プログラムに関連する資源を用意します。CICSチュートリアルを既に終了している場合には、下記準備をする必要はありません。

- 1) キットに添付されている Tutorials.zip を C:¥ の直下へ解凍し、C:¥Tutorials を作成します。
- 2) 作業用に使用する C:¥work フォルダを作成します。

## 5.3 Enterprise Developer の起動

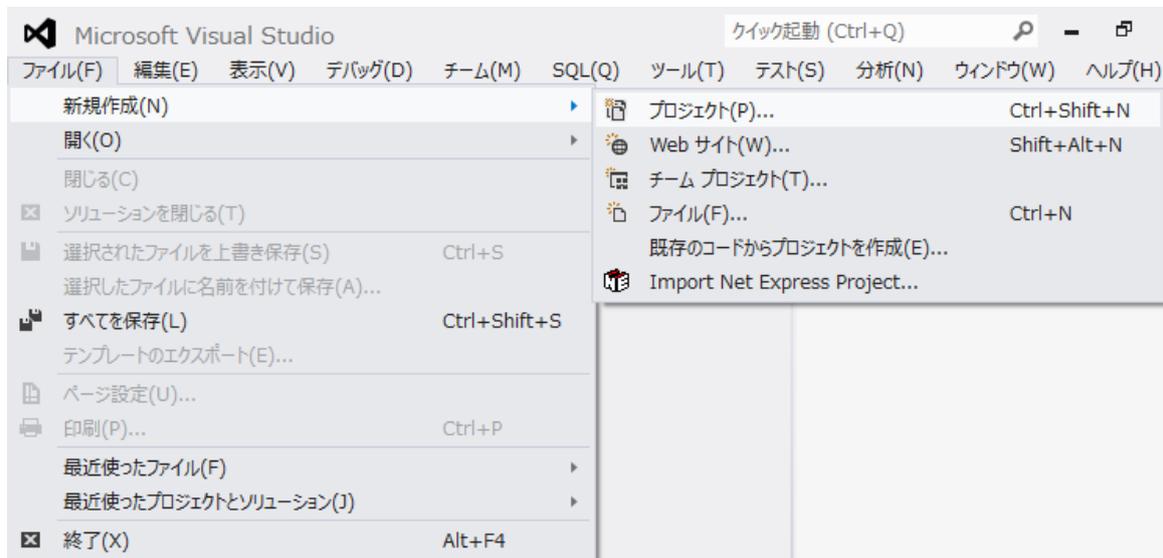
スタートアイコンから Enterprise Developer for Visual Studio 2012 を起動します。



## 5.4 ソリューションとプロジェクトの新規作成

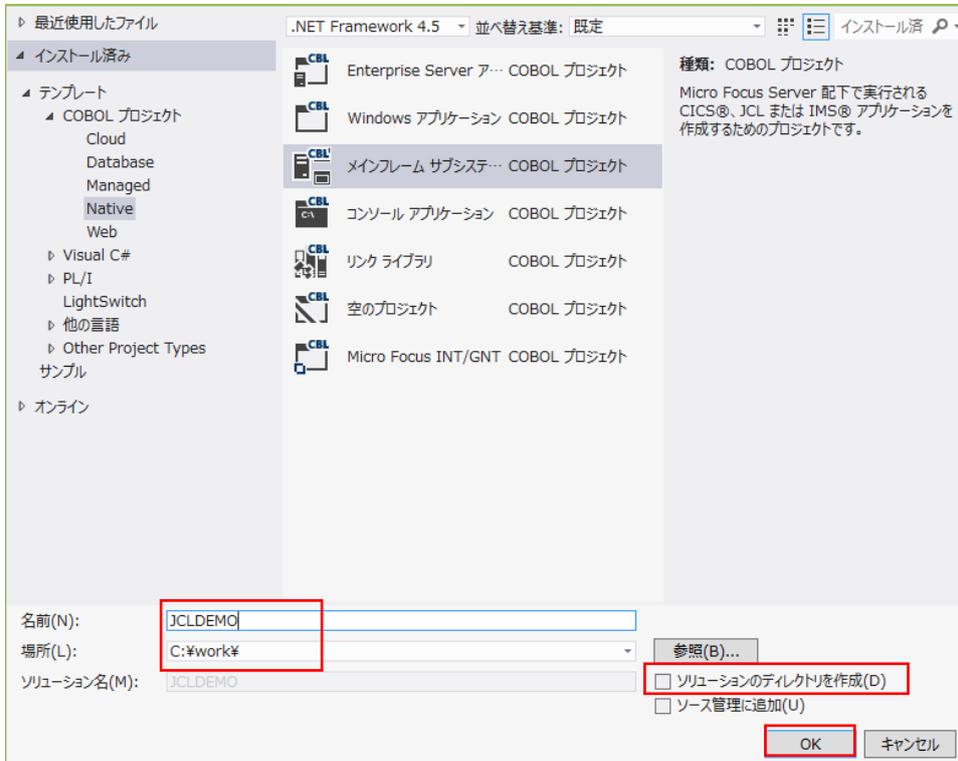
- 1) 新しいソリューションとプロジェクトを作成します。

プロダクションメニューの [ファイル]> [新規作成]> [プロジェクト] を選択します。

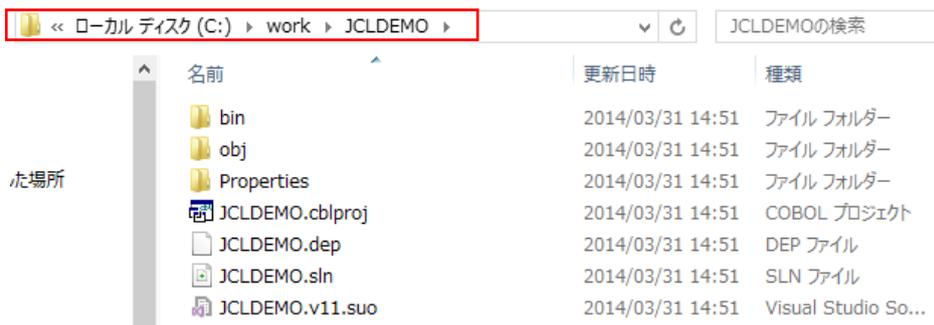


2) 新しいソリューションとプロジェクトを作成します。

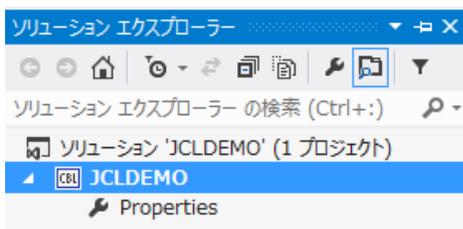
- ① ペインツリーの [テンプレート]>[COBOL プロジェクト]>[Native] 内から “メインフレーム サブシステム アプリケーション”を選択します。
- ② [名前] は “JCLDEMO” を入力します。
- ③ [場所] は 5.2 で作成した work フォルダを指定します。
- ④ [ソリューションのディレクトリを作成] のチェックを外して [OK] をクリックします。



⑤ work フォルダ直下に JCLDEMO ソリューションとプロジェクトが作成されました。

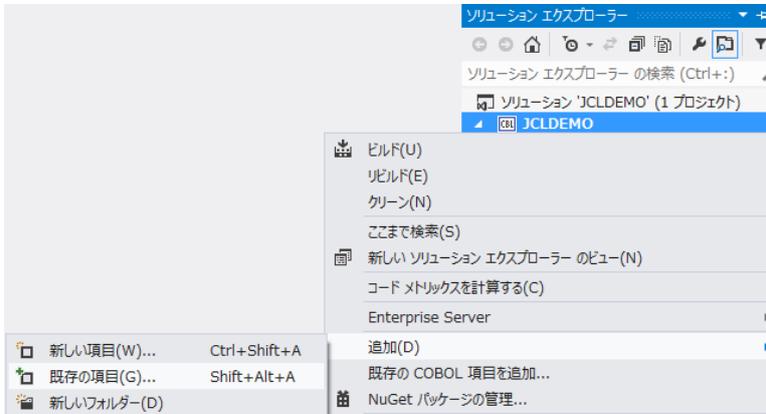


⑥ 作成したソリューションとプロジェクトが [ソリューション エクスプローラー] へ表示されます。

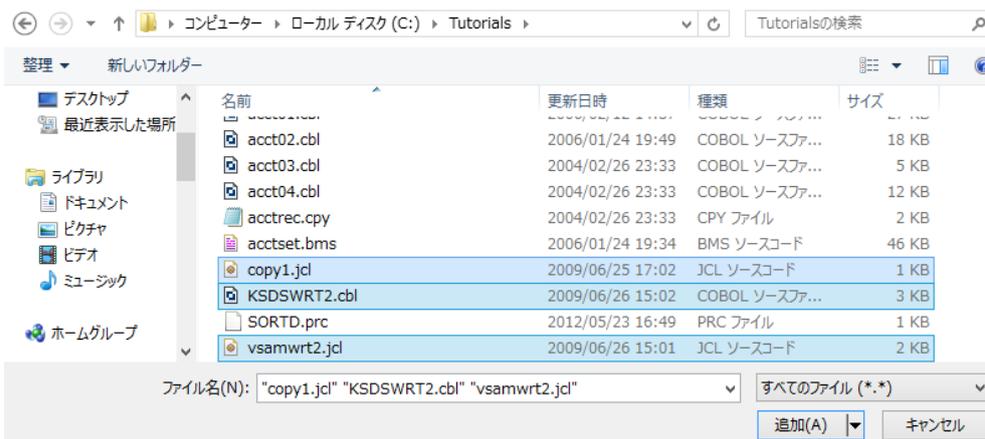


## 5.5 既存プログラムの取り込み

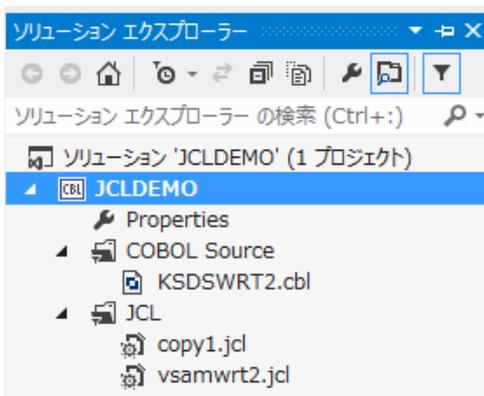
- 1) ソリューションエクスプローラでプロジェクトを右クリックし、[追加]>[既存の項目] を選択すると、ファイル選択のためのエクスプローラが表示されます。



- 2) C:\Tutorials 直下に存在する "copy1.jcl"、"KSDSWRT2.cbl"、"vsamwrt2.jcl" の3ファイルを選択して [追加] をクリックします。

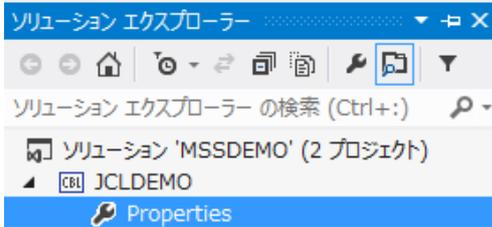


- 3) ソリューションエクスプローラへ追加されたファイルが表示されます。



## 5.6 ソースプロパティの設定とソースのビルド

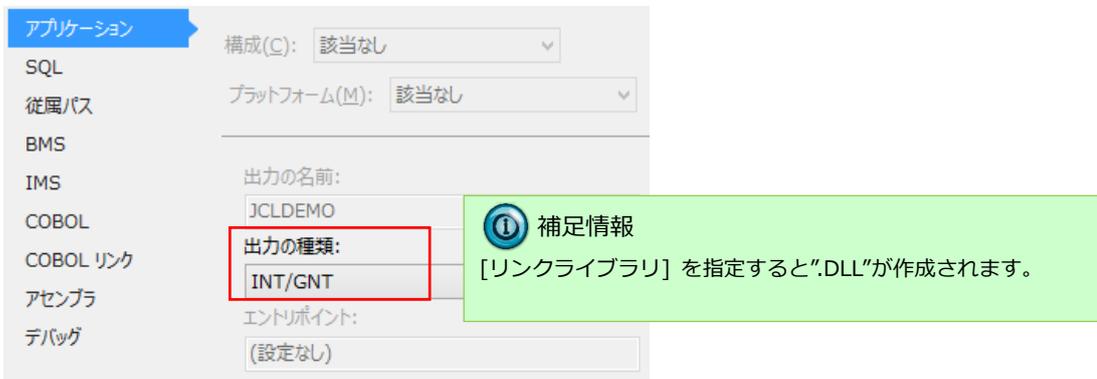
- 1) 作成したプロジェクトに必要なプロパティの設定を行います。ソリューション エクスプローラ内のプロジェクト [Properties] をダブルクリックし、設定ウィンドウを開きます。



- 2) プロジェクトのプロパティへ各種設定を行います。

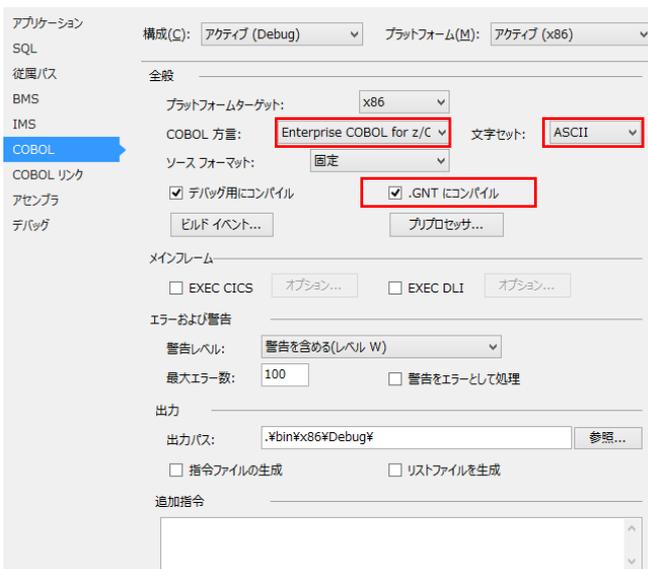
本チュートリアルでは IBM OS/VS COBOL の方言を使用するため下記設定を行います。

- ① ペインツリーの [アプリケーション]> [出力の種類] は [INT/GNT] オブジェクトタイプを選択します。



- ② ペインツリーの [COBOL] を選択し、以下の項目を設定します。

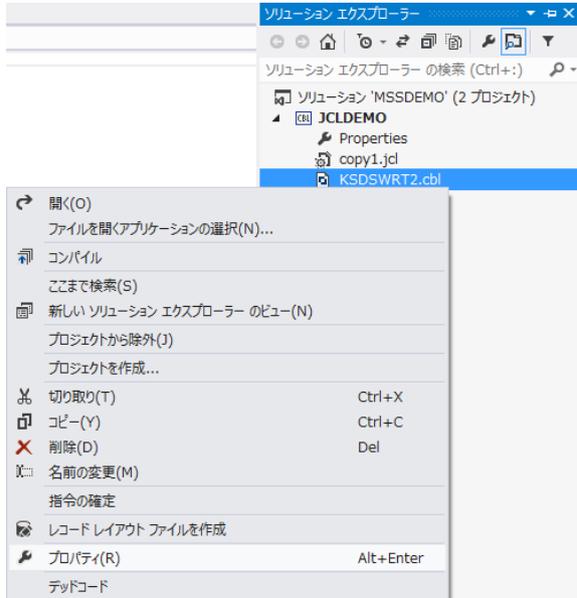
- ① [全般]> [COBOL 方言] は "Enterprise COBOL for z/OS" を選択します。
- ② [全般]> [文字セット] は "ASCII" を選択します。
- ③ [全般]> [.GNT にコンパイル] ヘチェックをして有効にします。



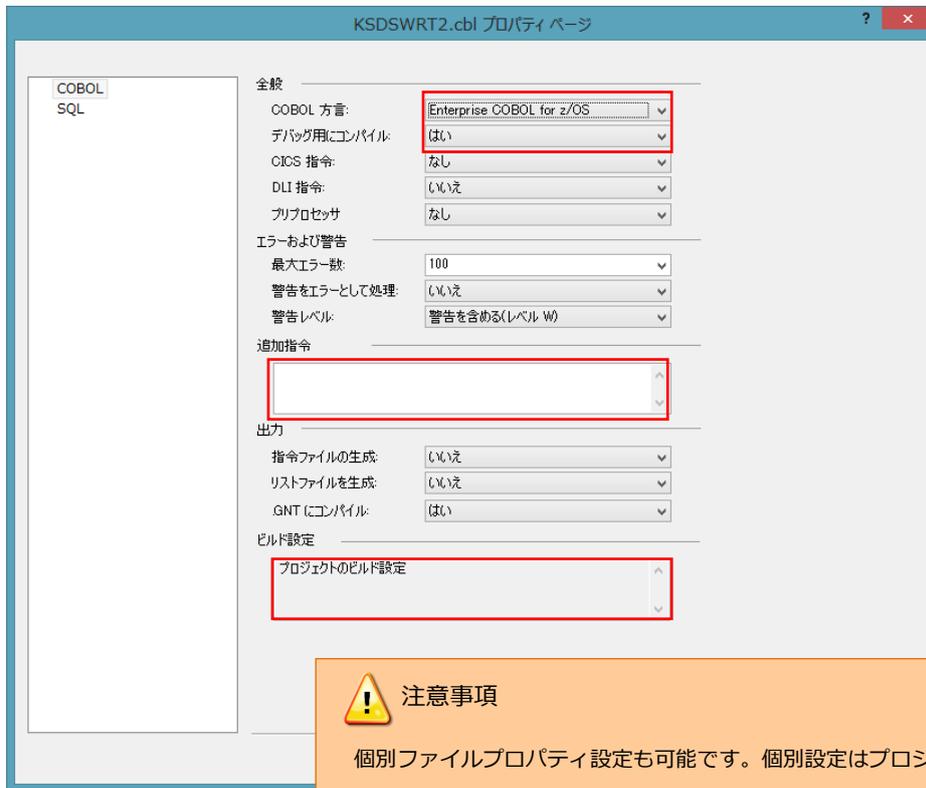
③ プロパティ設定ファイルを上書き保存します。



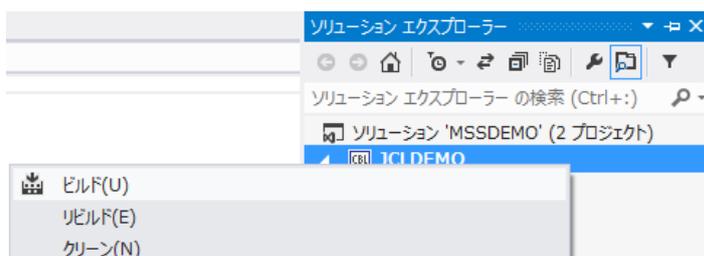
3) 個別ファイルのプロパティを確認するため、[KSDSWRT2.cbl] を右クリックして [プロパティ] を選択します。



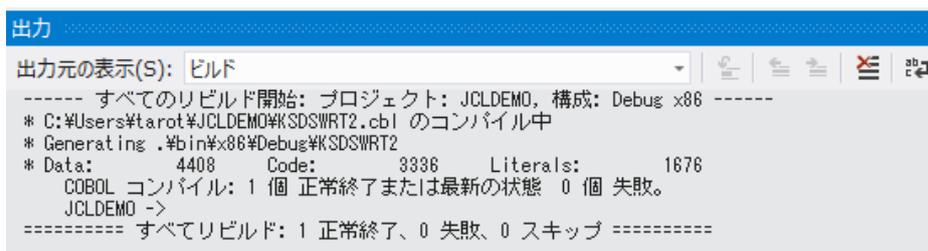
4) このチュートリアルでは個別に設定を行わないため、プロジェクト設定と相違がないか確認します。



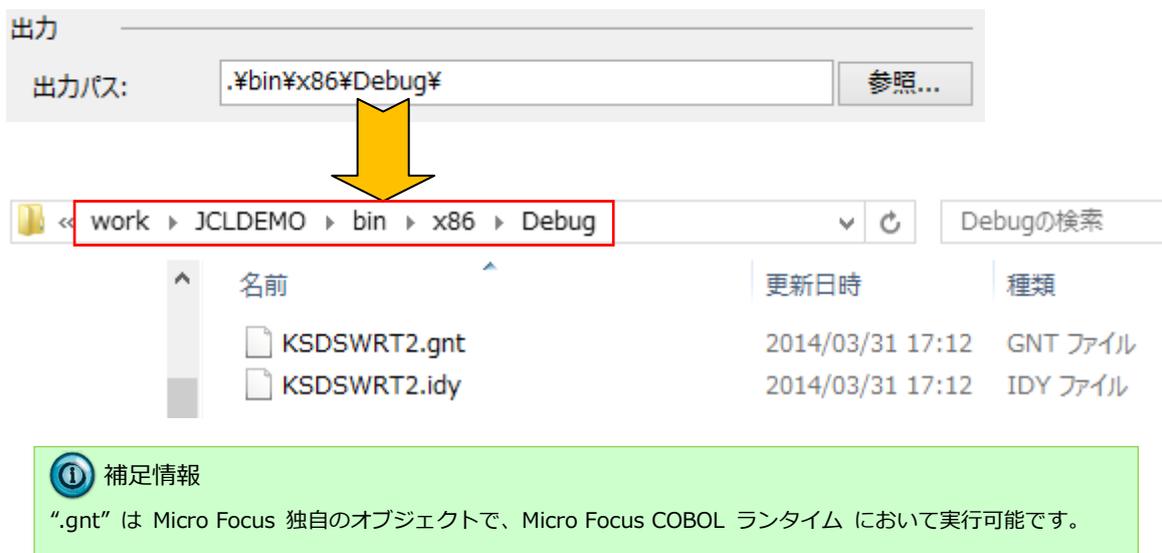
5) プロジェクトをビルドします。



6) コンパイル結果が出力ウィンドウへ表示されますので、正常終了を確認します。



7) 出力パスにはコンパイルされたオブジェクトが指定通りに作成されています。



## 5.7 Enterprise Server (リージョン) の構築

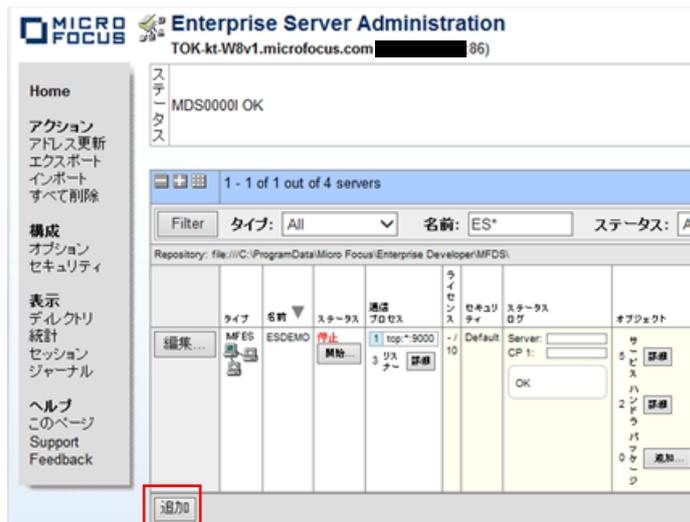
JCL を実行するために JES を使用可能としたリージョンを作成します。

- 1) 管理コンソールを表示します。

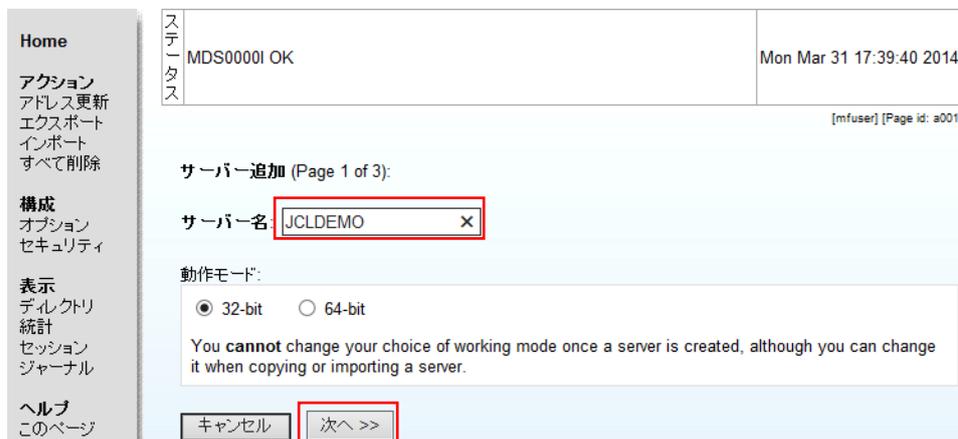
Enterprise Developer 内で開発用の Enterprise Server を操作するにはサーバーエクスプローラを使用します。サーバーエクスプローラが表示されていない場合は [表示]>[サーバー エクスプローラ] を選択して表示してください。 [Micro Focus Server] を右クリックして [管理] を選択して管理ウィンドウを開きます。



- 2) 画面下部の [追加] ボタンをクリックします。



- 3) 1/3 ページ画面に遷移し、サーバー名と動作モードを設定します。[サーバー名] は "JCLDEMO" を入力、[動作モード] は "32-bit" を選択し、[次へ] をクリックします。



- 4) 2/3 ページ画面ではサーバタイプを設定します。“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:

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.

- 5) 3/3 ページ画面では下記項目を設定して [追加] ボタンをクリックします。

- ① [ローカルコンソールを表示] : チェックをオンにします。
- ② [TN3270 リスナーの作成] : チェックをオフにします。

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"/>	システムアベンド時ダンプ: <input checked="" type="checkbox"/>
補助トレースアクティブ: <input type="checkbox"/>	<b>ローカルコンソールを表示: <input checked="" type="checkbox"/></b>
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

- 6) 管理ウィンドウにて JCLDEMO リージョンの [編集] ボタンをクリックします。

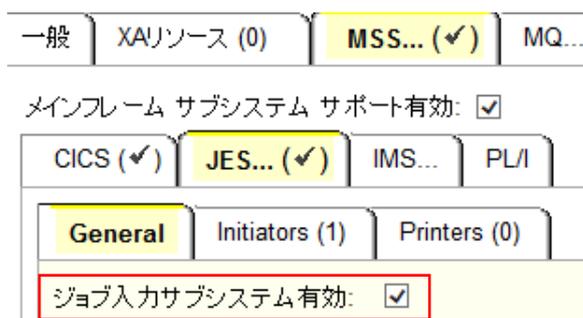


- 7) [動的デバッグを許可] チェックを [Apply] ボタンをクリックします。



- 8) JES サービスを有効にするための設定を行います。

[サーバー...] > [プロパティ...] > [MSS...] > [JES...] > [General] タブの [ジョブ入力サブシステム有効] をチェックして有効にします。



**!** 注意事項

JES を有効にしないとサービスが始動しないため、ESMAC 画面の左側メニューのリソース選択肢に [JES] が現れません。

9) JES の下記内容を設定します。

下記項目を入力して [Apply] ボタンをクリックします。これらのフィールドへ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 COBOL アプリケーションの探索先パスです。オブジェクトの出力パスを指定します。

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。

一般 XAリソース (0) MSS... (✓) MQ... スクリプト アクセ

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\JCLDEMO\bin\x86\Debug

システムカタログ:  
C:\work\JCLDEMO\DATAFILE\CATALOG.DAT

データセットの省略時ロケーション:  
C:\work\JCLDEMO\DATAFILE

システムプロシージャライブラリ:  
SYS1.PROCLIB

Fileshare Configuration Location:  
[Empty field]

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。

10) JES イニシエータを追加します。

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

一般 XAリソース (0) MSS... (✓) スク

Mainframe Subsystem Support enabled:

CICS (✓) JES... (✓) IMS... PL/I

一般 Initiators (0) Printers (0)

追加

11) JES イニシエータ入カウィンドウへ下記項目を入力し [追加] ボタンをクリックします。

**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。

12) 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。

名前	クラス	説明
INITABC	ABC	クラスABCのイニシエータ

13) JES タブで指定したパスの"DATAFILE"ディレクトリが存在しない場合は作成しておきます。

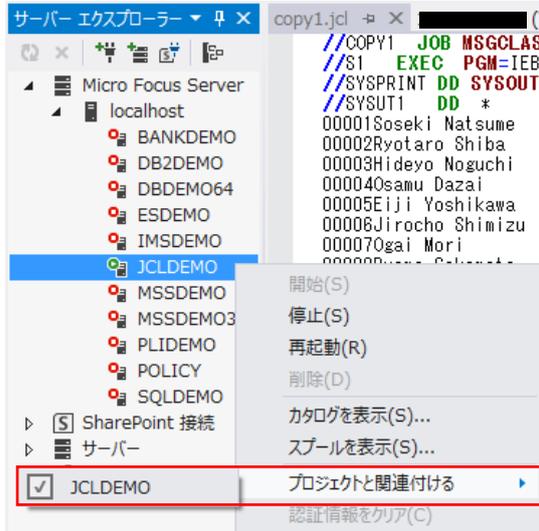
名前	更新日時	種類
bin	2014/03/31 14:51	ファイル フォルダ
DATAFILE	2014/04/01 9:19	ファイル フォルダ
obj	2014/03/31 14:51	ファイル フォルダ
Properties	2014/03/31 14:51	ファイル フォルダ

## 5.8 簡易 JCL の実行と結果の確認

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) JES リージョンとプロジェクトを関連付けます。

サーバーエクスプローラで [JCLDEMO] を右クリックし、[プロジェクトと関連付ける] から [JCLDEMO] を選択します。



- 2) JCLDEMO を起動します。

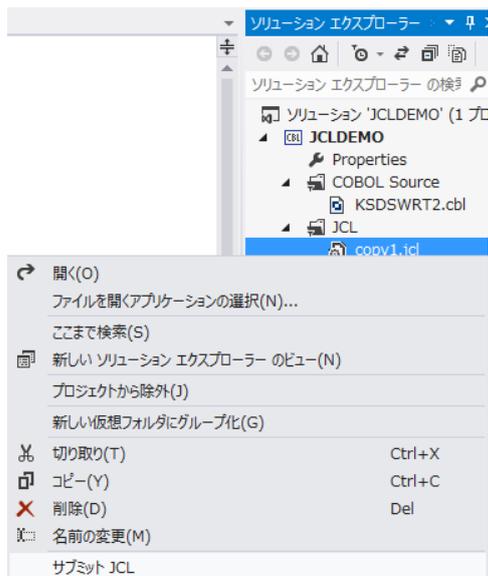


- 3) ソリューション エクスプローラで copy1.jcl をダブルクリックし、エディタでその内容を確認します。

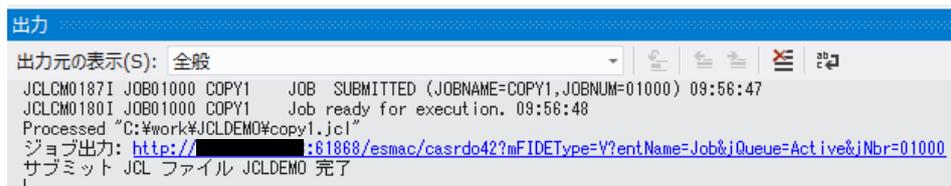


- 4) このジョブは IEBGENER ユーティリティを起動して JCL 内に書かれたインラインデータを SYSOUT に書き出していることが確認できます。

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



- 6) 出力ウィンドウ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ので、ジョブ出力リンクをクリックします。



- 7) 以下のようにスプールビューが開きます。

JOB01000		Name: COPY1	Status: Output Hold				
Release	Class: A	User: JESUSER	Priority: 00				
Update		COND: 00000					
JCLCM0188I JOB01000 COPY1 JOB STARTED 11:12:59 JCLCM0182I JOB01000 COPY1 JOB ENDED - COND CODE 0000 11:12:59							
	Status	Class	DD Name	Step	Nbr.	Proc Step	Records
Details	Hold	A	JESYSMSG		0		31
Details	Ready	A	SYSPRINT	S1	1		4
Details	Ready	A	SYSUT2	S1	1		10

- 8) JESYSMSG をクリックすると以下のように COPY1 ジョブのジョブログが表示されます。

```
Content-Type: text/plain

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*-*-* Micro Focus ESJCL ASCII JES2 Version ED22.00.00_022 *-*
*-*-* Copyright (C) 1997-2013 Micro Focus. All rights reserved. *-*
*-*-* Job: 01000 Name: COPY1 User: JESUSER Date: 03/06/14 Time: 11:12:58 *-*
*-*-* File: $TXRFDIR/T000000012.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 MSSDEMO Process 8984

11:12:59 JCLCM0188I JOB STARTED

11:12:59 JCLCM0190I STEP STARTED S1
11:12:59 JCLCM0199I Program MFJGENER is COBOL ASCII Big-Endian NOAMODE.
MFE2014.S0306.S111258.J01000.D00001.SYSPRINT SYSPRINT
C:¥WORK¥MSSDEMO¥DATAFILE¥MFE2014¥1258.J01000.D00001.SYSPRINT.DAT SPOOLED
MFE2014.S0306.S111258.J01000.D00002.SYSUT1 SYSUT1
C:¥WORK¥MSSDEMO¥DATAFILE¥MFE2014¥111258.J01000.D00002.SYSUT1.DAT DELETED
MFE2014.S0306.S111258.J01000.D00003.SYSUT2 SYSUT2
C:¥WORK¥MSSDEMO¥DATAFILE¥MFE2014¥111258.J01000.D00003.SYSUT2.DAT SPOOLED
DUMMY SYSIN
---> 11:12:59 JCLCM0191I STEP ENDED S1 - COND CODE 0000
---> 11:12:59 JCLCM0182I JOB ENDED - COND CODE 0000
```

- 9) スプールビューに戻り、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。

```
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
```

- 10) スプールビューに戻り、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。

```
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,Miyatomomura,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
```

以上で簡単なジョブの実行が確認できました。

## 5.9 プロシージャライブラリの作成

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パーティション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) Enterprise Server 管理ウィンドウから JCLDEMO の [詳細] ボタンをクリックします。

	タイプ	名前 ▼	ステータス	説明
編集...	MFES (MSS)	JCLDEMO	開始 詳細 停止...	Micro Focus Enterprise Server

- 2) 以下の [ES モニター&コントロール] ボタンをクリックします。

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...	コントロール	診断...	過去の統計					
ESモニター&コントロール...								
2 サービス実行プロセス								
Count	Type	PID	TR Count	State	Executing	Start Time	Local Time	Duration
1	Normal	39980	2	Idle				
2	Normal	40984	1	Idle				

- 3) ESMAC 画面が表示されますので、左下の [Resources] プルダウンで [JES] を選択します。

- 4) [Catalog] ボタンをクリックすると右側ペインに以下のような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。

- 5) 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。

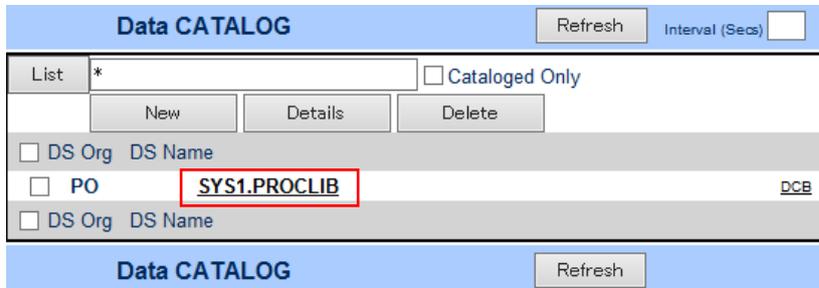
- 6) 以下のカタログエントリの新規作成ダイアログが現れます。以下のように入力して [Apply] をクリックします。「PO」はパーティションドデータセットであることを示します。また、この PO が拡張子 .PRC のテキストファイルをフォルダ配下に保持する動的 PDS であることを指定しています。

- 7) List で確認すると以下のようにカタログエントリ SYS1.PROCLIB が作成されました。

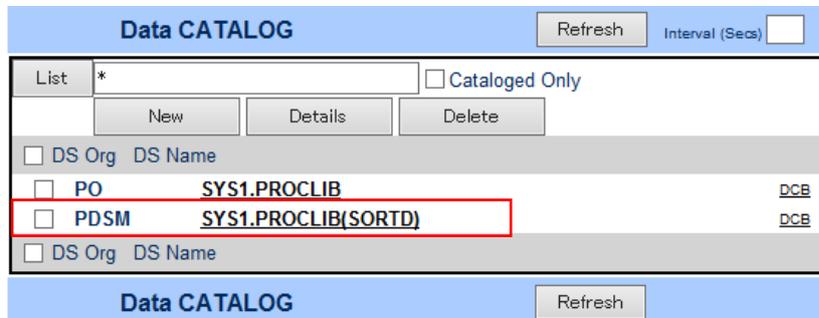
- 8) 指定した物理フォルダ C:¥work¥JCLDEMO¥DATAFILE¥SYS1.PROCLIB を Windows エクスプローラで作成し、その下に C:¥Tutorials にある SORTD.prc をコピーします。



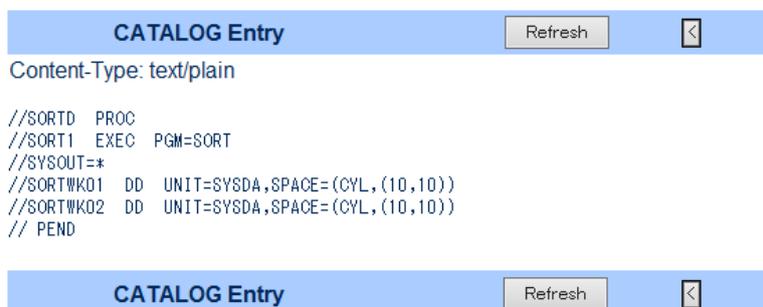
- 9) カタログビューで SYS1.PROCLIB をクリックします。



- 10) 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。



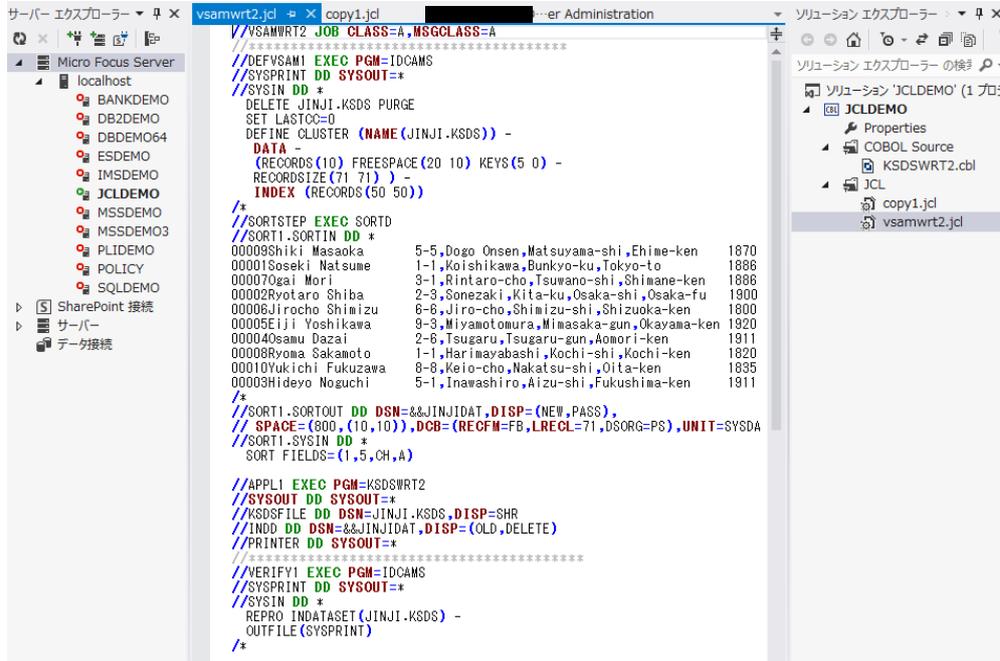
- 11) SYS1.PROCLIB(SORTD) をクリックすると以下のようにその内容を表示させることができます。



## 5.10 COBOL バッチプログラムの実行

COBOL バッチプログラム KSDSWRT2.cbl を含む、より実践的なジョブを実行します。

1) ソリューション エクスプローラから vsamwrt2.jcl をダブルクリックしエディタで開きます。



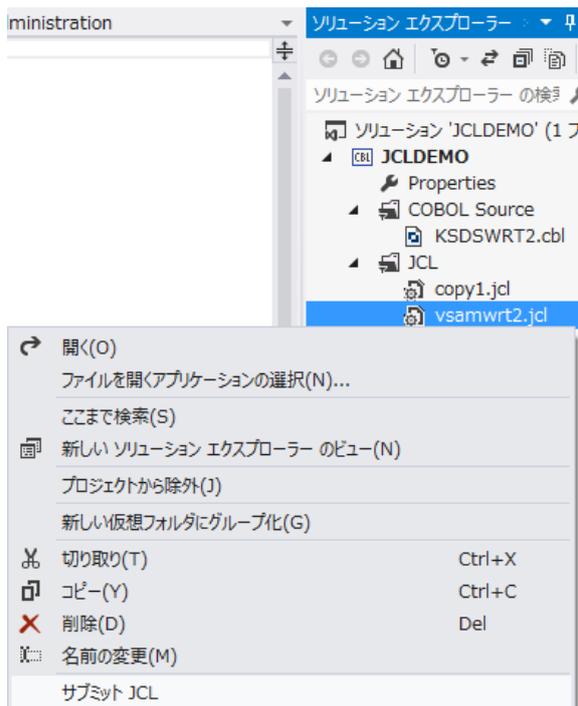
```
//VSAMWRT2 JOB CLASS=A,MSGCLASS=A
//*****
//DEFVSAM1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE JINJI.KSDS PURGE
SET LASTCC=0
DEFINE CLUSTER (NAME(JINJI.KSDS) -
  DATA -
  (RECORDS(10) FREESPACE(20 10) KEYS(5 0) -
  RECORDSIZE(71 71) ) -
  INDEX (RECORDS(50 50))
/*
//SORTSTEP EXEC SORTD
//SORT1.SORTIN DD *
0000$Shiki Masaoka      5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken    1870
0001$oseki Natsume     1-1,Koishikawa,Bunkyo-ku,Tokyo-to        1888
0002$ogai Mori         3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken   1888
0003$yotaro Shiba     2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu  1900
0004$jirocho Shimizu  6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken    1800
0005$Eiji Yoshikawa   9-3,Miyatomura,Mimasaka-gun,Okayama-ken 1920
0006$Osamu Dazai      2-6,Tsugaru,Tsugaru-gun,Aomori-ken      1911
0007$Ryoma Sakamoto   1-1,Harimayabashi,Kochi-shi,Kochi-ken    1820
0008$Yukichi Fukuzawa 8-8,Keio-cho,Nakatsu-shi,Oita-ken        1835
0009$Hideyo Noguchi   5-1,Inawashiro,Aizu-shi,Fukushima-ken    1911
/*
//SORT1.SORTOUT DD DSN=JINJIDAT,DISP=(NEW,PASS),
//SPACE=(800,(10,10)),DCB=(RECFM=FB,LRECL=71,DSORG=PS),UNIT=SYSDA
//SORT1.SYSIN DD *
SORT FIELDS=(1,5,CH,A)

//APPL1 EXEC PGM=KSDSWRT2
//SYSOUT DD SYSOUT=*
//KSDSFILE DD DSN=JINJI.KSDS,DISP=SHR
//INDD DD DSN=JINJIDAT,DISP=(OLD,DELETE)
//PRINTER DD SYSOUT=*
//*****
//VERIFY1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO INDATASET(JINJI.KSDS) -
  OUTFILE(SYSPRINT)
/*
```

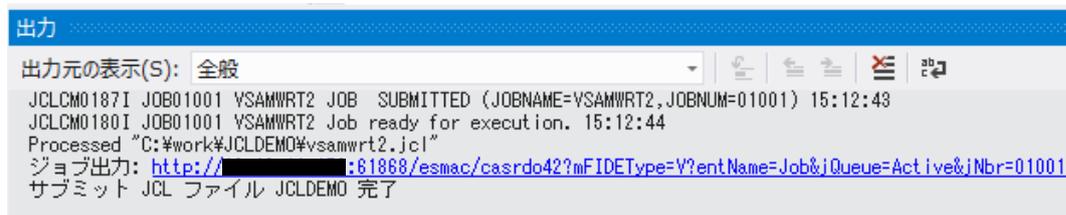
このジョブは4つのステップから構成され下記の様に連携されています。

- STEP1: DEFVSAM1  
IDCAMS を使用して KSDS クラスタ“JINJI.KSDS”を削除し再作成
- STEP2: SORTSTEP  
事前に登録したカタログ式プロシジャーの SORTD を使用して JINJI.KSDS ファイルへの書き込み用データをソート
- STEP3: APPL1  
アプリケーション KSDSWRT2 を呼び出し STEP2 でソートされたデータを使って STEP1 で定義された VSAM クラスタに書き込みを行う。同時に書き込まれたデータを DD=PRINTER へ出力
- STEP4: 書き込まれた内容の確認のため AMS の REPRO で内容を出力

- 2) ソリューション エクスプローラから vsamwrt2.jcl を右クリックして [サブミット JCL] を選択します。



- 3) 以下のように VSAMWRT2 ジョブが実行された旨のメッセージが出力ウィンドウに表示されますので、ジョブ出力のリンクをクリックします。





- 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。

```
MFE2014.S0401.S151243.J01001.D00002.SYSIN          SYSIN
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*S151243.J01001.D00002.SYSIN.DAT  DELETED
---> 15:12:45 JCLCM0191I STEP ENDED      DEFVSAM1 - COND CODE 0008

15:12:45 JCLCM0190I STEP STARTED  SORTSTEP.SORT1
15:12:45 JCLCM0199I Program MFJSORT is COBOL ANS85 ASCII Big-Endian  NOAMODE.
MFE2014.S0401.S151243.J01001.D00003.SORTIN          SORTIN
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*151243.J01001.D00003.SORTIN.DAT  DELETED
MFE2014.S0401.S151243.J01001.ANDAND.JINJIDAT          SORTOUT
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*1243.J01001.ANDAND.JINJIDAT.DAT  PASSED
MFE2014.S0401.S151243.J01001.D00004.SYSIN          SYSIN
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*S151243.J01001.D00004.SYSIN.DAT  DELETED
MFE2014.S0401.S151243.J01001.D00005.SYSOUT          SYSOUT
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*151243.J01001.D00005.SYSOUT.DAT  SPOOLED
---> 15:12:45 JCLCM0191I STEP ENDED  SORTSTEP.SORT1 - COND CODE 0000

15:12:45 JCLCM0190I STEP STARTED  APPL1
15:12:45 JCLCM0220I Job restart will not be permitted at this step.
15:12:45 JCLCM0199I Program KSDSWRT2 is COBOL VSC2 ASCII Big-Endian  NOAMODE.
MFE2014.S0401.S151243.J01001.D00006.SYSOUT          SYSOUT
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*151243.J01001.D00006.SYSOUT.DAT  SPOOLED
JINJI.KSDS                                           KSDSFILE
C:¥WORK¥JCLDEMO¥DATAFILE¥JINJI.KSDS.DAT             RETAINED
MFE2014.S0401.S151243.J01001.ANDAND.JINJIDAT          INDD
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*1243.J01001.ANDAND.JINJIDAT.DAT  DELETED
MFE2014.S0401.S151243.J01001.D00007.PRINTER          PRINTER
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*151243.J01001.D00007.PRINTER.DAT  SPOOLED
---> 15:12:45 JCLCM0191I STEP ENDED  APPL1 - COND CODE 0000

15:12:45 JCLCM0190I STEP STARTED  VERIFY1
15:12:45 JCLCM0199I Program MFJAMS is COBOL          ASCII Big-Endian  NOAMODE.
MFE2014.S0401.S151243.J01001.D00008.SYSPRINT          SYSPRINT
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*1243.J01001.D00008.SYSPRINT.DAT  SPOOLED
MFE2014.S0401.S151243.J01001.D00009.SYSIN          SYSIN
C:¥WORK¥JCLDEMO¥DATAFILE¥MFE2014*S151243.J01001.D00009.SYSIN.DAT  DELETED
---> 15:12:45 JCLCM0191I STEP ENDED  VERIFY1 - COND CODE 0000
```

- 7) 以下、順番に内容を確認します。最初は IDCAMS の SYSPRINT です。

Content-Type: text/plain

```
Micro Focus MFJAMS Utility Version ED22.00.00_02
Copyright (C) 1997-2013 Micro Focus. All rights reserved.
```

```
DELETE JINJI.KSDS PURGE
```

```
JCLAM0114I(00) - ENTRYNAME DELETED [JINJI.KSDS]
```

```
SET LASTCC=0
```

```
JCLAM0140I(00) - LASTCC set to 0.
```

```
DEFINE CLUSTER (NAME(JINJI.KSDS)) -
  DATA -
  (RECORDS(10) FREESPACE(20 10) KEYS(5 0) -
  RECORDSIZE(71 71) ) -
  INDEX (RECORDS(50 50))
```

```
JCLAM0113I(00) - ENTRYNAME DEFINED [JINJI.KSDS]
```

- 8) 次は SORTD プロシージャ経由で起動される SORT ステップの SYSOUT です。

```
Content-Type: text/plain

Micro Focus MFJSORT ユーティリティ 3.0.00

SORT FIELDS=(1,5,CH,A)

SORT2041: ***** ソート結果 *****
SORT2051: INPUT   ファイル 'SORTIN'
              入力レコード              10 件
              使用レコード              10 件
SORT2061: OUTPUT  ファイル 'SORTOUT'
              使用レコード              10 件
              出力レコード              10 件
SORT3991: Micro Focus MFJSORT ユーティリティ終了
```

- 9) 次は COBOL プログラム KSDSWRT2 の DISPLAY 文による出力です。

```
Content-Type: text/plain

**END OF JOB**
```

- 10) 次は COBOL プログラム KSDSWRT2 の PRINTER 出力です。

```
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,Miyatomomura,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
```

- 11) 次は最後の IDCAMS のステップの SYSPRINT です。

```
Content-Type: text/plain

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

REPRO INDATASET(JINJI.KSDS) -
OUTFILE(SYSPRINT)

LISTING OF DATASET - JINJI.KSDS

KEY OF RECORD - 00001
00001Soseki Natsume      1-1,Koishikawa,Bunkyo-ku,Tokyo-to      1886

KEY OF RECORD - 00002
00002Ryotaro Shiba      2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 1900

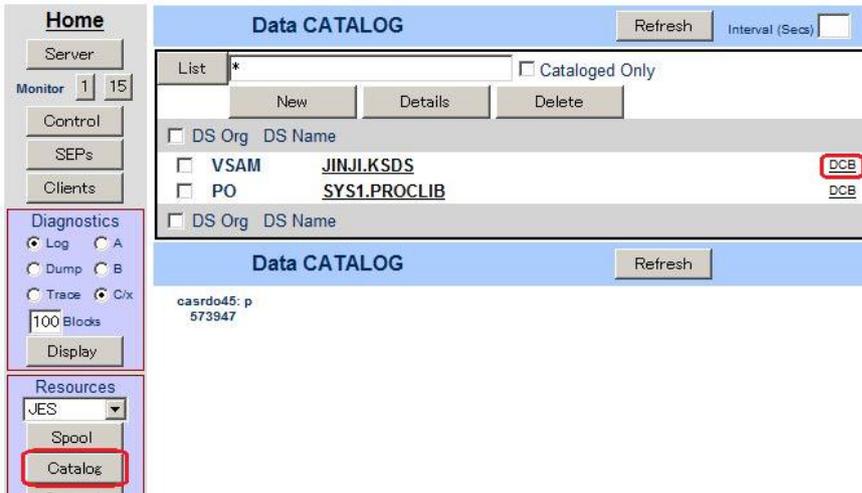
KEY OF RECORD - 00003
00003Hideyo Noguchi    5-1,Inawashiro,Aizu-shi,Fukushima-ken  1911

KEY OF RECORD - 00004
00004Osamu Dazai       2-6,Tsugaru,Tsugaru-gun,Aomori-ken     1911

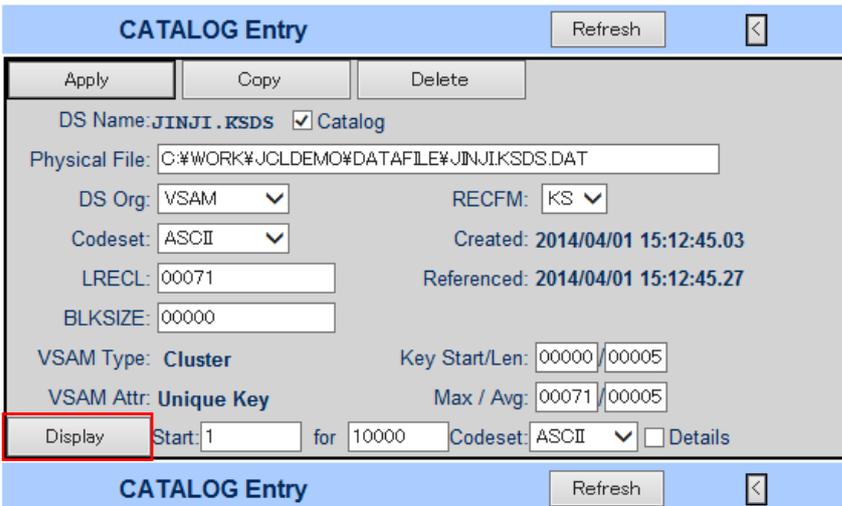
KEY OF RECORD - 00005
00005Eiji Yoshikawa    9-3,Miyatomomura,Mimasaka-gun,Okayama-ken 1920

KEY OF RECORD - 00006
00006Jirocho Shimizu   6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken   1800
```

12) 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。以下のように ESMAC 画面でカタログビューを開き、[List] ボタンをクリックします。VSAM ファイル JINJI.KSDS がカタログされていることがわかります。



13) 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。



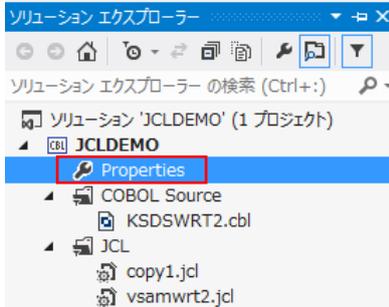
14) [Display]ボタンをクリックすると以下のようにデータセットの内容が表示されます。



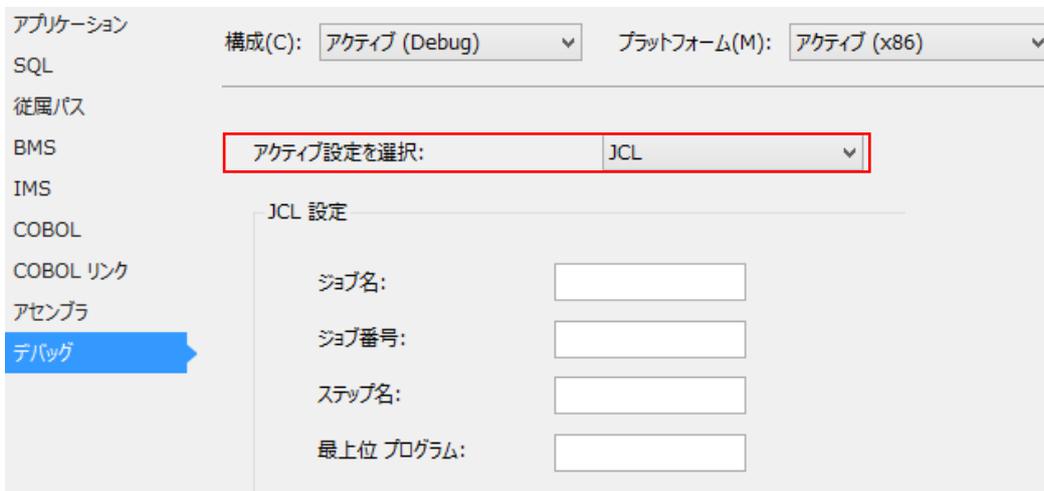
## 5.11 COBOL バッチプログラムのデバッグ

JES 配下で実行される COBOL プログラムをステップ実行でデバッグすることができます。

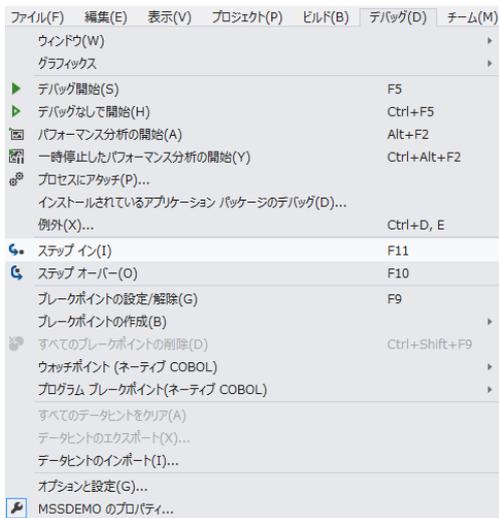
- 1) ソリューション エクスプローラからプロジェクトの [Properties] をダブルクリックします。



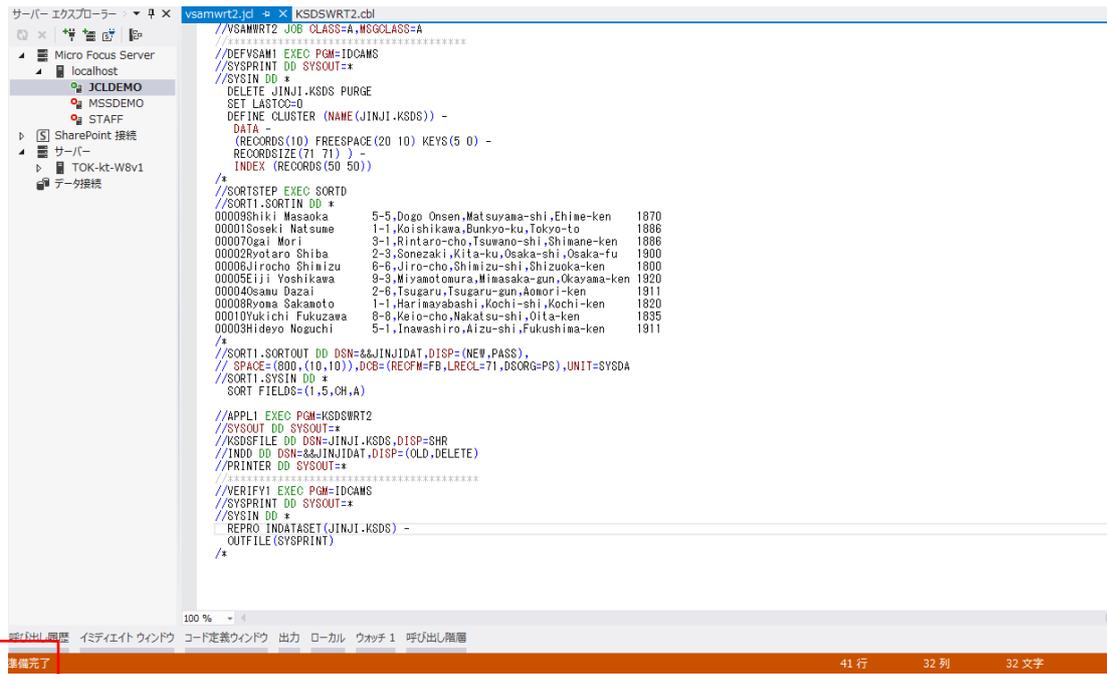
- 2) ペインツリーで [デバッグ] を選択し、[アクティブ設定を選択] に “JCL” を選択して保存します。



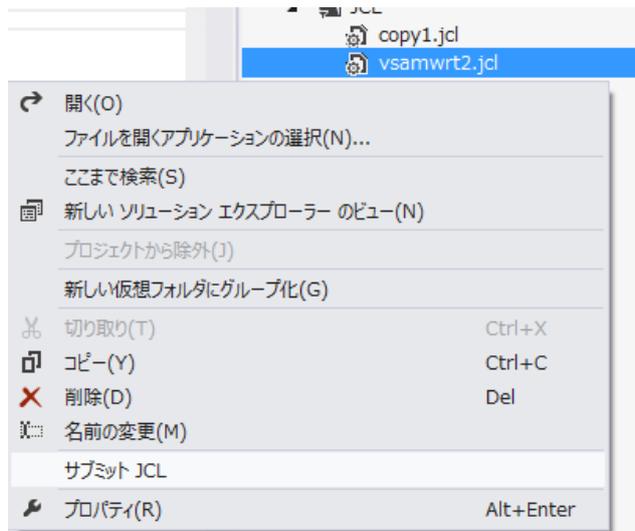
- 3) プルダウンメニューの [デバッグ]>[ステップイン] を実行します。



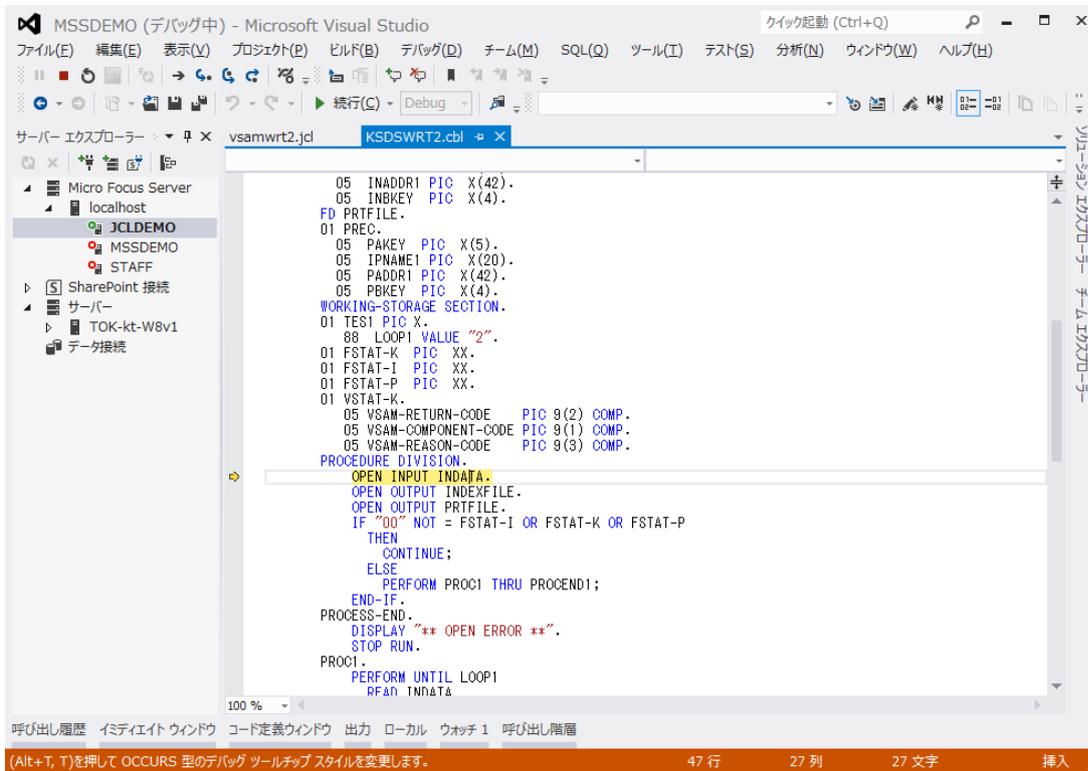
4) 実行待機状態で起動します。



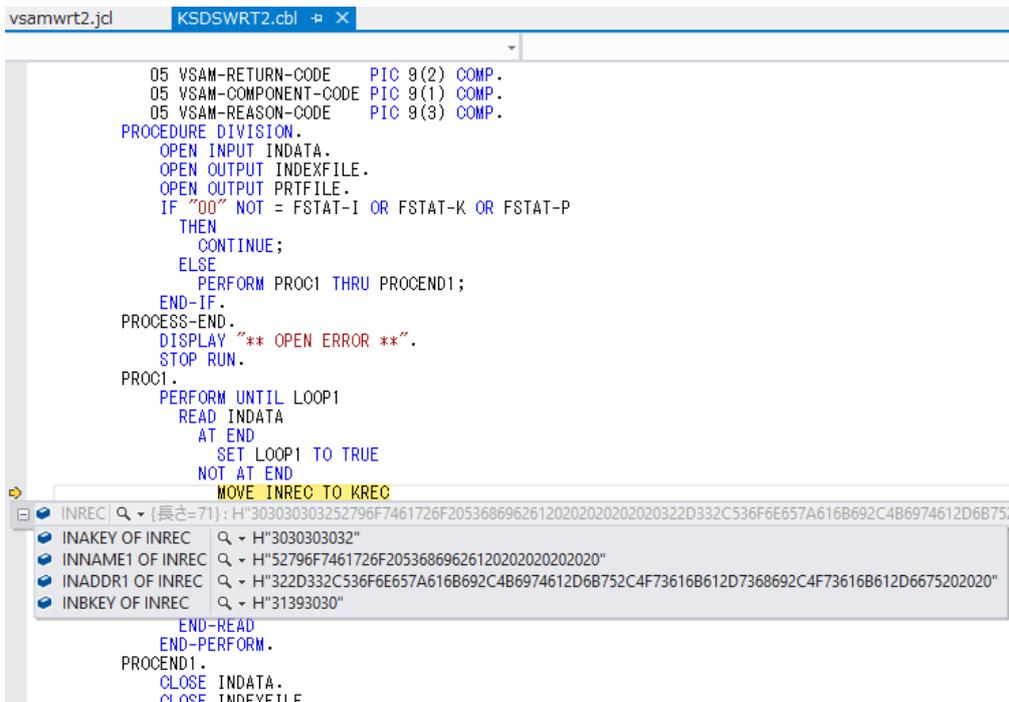
5) ソリューション エクスプローラから vsamwr2.jcl を右クリックし [サブミット JCL] を選択します。



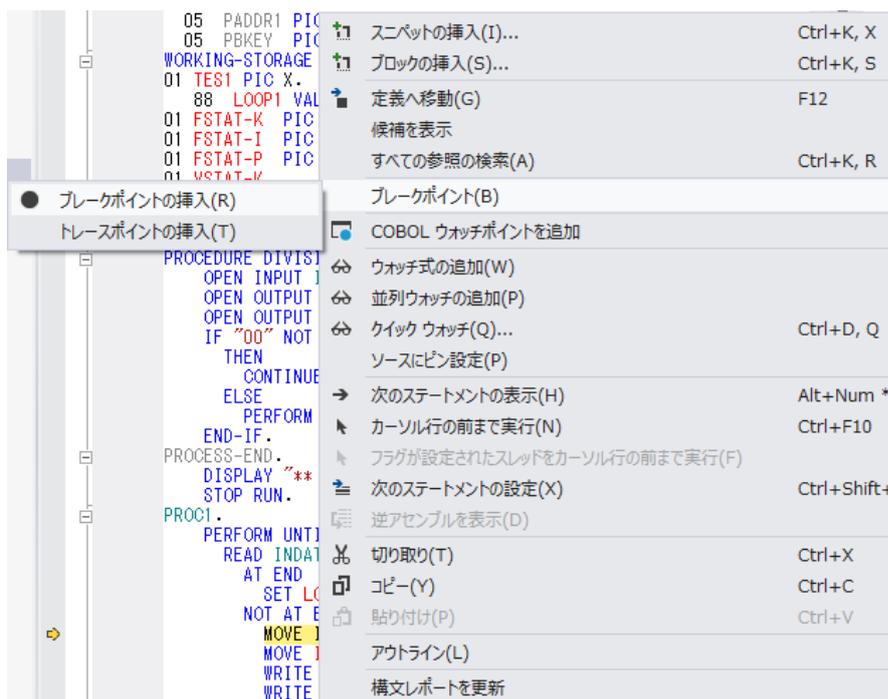
- 6) 以下のようにデバッグセッションが開始し、COBOL プログラム KSDSWRT2 の手続き部の先頭でハイライトされています。F11 キーでステップ実行が可能となります。また、ウォッチ式の追加によりオブジェクト内容の確認も可能です。



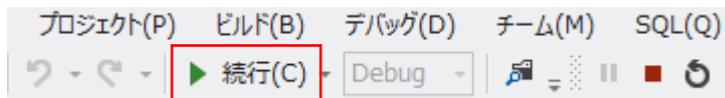
- 7) INREC オブジェクトへフォーカスをあてると、以下のようにデータ項目内容が確認できます。



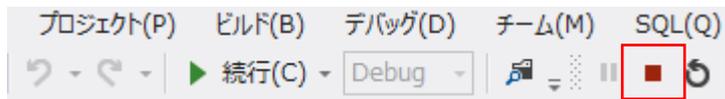
8) ソース行を右クリックすると、以下のようにブレークポイントの挿入または削除が可能です。



9) [続行] アイコンをクリックするとブレークポイントまで処理が進みます。



10) デバッグを終了します。



**① 補足情報**  
 リージョン運用状況の [サーバ]> [コントロール(構成)]> [ES モニター & コントロール] ウィンドウの左下 [Dyn-Debug] ボタンで表示されるウィンドウで、待機中のデバッグを確認することができます。

11) JCLDEMO リージョンを終了します。

以上で JCL チュートリアルを終了します。