

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 JCL チュートリアル

5.1 チュートリアルの準備

本チュートリアルを実行する前に「4. Enterprise Developer CICS チュートリアル」を終了している必要があります。ここで使用する例題プログラムは、CICS チュートリアルとともに Tutorials.zip に圧縮されており、すでに C:\¥Tutorials の下に解凍されており、MSSDEMO プロジェクトに追加されています。

5.2 JES リージョンの構成

本チュートリアルでは例題 JCL を Enterprise Server にサブミットして実行します。ここでは CICS チュートリアルで使用した MSSDEMO リージョンをそのまま使用しますので、JES を使用可能とするための準備を行います。

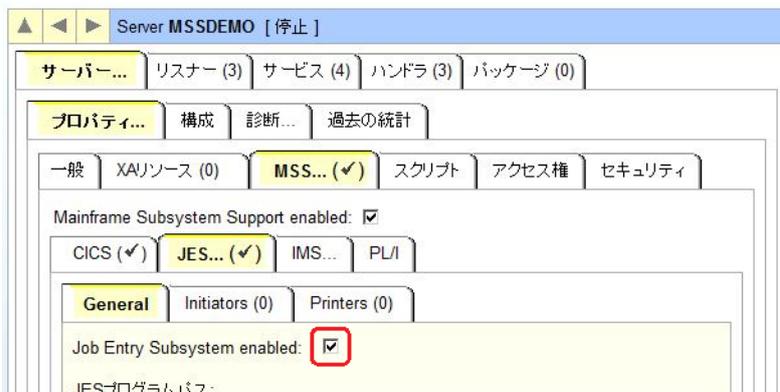
- 1) MSSDEMO が開始状態の場合にはいったん停止します。
- 2) CICS チュートリアルで作成された MSSDEMO プロジェクトのフォルダ C:\¥work¥MSSDEMO¥MSSDEMO 下に DATAFILE フォルダを作成しておきます。



- 3) Enterprise Server Admin にて MSSDEMO の [編集] ボタンをクリックします。



- 4) [サーバー...] > [プロパティ...] > [MSS...] > [JES...] > [General] タブを開きます。以下のように [Job Entry Subsystem enabled] のチェックをオンにします。



- 5) 以下のように各フィールドに入力し、[Apply] ボタンをクリックします。

ここで、[JES プログラムパス] は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。

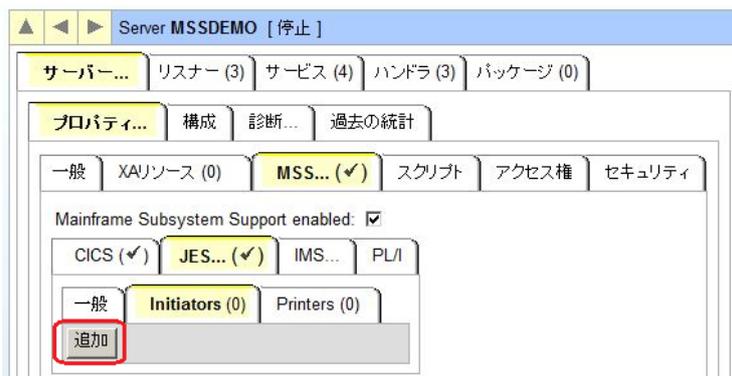
[データセットの省略時ロケーション] はジョブの実行とともに生成されるスプールデータやカタログされるデータセットの置き場所です。

[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。

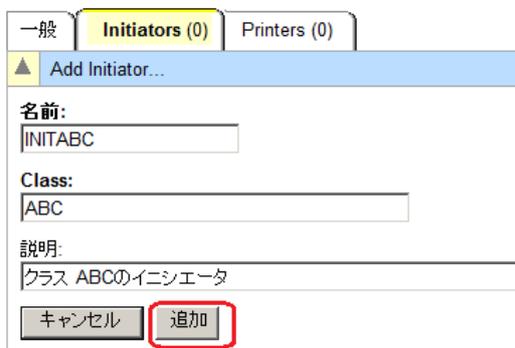
なお、これらのフィールドでは改行を入れないように注意してください。



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



7) 以下のように入力し [追加] ボタンをクリックします。



8) 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。

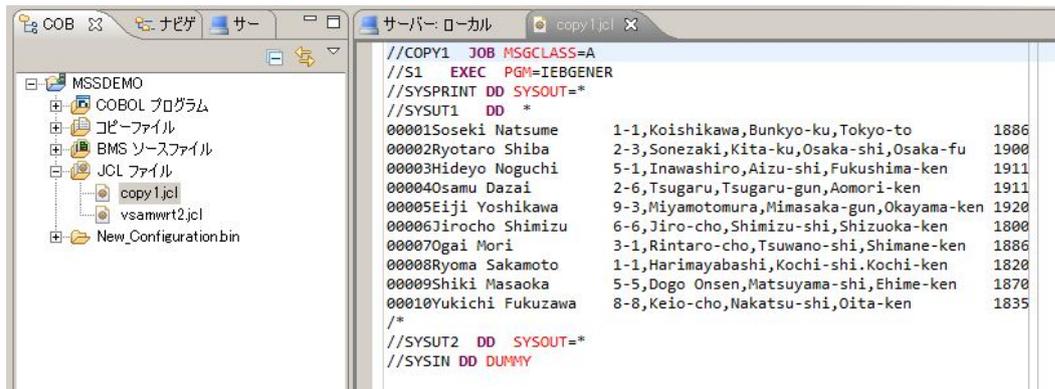


以上で JES リージョンの設定は完了しました。

5.3 簡単な JCL の実行

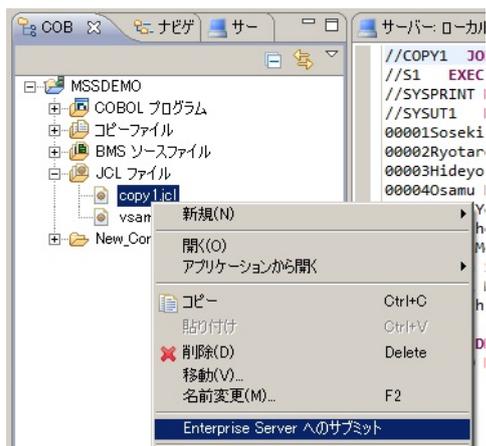
まずもっとも簡単な JCL をこの JES リージョンにサブミットして実行してみます。

- 1) MSSDEMO を再起動します。
- 2) プロジェクト内の COBOL 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。

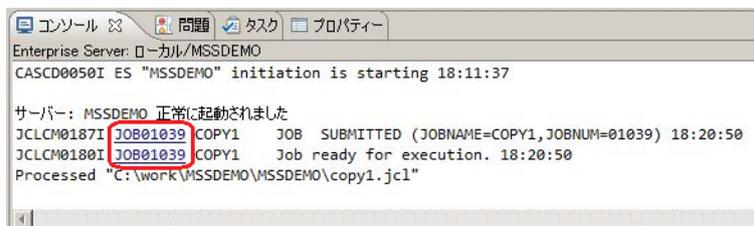


3) このジョブは IEBGENER ユーティリティを起動して JCL 内に書かれたインラインデータを SYSOUT に書き出しているだけのものです。

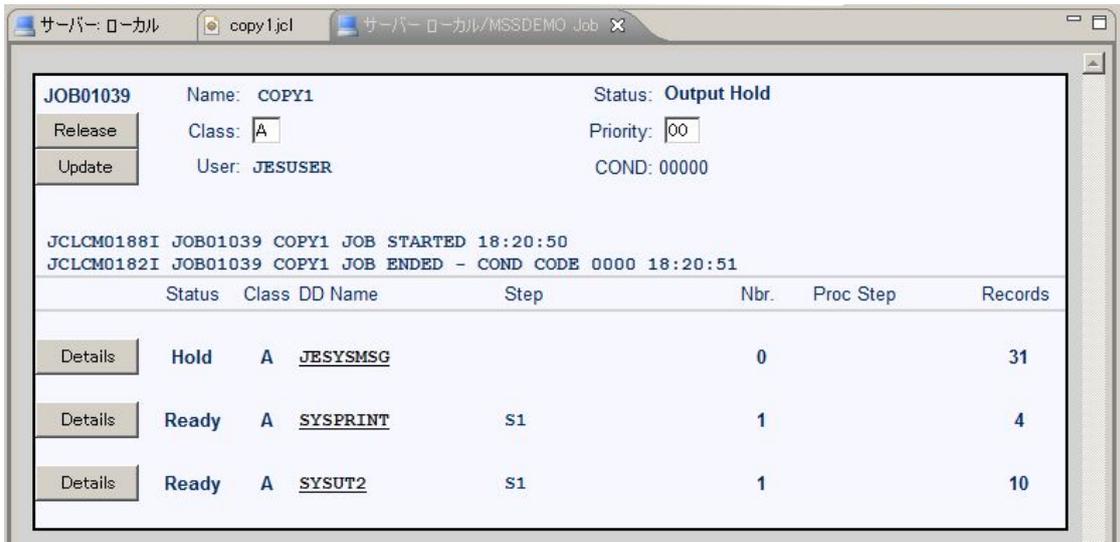
4) COBOL エクスプローラ内で copy1.jcl を右クリックし「Enterprise Server へのサブミット」を選択します。



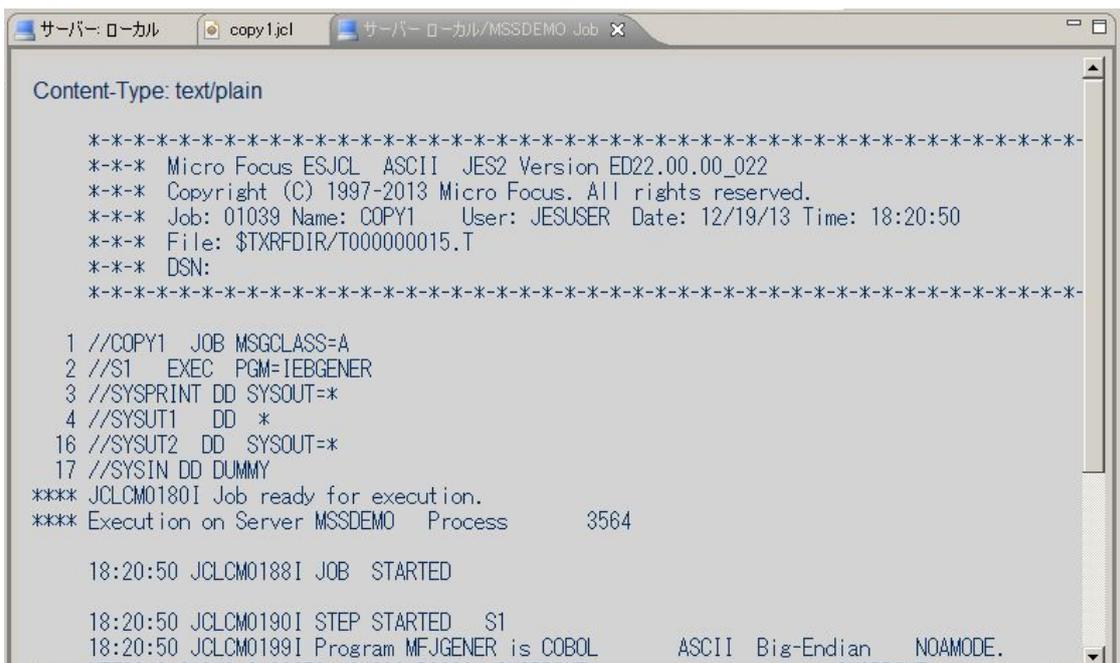
5) コンソールにジョブがサブミットされたことを示すメッセージが表示されますので、ジョブ番号をクリックします。



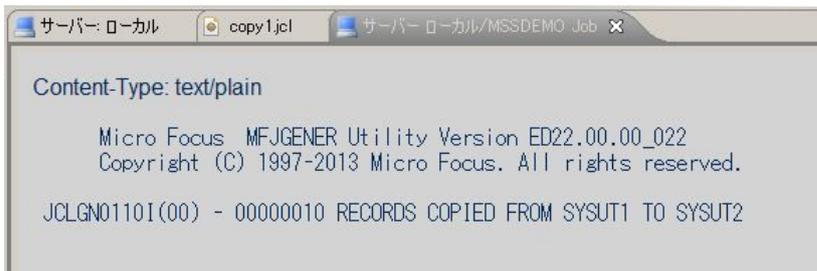
6) 以下のようにスプールビューが開きます。



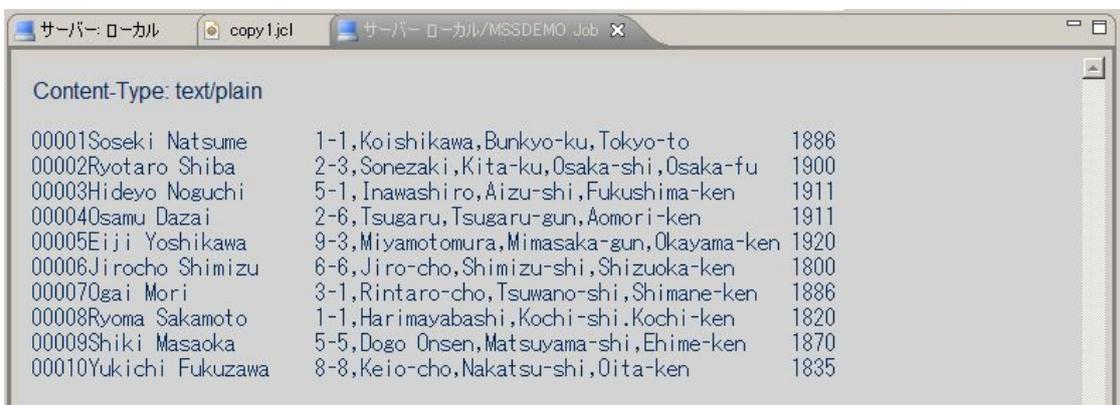
7) JESYSMSG をクリックすると以下のように COPY1 ジョブのジョブログが表示されます。



8) 右クリックし[前に戻る]を選択し、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。



9) 前画面に戻り、SYSUT2 をクリックします。以下のように出力されたスプールの内容が表示されます。

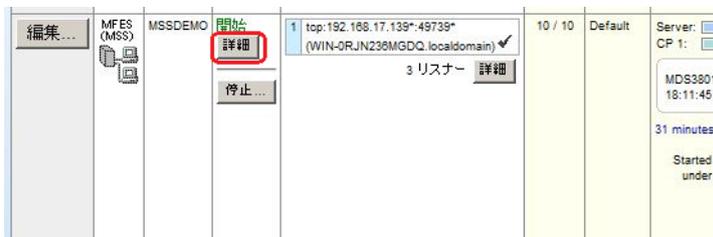


以上で簡単なジョブの実行が確認できました。

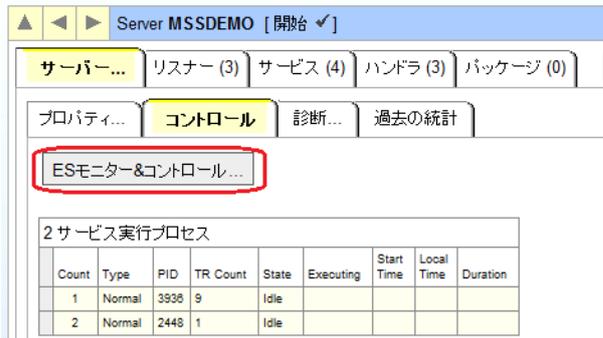
5.4 プロシージャライブラリの作成

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 管理コンソールで MSSDEMO の [詳細] をクリックします。



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。



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。

Server: **MSSDEMO**
 Host: **192.168.17.139**
 Address: 192.168.17.139

User ID: **mfuser**
 Group: **mfuser**
 Time: 12/19/2013 18:52:30

Server Information Refresh Interval (Secs)

Times: Start: 12/19/2013 - 18:11:43 Number: 3
 Elapsed: 0 - 0:40:47

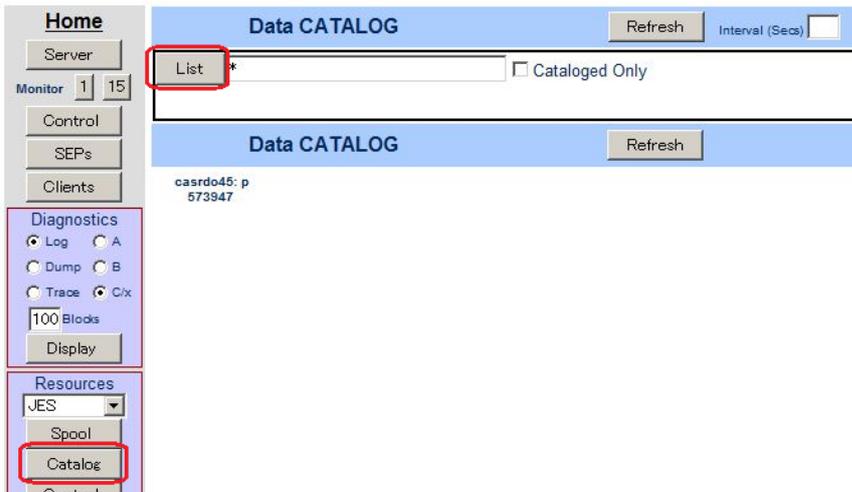
Counts: Transactions: 237 Dumps: 0
 Per Hour: 348 Trace Blocks: 0
 Active Clients: 1 Limit: 10

Sizes: Max Tasks: 2 (admin:1) HTTP Out: 63 k segments
 Diagnostics Size: 4190208 k ID Timeout: 0 (minutes)
 Shared Memory: 512 pages (4k) SM Total: 2,048 k
 SM Cushion: 131 k SM Free: 1,809 k (3)

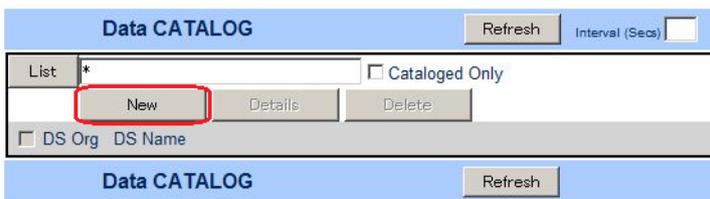
States: Working mode: 32 Perf Enabled: No
 Force Phase In: Yes Dump: A
 Dump on ABEND: Sys:Yes Tran:No Trace: A
 Active trace: api, kcp

Startup: SIT: DBCS PLT-PI:
 SYSID: \$IVP PLT-SD:
 JES/JCL: Yes IMS: No
 Protected TS/TD: Warm/Warm unP. TS/TD: Warm/Warm
 Fileshare User: ES_MSSDEMO0000003812

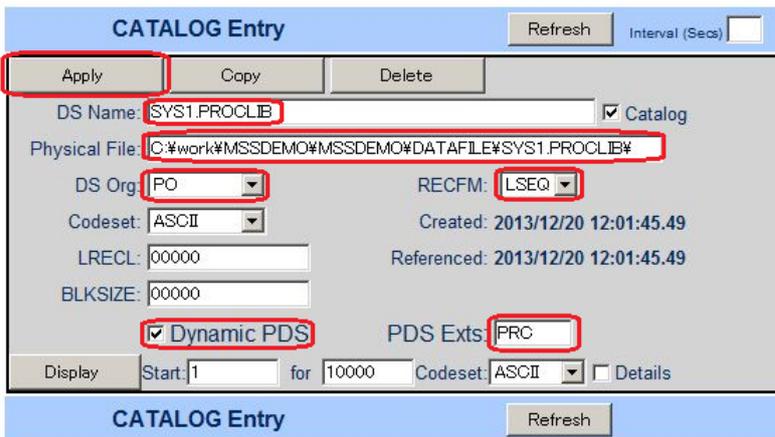
4) [Catalog] ボタンをクリックすると右側ペインに以下のようなカタログビューが表示されます。ここで [List]ボタンをクリックします。



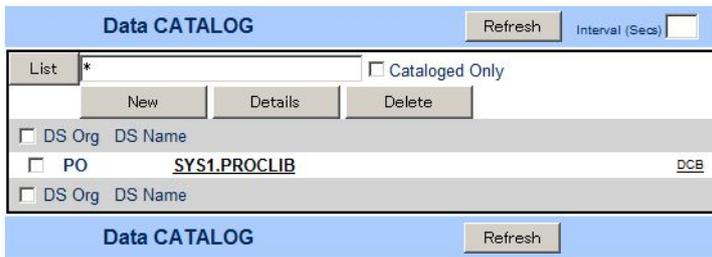
5) 現在カタログされているデータセットは何もありません。そこで [New] ボタンをクリックします。



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。



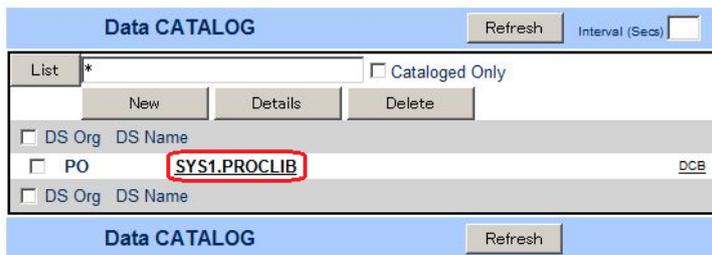
7) 以下のようにカタログエントリ SYS1.PROCLIB が作成されました。



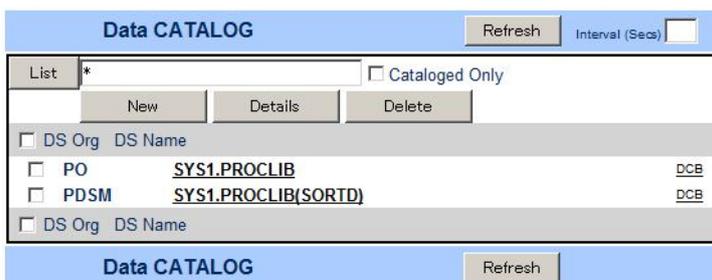
- 8) 指定した物理フォルダ C:\work\MSSDEMO\MSSDEMO\DATAFILE\SYS1.PROCLIB を Windows エクスプローラで作成し、その下に C:\Tutorials にある SORTD.prc をコピーします。



- 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。

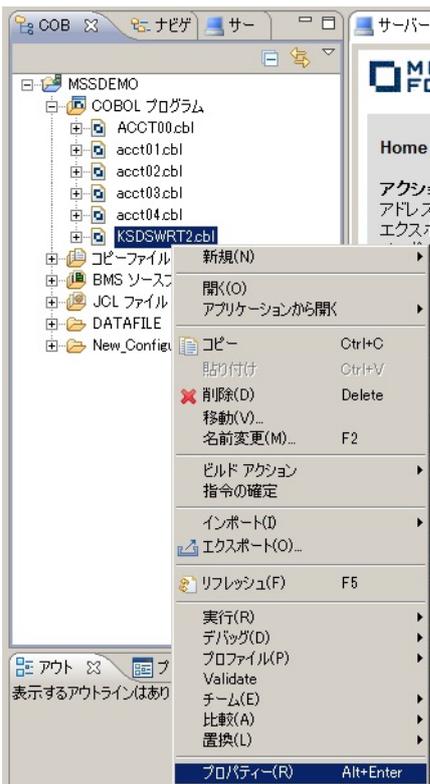
```
CATALOG Entry Refresh <
Content-Type: text/plain

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

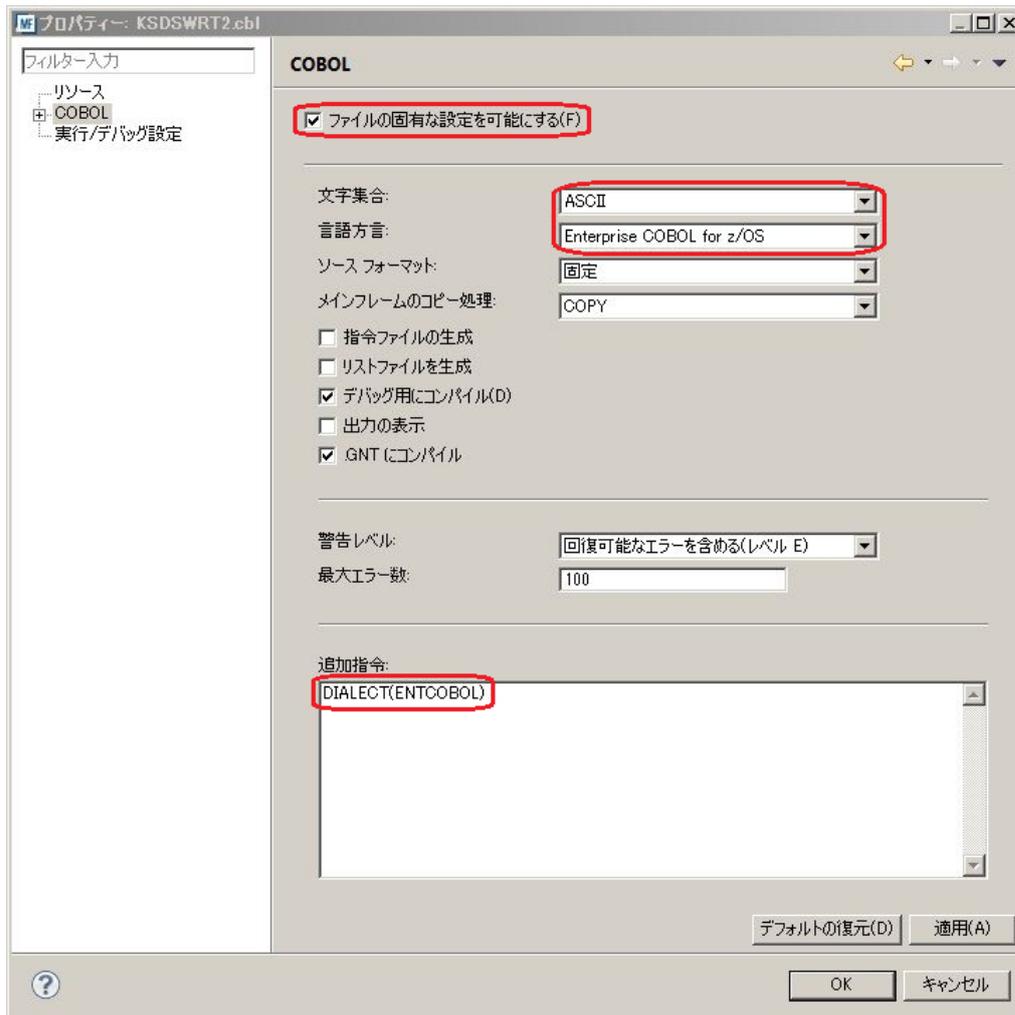
5.5 バッチプログラムのコンパイル

本チュートリアルで使用する COBOL バッチプログラム KSDSWRT2.cbl は既に MSSDEMO プロジェクトに追加されています。しかし、JCL から実行されるためにはメインフレーム方言にてコンパイルする必要があるため、設定を追加して再コンパイルしておきます。

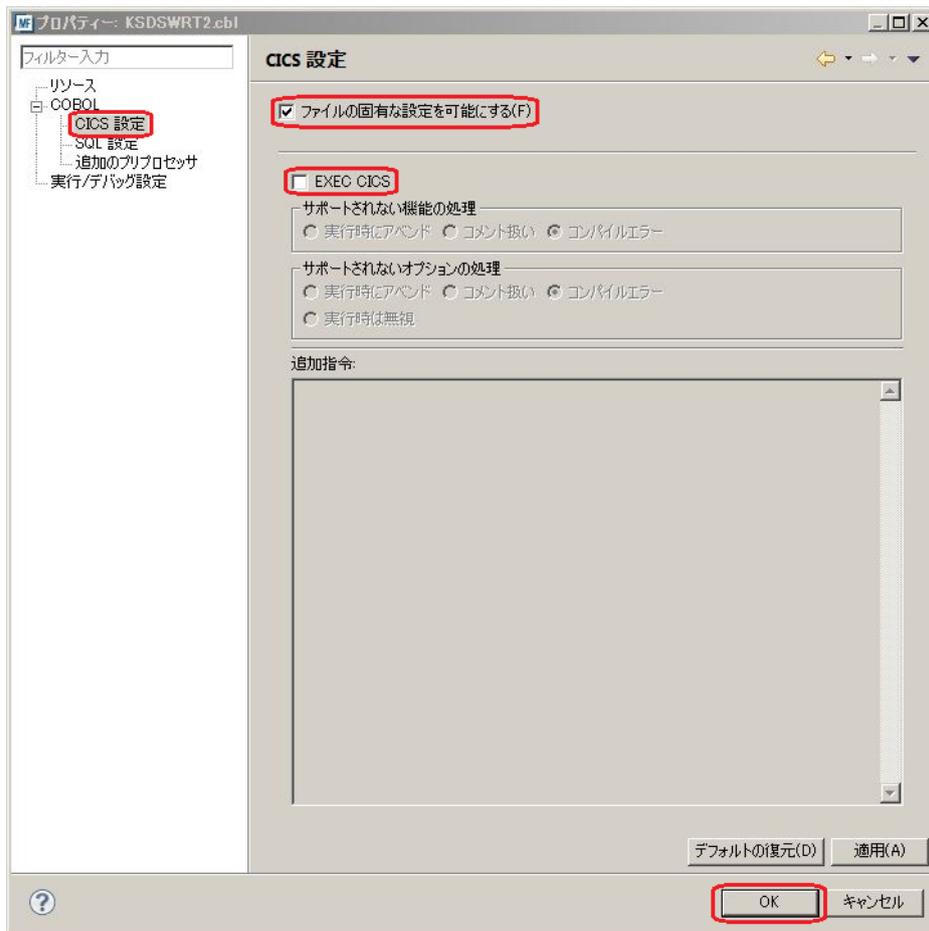
1) MSSDEMO プロジェクトの COBOL エクスプローラ内で KSDSWRT2.cbl を右クリックし [プロパティ] を選択します。



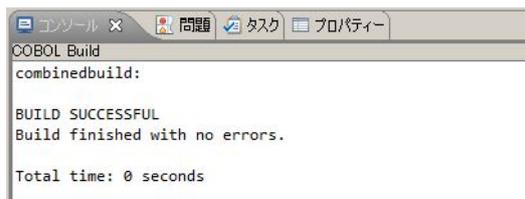
2) 以下のダイアログにてプロジェクト内の個別のソースファイルに対する設定を行うことができます。ここでの設定はプロジェクトに対する設定より優先して採用されます。以下のように設定します。



3) 同様に CICS プリプロセッサも不要ですので以下のように外します。



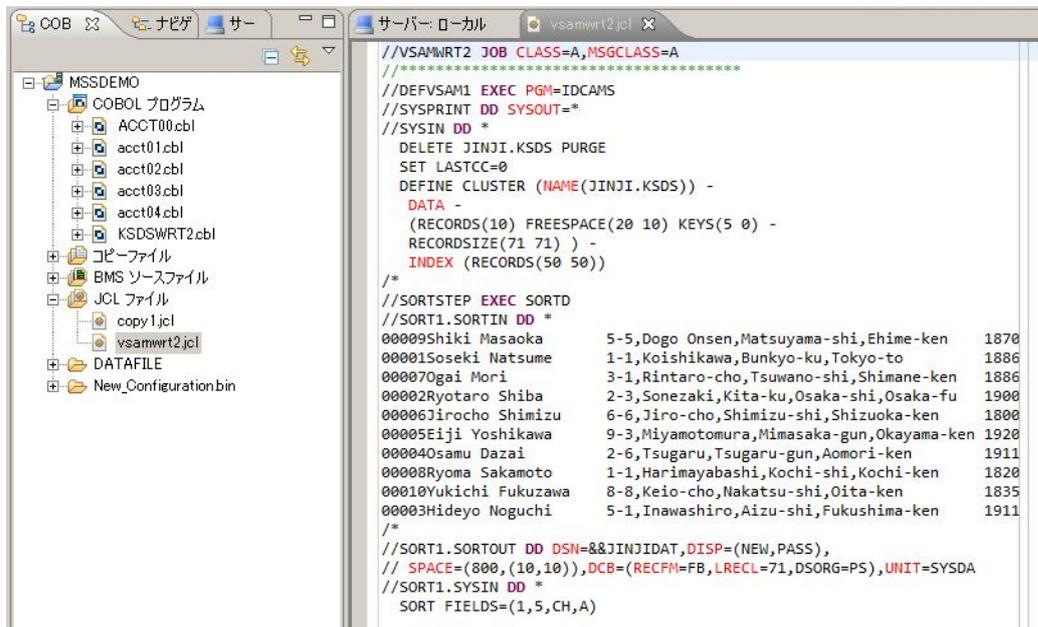
4) [OK] をクリックすると自動ビルドが実行され、以下のように再コンパイルが完了します。



5.6 COBOL バッチプログラムの実行

COBOL 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) COBOL エクスプローラ内で vsamwrt2.jcl をダブルクリックしエディタで開きます。



このジョブは4つのステップから構成され下記の様に連携されています。

- STEP1: DEFVSAM1

IDCAMSを使用して KSDS クラスター”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セージャ 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 クラスターに書き込みを行う。同時に書き込まれたデータを DD=PRINTER へ出力

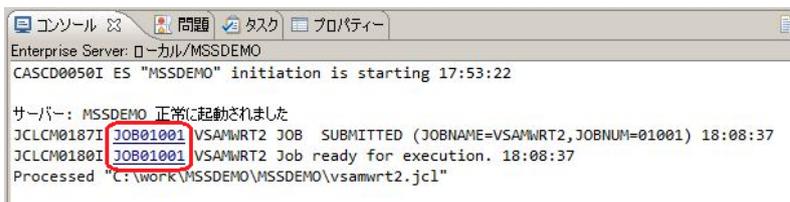
- STEP4: 書き込まれた内容の確認のため AMS の REPRO で内容を出力

2) COBOL エクスプローラ内で vsamwrt2.jcl を右クリックして [Enterprise Server へのサブミット] を選択します。

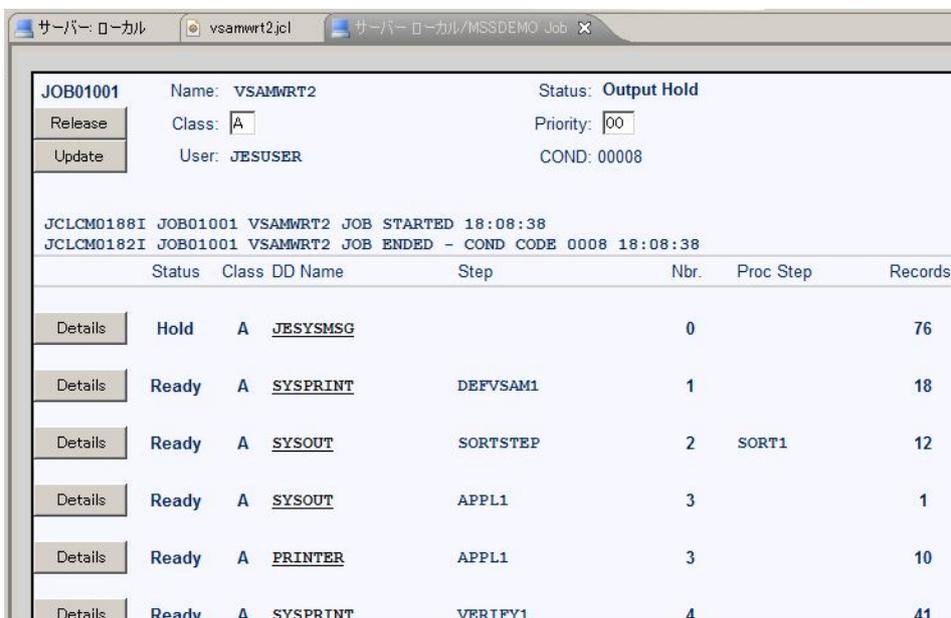


3) 以下のように VSAMWRT2 ジョブが実行された旨のメッセージがコンソールに表示されますので、

ジョブ番号をクリックします。



- 4) ジョブの実行結果を確認します。実行された VSAMWRT2 のジョブ結果で COND: 00008 が出ています。これは、このジョブの STEP1 において一旦 JINJI.KSDS ファイルを削除する処理が入っているにも関わらず、今回が最初の実行であるため、そのファイルが存在していなかったことが理由ですので問題はありません。2度目以降の実行であれば、0000 でジョブが正常に終了するはずです。



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


```
Content-Type: text/plain

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

DELETE JINJI.KSDS PURGE

JCLAM0115E(08) - ENTRYNAME NOT CATALOGED [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)

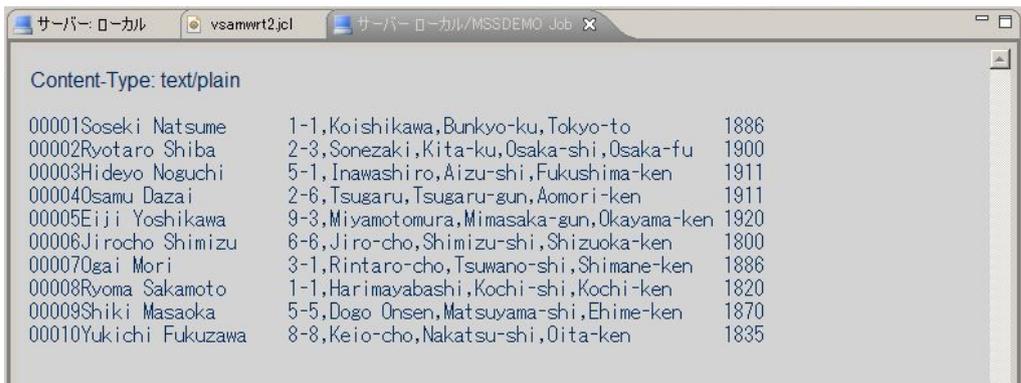
SORT204I: ***** ソート結果 *****
SORT205I: INPUT ファイル 'SORTIN'
          入力レコード          10 件
          使用レコード          10 件
SORT206I: OUTPUT ファイル 'SORTOUT'
          使用レコード          10 件
          出力レコード          10 件
SORT399I: Micro Focus MFJSORT ユーティリティ終了
```

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**
```

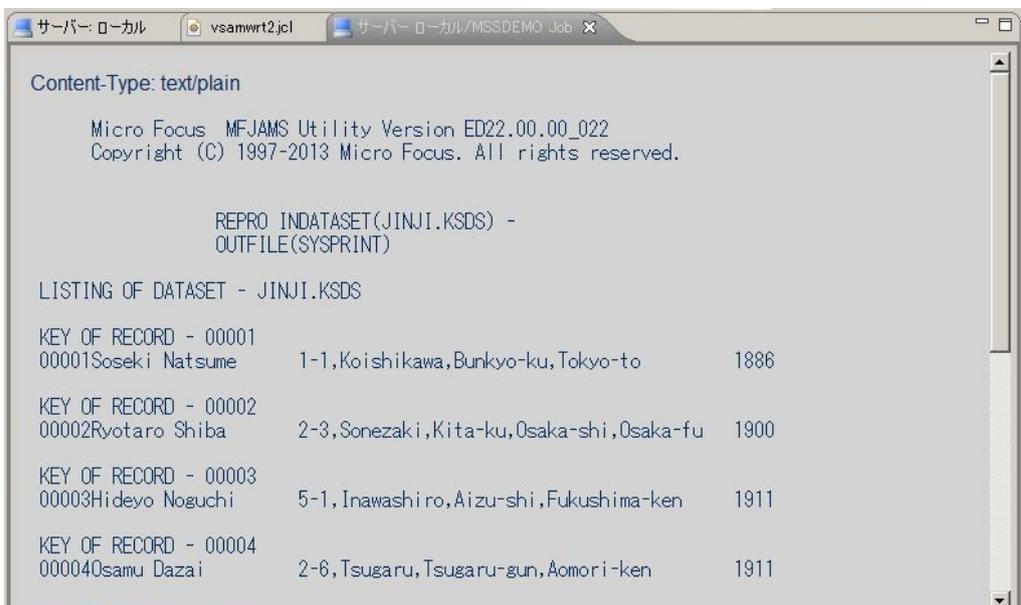
10) 次は COBOL プログラム KSDSWRT2 の PRINTER 出力です。



```
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,Tsumano-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

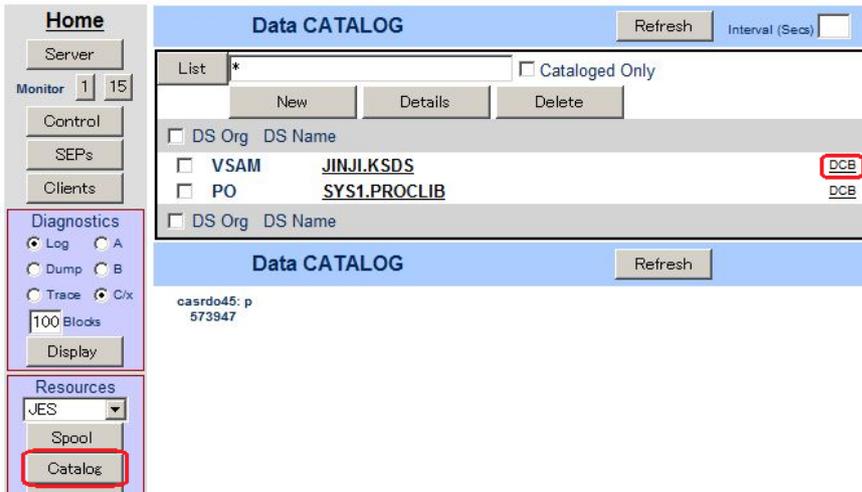
KEY OF RECORD - 00002
00002Ryotaro Shiba      2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 1900

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

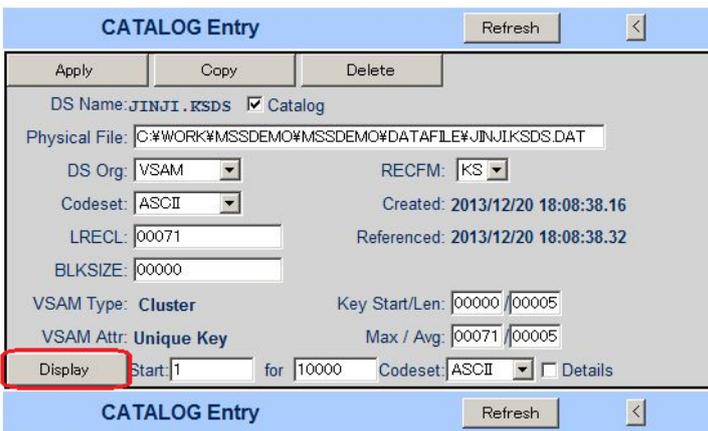
KEY OF RECORD - 00004
00004Osamu Dazai        2-6,Tsugaru,Tsugaru-gun,Aomori-ken     1911

KEY OF RECORD - 00005
```

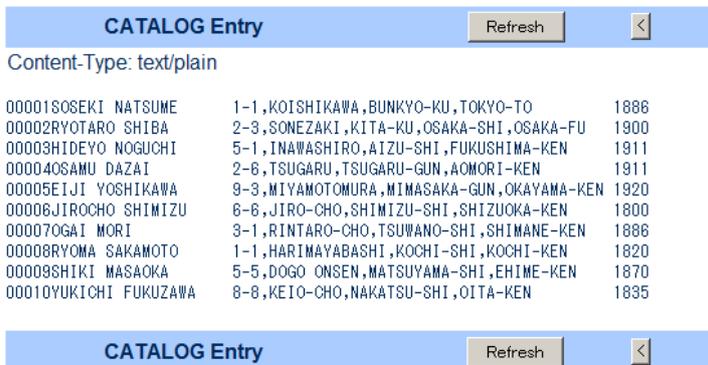
12) 続いてこのジョブの実行によってカタログされたデータセットを見えます。以下のように ESMAC 画面でカタログビューを開き、[List] ボタンをクリックします。VSAM ファイル JINJI.KSDS がカタログされていることがわかります。



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。



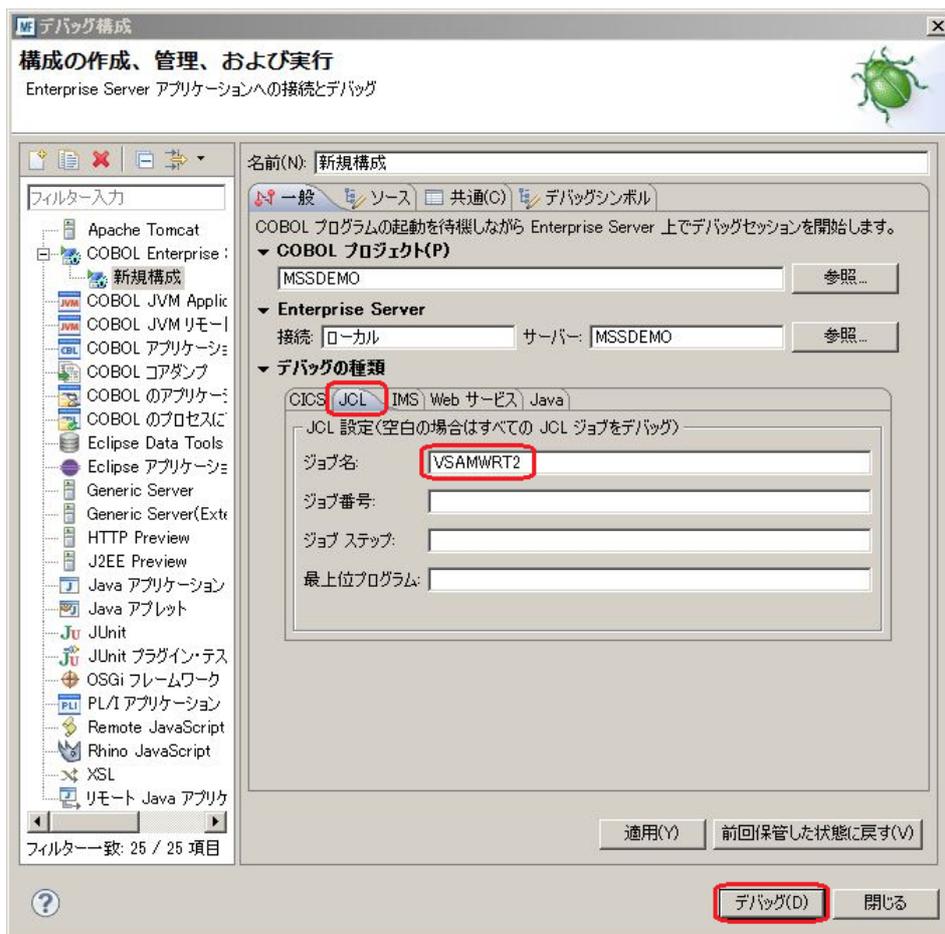
5.7 COBOL バッチプログラムのデバッグ

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。

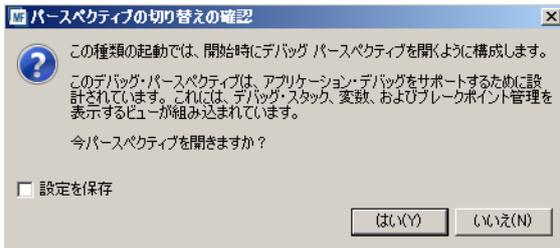
1) [実行] > [デバッグの構成...] を選択します。



2) 以下のダイアログで、COBOL プロジェクトとして MSSDEMO を選択し、[デバッグの種類] の JCL タブでデバッグ対象ジョブ名 VSAMWRT2 を入力します。[デバッグ] ボタンをクリックします。



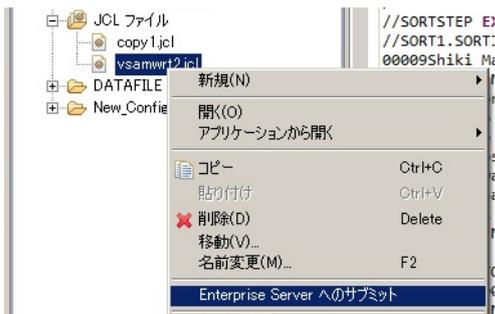
3) 以下のダイアログに対して [いいえ] をクリックします。



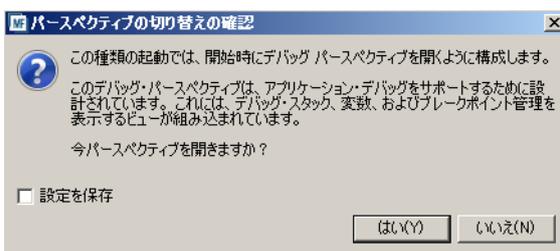
4) デバッガがアタッチの待機状態で起動します。



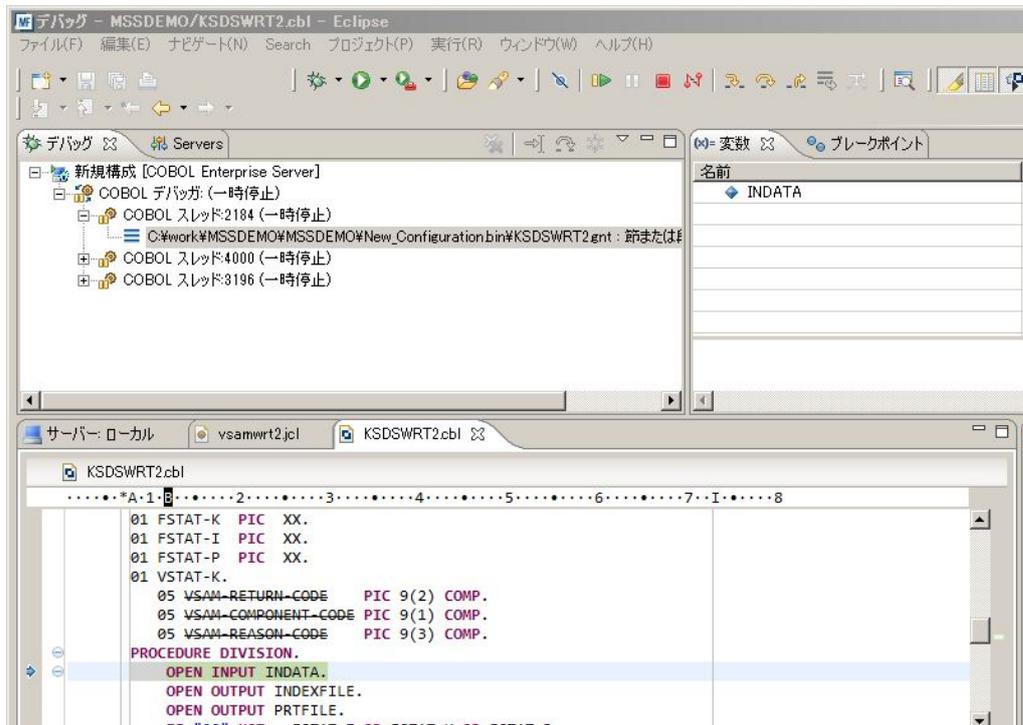
5) COBOL エクスプローラ内で vsamwrt2.jcl を右クリックし [Enterprise Server へのサブミット] を選択します。



6) しばらくして以下のダイアログが現れますので、今度は [[はい] をクリックします。



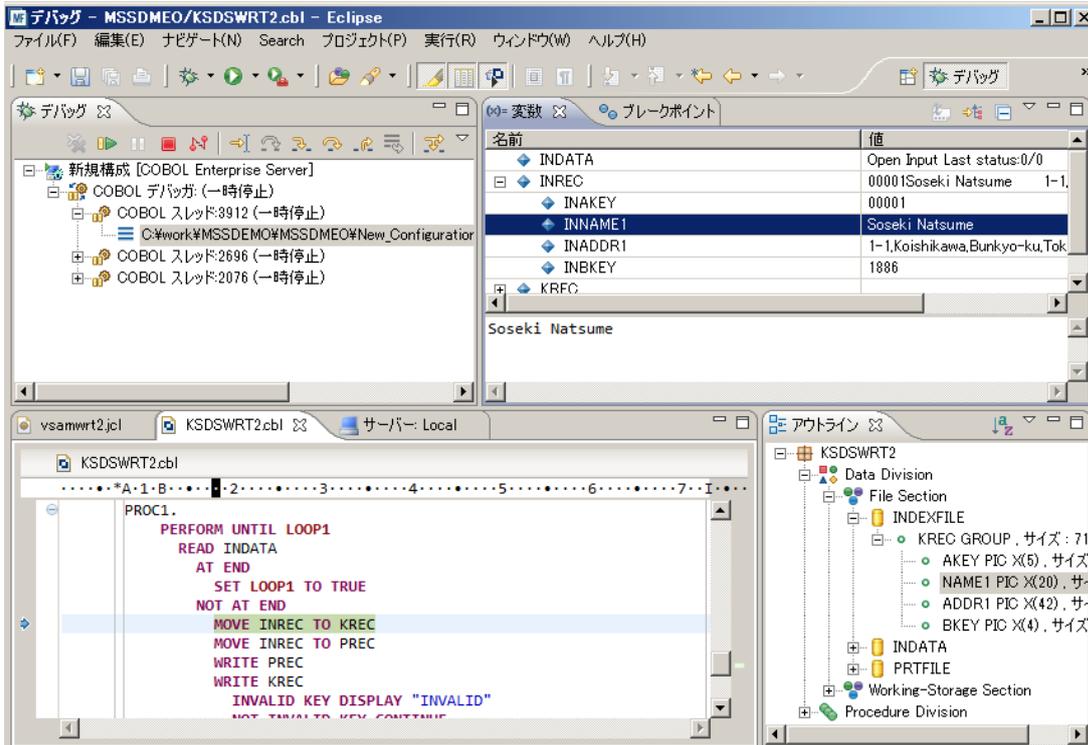
7) 以下のようにデバッグセッションが開始し、COBOL プログラム KSDSWRT2 の手続き部の先頭でハイライトされています。



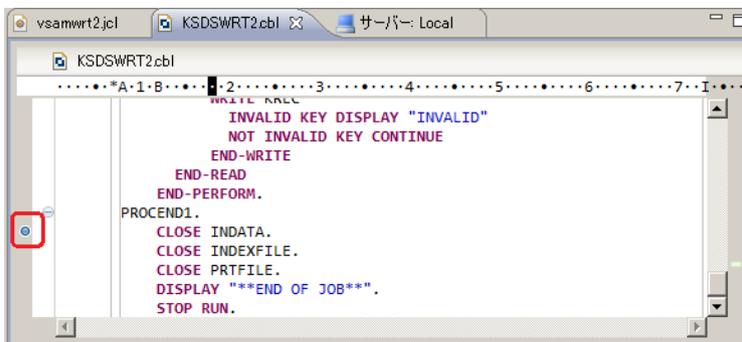
- 8) デバッグビュー内の [ステップイン] ボタンまたは F5 キーでプログラムをステップ実行することができます。



- 9) ステップ実行を進めてゆくと、以下のように変数ビュー内に更新されたデータ項目の内容が自動的に表示されます。以下は READ 文でレコード内にデータファイルから読み込まれた内容が表示されている様子を示しています。



10) 以下のように PROCEND1 段落の CLOSE 文の左端の領域をダブルクリックすると、この行にブレークポイントが設定されます。



11) [再開] ボタ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。



以上で JCL チュートリアルを終了します。