
Micro Focus Enterprise Developer チュートリアル

メインフレーム COBOL 開発 : JCL

Eclipse 編

1. 目的

本チュートリアルでは、Eclipse を使用したメインフレーム COBOL プロジェクトの作成、コンパイル、JCL の実行、デバッグまでを行い、その手順の習得を目的としています。

2. 前提

- 本チュートリアルで使用したマシン OS : Windows 8 Enterprise
- 使用マシンに Micro Focus Enterprise Developer 2.3 Update 2 for Eclipse がインストールされていること

3. チュートリアル手順の概要

1. チュートリアルの準備
2. Eclipse の起動
3. メインフレーム COBOL プロジェクトの作成
4. プロジェクトプロパティの設定
5. ビルドの実行
6. Enterprise Server インスタンスの設定
7. Enterprise Server インスタンスの開始と確認
8. JCL の実行
9. プロシージャライブラリの作成
10. COBOL バッチプログラムの実行
11. COBOL バッチプログラムのデバッグ
12. Enterprise Server インスタンスの停止

3.1 チュートリアル準備

例題プログラムに関連する資源を用意します。

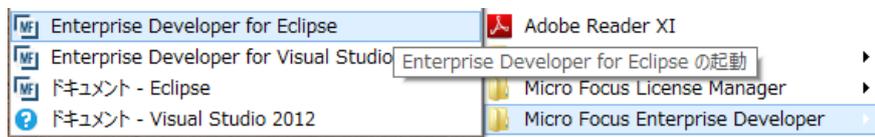
- 1) 使用する例題プログラムは、キットに添付されている Tutorials.zip に圧縮されています。これを C:¥ 直下に解凍します。



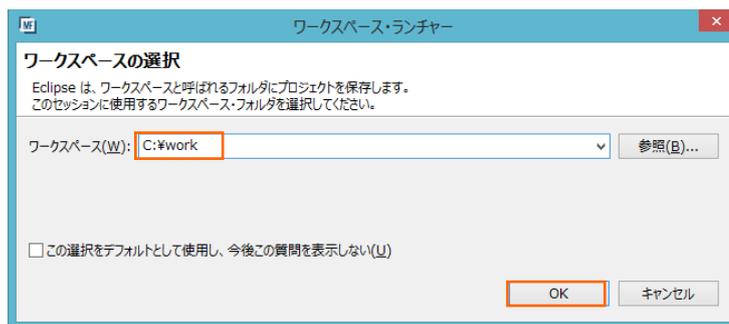
- 2) Eclipse のワークスペースで使用する「work」フォルダを C:¥ 直下に作成します。

3.2 Eclipse の起動

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



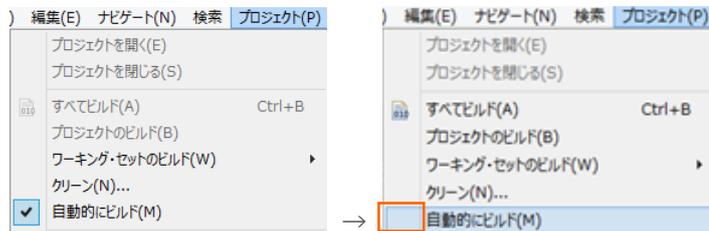
- 2) 前項で作成した「C:¥work」をワークスペースへ指定して、[OK] ボタンをクリックします。



- 3) [ようこそ] タブが表示されたら [Open COBOL Perspective] をクリックして、COBOL パースペクティブを開きます。

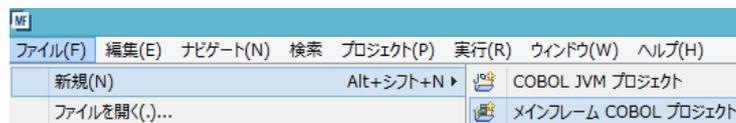


- 4) パースペクティブ表示後、[プロジェクト] プルダウンメニューの [自動的にビルド] を選択して、これをオフにします。

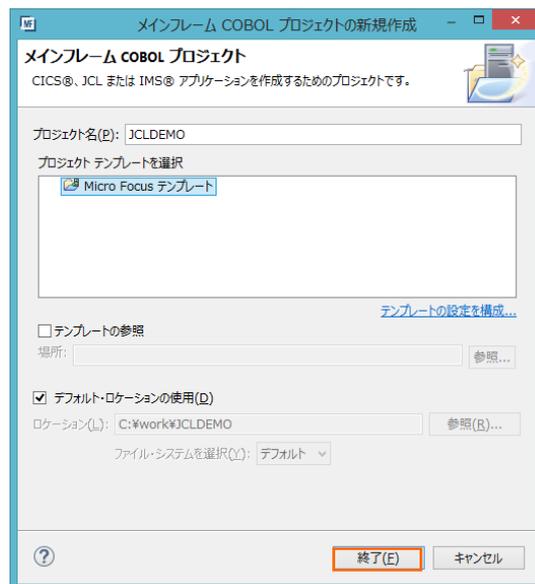


3.3 メインフレーム COBOL プロジェクトの作成

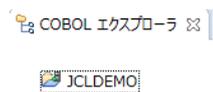
- 1) 用意した例題ソースをインポートします。[ファイル] プルダウンメニューから [新規] > [メインフレーム COBOL プロジェクト] を選択します。



- 2) [プロジェクト名] は任意ですが、ここでは “JCLDEMO” を入力して [終了] ボタンをクリックします。

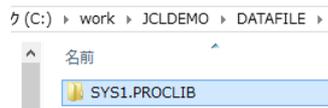


- 3) [COBOL エクスプローラー] へ作成したプロジェクトが表示されます。

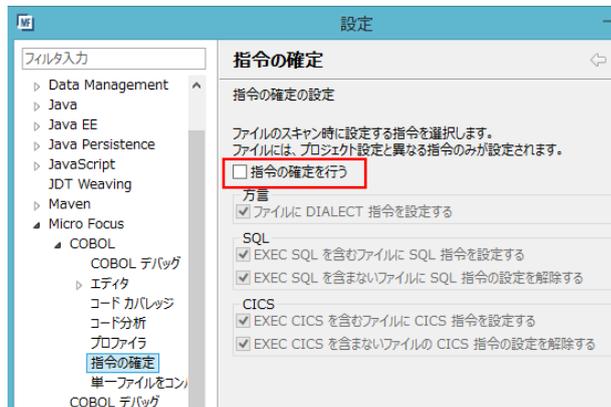


4) プロジェクトを作成したことにより C:\work\JCLDEMO フォルダが作成されています。このフォルダ配下に JES 機能で使用するフォルダをあらかじめ用意しておきます。

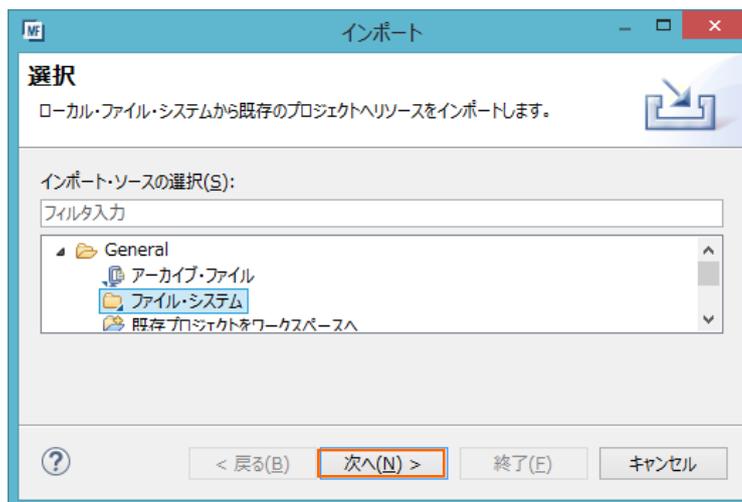
- ① カタログファイルやスプールファイルを配置するため “DATAFILE” フォルダを C:\work\JCLDEMO 配下へ作成します。
- ② プロシージャファイルを配置するため、プロシージャライブラリとして “SYS1.PROCLIB” フォルダを C:\work\JCLDEMO\DATAFILE 配下へ作成します。



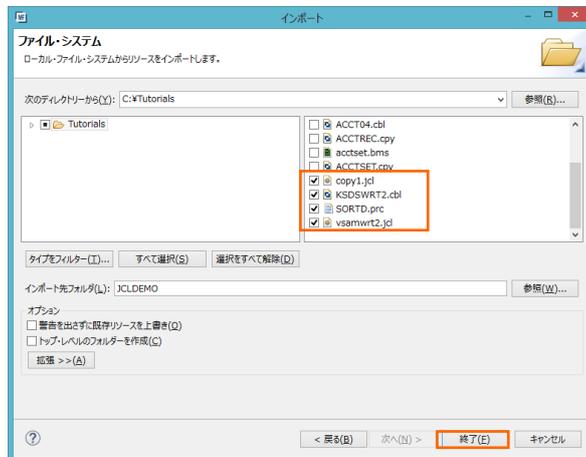
5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ウインドウ] プロダクションメニューの [設定] > [Micro Focus] > [COBOL] > [指令の確定] > [指令の確定を行う] チェックボックスをオフにして [OK] ボタンをクリックします。



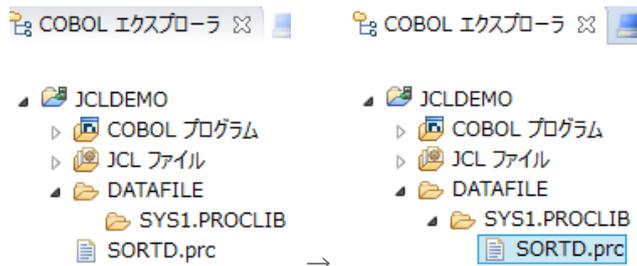
6) 用意した例題プログラム類をインポートします。[JCLDEMO] プロジェクトを右クリックして [インポート] > [インポート] を選択し、インポートウィンドウにて [General] > [ファイル・システム] を選択後 [次へ] ボタンをクリックします。



- 7) “C:\Tutorials” を [次のディレクトリーから] へ指定すると内容が表示されますので、最後から 4 ファイルをオンにして [終了] ボタンをクリックします。この実行により、プロジェクトフォルダへ例題プログラムが配置されます。

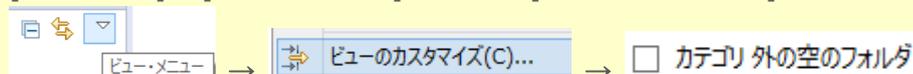


- 8) [COBOL エクスプローラー] 内に表示されている [JCLDEMO] にインポートしたファイルが表示されていることを確認して、[SORTD.prc] ファイルを作成した [SYS1.PROCLIB] ヘドラッグしてドロップします。プロシージャファイルが [SYS1.PROCLIB] フォルダ配下へ移動しました。



注意

フォルダ内が空のため作成したフォルダが表示されない場合があります。その際は [COBOL エクスプローラー] 右上の [ビュー・メニュー] > [ビューのカスタマイズ] を選択して [カテゴリ外の空のフォルダ] のチェックをオフにしてください。



3.4 プロジェクトプロパティの設定

プログラム内容に沿ったプロジェクトのプロパティを設定します。

- 1) [COBOL エクスプローラー] 内の [JCLDEMO] プロジェクトを右クリックして [プロパティ] を選択します。

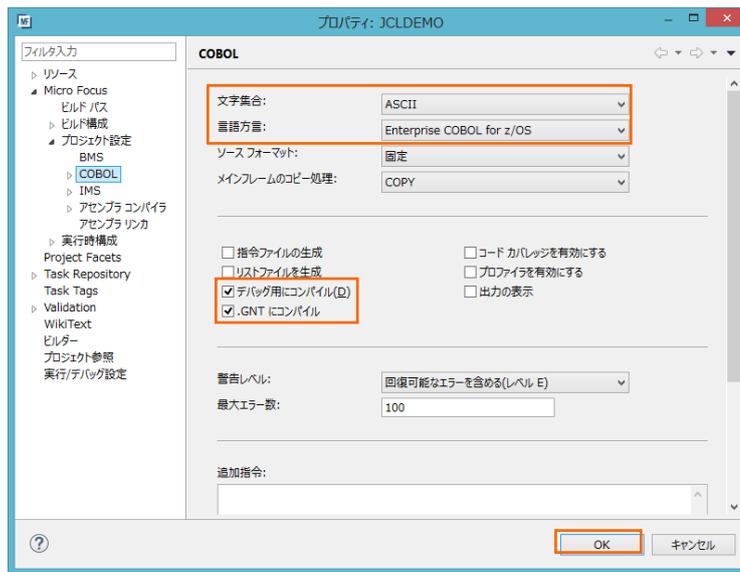
- 2) 左側ツリービューの [Micro Focus] > [ビルド構成] > [COBOL] を選択して、下記項目を指定します。指定後は [OK] ボタンをクリックしてください。

項目名	説明
ターゲットの種類	実行ファイル形式を指定します。ここでは [全て INT/GNT ファイル] を選択します。
プラットフォーム ターゲット	稼働ビット数を指定します。ここでは [64 ビット] を指定します。



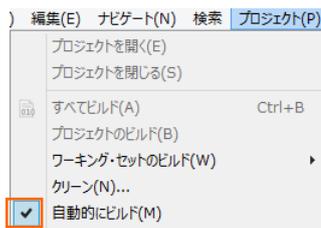
- 3) 再度プロパティウィンドウを開き、左側ツリービューの [Micro Focus] > [プロジェクト設定] > [COBOL] を選択して、下記項目を指定します。指定後は [OK] ボタンをクリックしてください。

項目名	説明
文字集合	EBCDIC または ASCII を指定します。ここでは [ASCII] を選択します。
言語方言	COBOL 言語方言を指定します。 例題プログラムは IBM Enterprise COBOL の方言を使用しているため、ここでは [Enterprise COBOL for z/OS] を指定します。
デバッグ用にコンパイル	デバッグ実行時に使用するファイルを生成するように指定します。
.GNT にコンパイル	実行ファイル形式を GNT に指定します。
追加指令	ここでは指定しません。

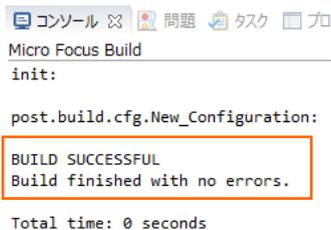


3.5 ビルドの実行

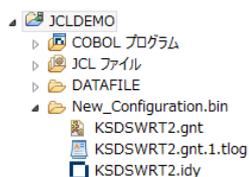
- 1) [プロジェクト] プルダウンメニューの [自動的にビルド] を選択して、これをオンすると自動的にビルドが実行されます。



- 2) [コンソール] タブで成功を確認します。



- 3) [COBOL エクスプローラー] のプロジェクト内に存在する [New_Configuration.bin] フォルダ配下に実行ファイル (.gnt ファイル) が作成されていることを確認してください。



3.6 Enterprise Server インスタンスの設定

Enterprise Server インスタンスには JES をエミュレーションする機能が搭載されており、この開発用インスタンスを使用してメインフレームアプリケーションのテスト実行やデバッグを行います。本番環境には実行製品である Enterprise Server をインストールし、本番用インスタンス上でアプリケーションを稼働させます。

- 1) Enterprise Server インスタンスを作成します。[サーバー エクスプローラー] タブの [ローカル] を右クリックして [Administration ページを開く] を選択します。デフォルトポート番号は 86 です。



- 2) Enterprise Server Administration 画面に Enterprise Server インスタンス一覧が表示されますので、画面の左下にある [追加] ボタンをクリックします。



- 3) サーバー名には [JCLDEMO] を入力、動作モードは 64-bit を指定して [次へ] ボタンをクリックします。

サーバー追加 (Page 1 of 3):

サーバー名:

動作モード:

32-bit 64-bit

You cannot change your choice of work

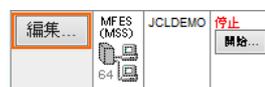


実行ファイル生成に指定した稼働ビット数 = Enterprise Server インスタンス稼働ビット数である必要があります。

- 4) 画面の Page 2/3 ではそのまま [次へ] ボタンを、Page 3/3 では [TN3270 リスナーの作成] チェックボックスをオフにして [追加] ボタンをクリックすると、[JCLDEMO] という名前の 64 ビットアプリケーション稼働用 Enterprise Server インスタンスが追加されます。



- 5) 左にある [編集] ボタンをクリックします。



6) [サーバー] > [プロパティ] > [一般] タブ内の下記項目を設定します。

- ① [動的デバッグを許可] チェックボックスをオンにします。この指定により、Eclipse からの動的デバッグが可能になります。

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"/>	要求ライセンス:	<input type="text" value="10"/>
ローカルコンソールを表示:	<input type="checkbox"/>	動的デバッグを許可:	<input checked="" type="checkbox"/>
Start on System Start:	<input type="checkbox"/>	64-Bit Working Mode:	<input checked="" type="checkbox"/>
以前のログを削除:	<input type="checkbox"/>	コンソールログサイズ (K):	<input type="text" value="0"/>

- ② [Apply] ボタンをクリックします。

7) [サーバー] > [プロパティ] > [MSS] > [JES] タブで表示される画面の各項目を設定します。入力後は [Apply] ボタンをクリックします。

項目名	説明
メインフレーム サブシステム サポート有効	[MSS] タブ配下の設定をオン、オフ指定します。ここではオンに指定します。
ジョブ入力サブシステム 有効	[JES] タブ配下の設定をオン、オフ指定します。ここではオンに指定します。
JES プログラム パス	COBOL アプリケーション実行ファイルが存在するパスを指定します。
システムカタログ	カタログファイルが存在するパスと、そのファイル名称を指定します。
データセットの省略時ロケーション	ジョブ実行時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デフォルトパスを指定します。
システムプロシージャライブラリ	プロシージャライブラリの名前を指定します。 ここでは "SYS1.PROCLIB" を入力します。

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 Initiators (0) Printers (0)

ジョブ入力サブシステム有効:

JESプログラムパス:

システムカタログ:

データセットの省略時ロケーション:

システムプロシージャライブラリ:

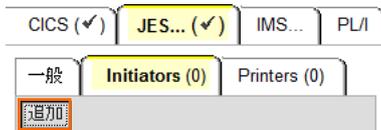
Fileshare 構成 ロケーション:

Apply

 **重要**

入力値は全て半角英数字で指定してください。
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。

- 8) [サーバー] > [プロパティ] > [MSS] > [JES] > [Initiators] タブを表示し、左下の [追加] ボタンをクリックします。



- 9) 下記画面のように入力して [追加] ボタンをクリックします。この指定により [JCLDEMO] インスタンスが開始時にイニシエータが稼働し、ジョブクラス A,B,C のジョブが実行可能になります。

名前:

Class:

説明:

- 10) 画面左上の [Home] をクリックして一覧画面に戻ります。



3.7 Enterprise Server インスタンスの開始と確認

- 1) [サーバー エクスプローラー] 内に [JCLDEMO] インスタンスが表示されていることを確認します。表示されていない場合は [ローカル] を右クリックし、[更新] を選択してリフレッシュしてください。
- 2) [サーバー エクスプローラー] 内の [JCLDEMO] インスタンスを右クリックし、[プロジェクトに関連付ける] > [JCLDEMO] を選択します。これにより [JCLDEMO] プロジェクトから実行されるアプリケーションは [JCLDEMO] インスタンスで処理されることになります。



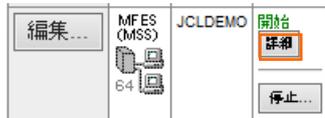
- 3) [JCLDEMO] インスタンスを右クリックして [開始] を選択します。



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

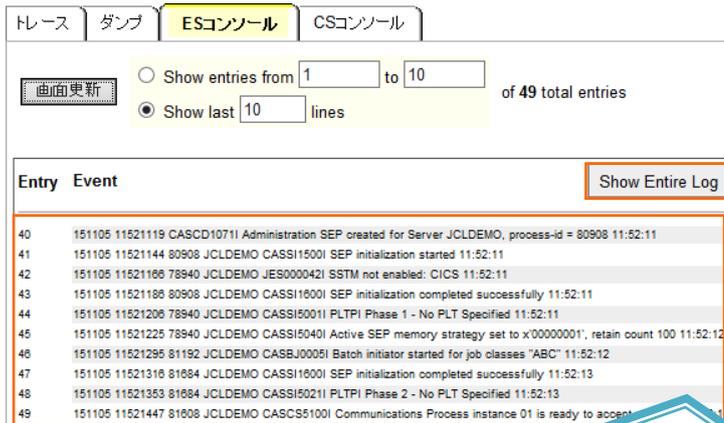


- 5) Enterprise Server Administration 画面へ移動して開始状態であることを確認後、[詳細] ボタンをクリックします。



- 6) [サーバー] > [診断] > [ES コンソール] で [JCLDEMO] インスタンスのコンソールログをリアルタイムにチェックすることができます。また [Show Entire Log] をクリックしてログ全体を表示させることも可能です。

正常に開始されたことを確認します。



【JES 機能の正常開始ログ抜粋】

```
JES000051I Job Entry Subsystem (JES) services initialized 14:09:08
CASCD1060I JES Initiator created for Server DBDEMO, process-id = 6788 14:09:08
CASBJ0006I Batch initiator initialization started 14:09:08
CASBJ0023I Batch initiator INITABC: class(es) "ABC" 14:09:08
```



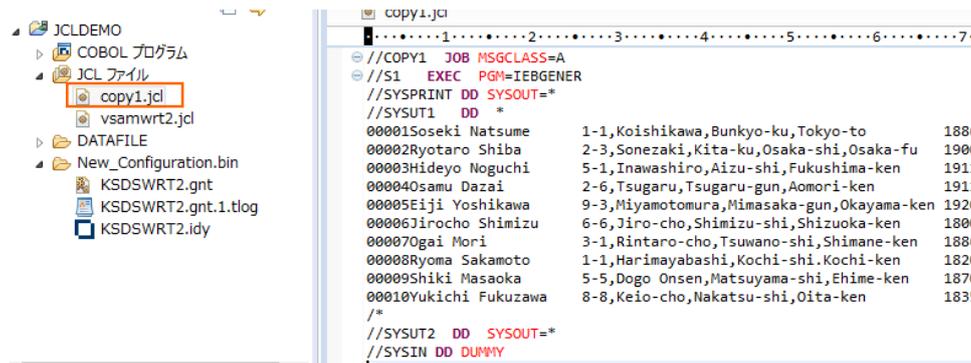
いくつかのサービス開始が失敗してもインスタンスは開始されますので、ログ内容を必ず確認してください。

- 7) 画面左上の [Home] をクリックして一覧画面に戻ります。

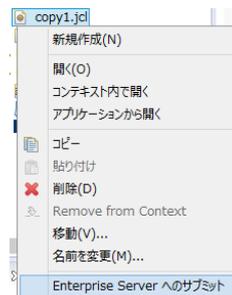
3.8 JCL の実行

現在 [JCLDEMO] インスタンスが稼働していますので、例題プログラムを実行することができます。まずは簡単な JCL を実行してみます。

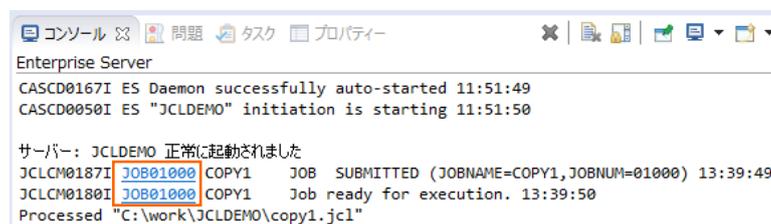
- 1) [COBOL エクスプローラー] 内にある [JCLDEMO] プロジェクト配下の [copy1.jcl] をダブルクリックし、エディタで内容を確認します。この JCL は IEBGENER ユーティリティを使用して、JCL 内に書かれたインラインデータを SYSOUT に書き出しています。



- 2) [COBOL エクスプローラー] 内の [copy1.jcl] を右クリックして [Enterprise Server へのサブミット] を選択すると、この JCL が実行されます。



- 3) [コンソール] タブに JOB 実行ログと JOB 番号が表示されますので、リンクをクリックします。



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

JOB01000	Name: COPY1	Status: Output Hold				
Release	Class: A	Priority: 00				
Update	User: JESUSER	COND: 00000				
JCLCM0188I JOB01000 COPY1 JOB STARTED 13:39:50 JCLCM0182I JOB01000 COPY1 JOB ENDED - COND CODE 0000 13:39:51						
Status	Class	DD Name	Step	Nbr.	Proc Step	Records
Details	Hold	A	JESYSMSG	0		31
Details	Ready	A	SYSPRINT	1	s1	4
Details	Ready	A	SYSUT2	1	s1	10

- 5) ジョブログの内容を確認すると、この JOB が正常に終了していることが確認できます。

```
JCLCM0182I JOB ENDED - COND CODE 0000
```

- 6) 右クリックで [前へ戻る] を選択し、スプール一覧から [SYSPRINT] をクリックすると、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。

```
Micro Focus MFJGENER Utility Version ED2.3.2_HF1F
Copyright (C) 1997-2015 Micro Focus. All rights reserved.

JCLGN0110I(00) - 0000000010 RECORDS COPIED FROM SYSUT1 TO SYSUT2
```

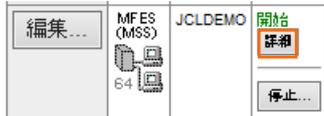
- 7) 右クリックで [前へ戻る] を選択し、スプール一覧から [SYSUT2] をクリックすると、出力されたスプールの内容が確認できます。

```
00001Soseki Natsume      1-1,Koishikawa,Bunkyo-ku,Tokyo-to      1888
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
```

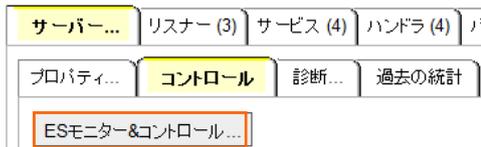
3.9 プロシージャライブラリの作成

プロシージャを使用する JCL を実行するために、プロシージャライブラリを作成します。Enterprise Server インスタンスではプロシージャを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。

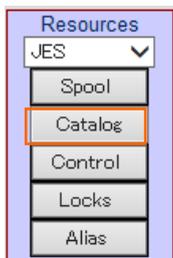
- 1) Enterprise Server Administration 画面へ移動して [JCLDEMO] インスタンスの [詳細] ボタンをクリックします。



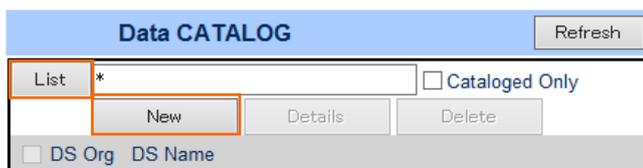
- 2) [サーバー] > [コントロール] > [ES モニター & コントロール] ボタンをクリックします。



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

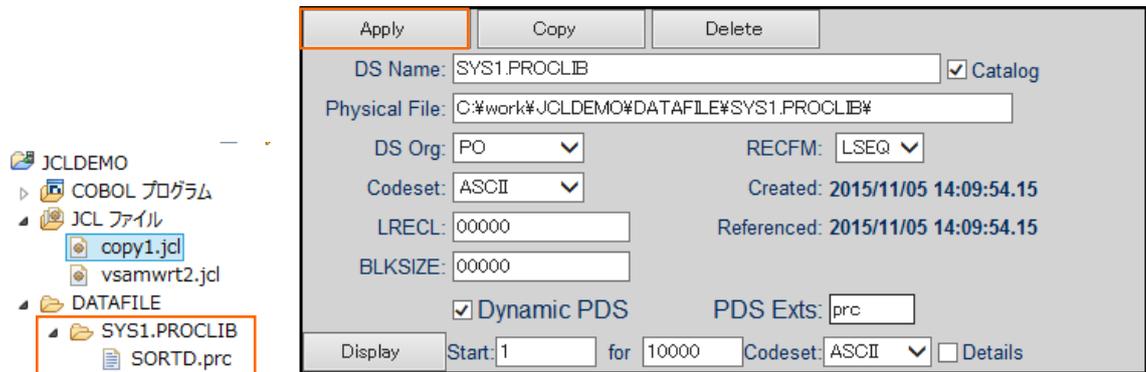


- 4) [List] ボタンをクリックして、カタログ情報の一覧を表示すると、現在は何も登録されていないことが確認できます。新規に作成するため [New] ボタンをクリックします。

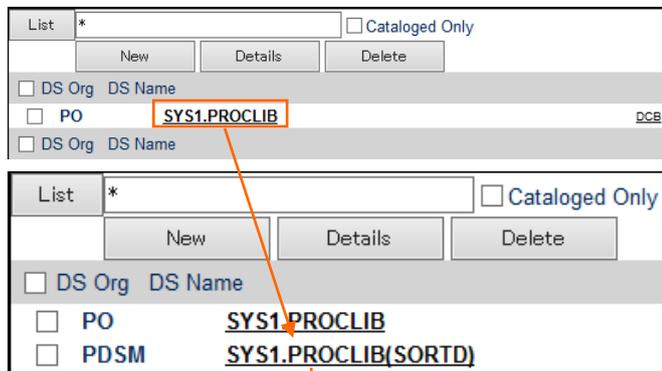


- 5) カタログエントリの画面が表示されますので、以下のように入力し [Apply] ボタンをクリックします。

項目名	説明
DS Name	“SYS1.PROCLIB” を入力します。
Physical File	存在するパスを指定。ここでは前項で作成したフォルダパスを入力します。
DS Org	区分データセットである “PO” を選択します。
RECFM	行順である “LSEQ” を選択します。
Dynamic PDS	プロシージャファイルをフォルダ配下に保持する動的 PDS の場合にオンにします。ここではオンを指定します。
PDS Exts	プロシージャファイル拡張子を指定します。ここでは “prc” を入力します。



- 6) カタログ一覧に戻り [List] ボタンをクリックすると作成した PO が表示されます。[SYS1.PROCLIB] をクリックすると配置されているメンバーが確認できます。



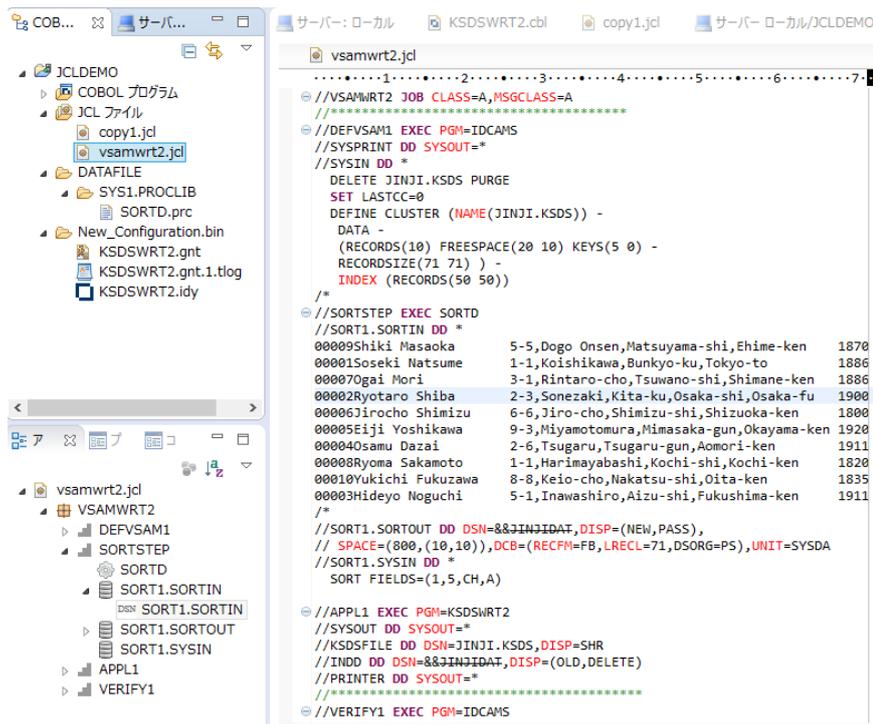
[SYS1.PROCLIB(SORTD)] をクリックすると内容が確認できます。

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

3.10 COBOL バッチプログラムの実行

COBOL プログラムを含む JOB を実行してみます。

- 1) [COBOL エクスプローラー] 内にある [JCLDEMO] プロジェクト配下の [vsamwrt2.jcl] をダブルクリックし、エディタで内容を確認します。



① ステップ 1 : DEFVSAM1

IDCAMS を使用して VSAM データセット “JINJI.KSDS” を削除し、クラスターを持つ KSDS として再作成します。

② ステップ 2 : SORTSTEP

前項で登録した SORTD.prc 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します。

③ ステップ 3 : APPL1

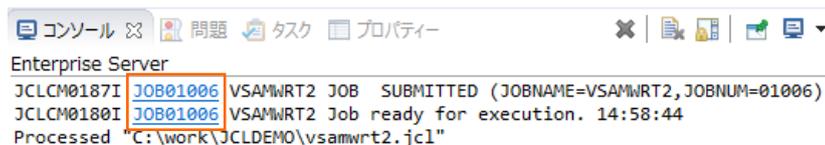
アプリケーション KSDSWRT2 を呼び出しステップ 2 でソートされたデータをステップ 1 で定義した VSAM クラスターに書き込みを行います。同時に書き込まれたデータを DD=PRINTER へ出力します。

④ ステップ 4 : VERIFY1

出力内容確認のため、IDCAMS の REPRO で内容を出力します。

2) [COBOL エクスプローラー] 内の [vsamwrt2.jcl] を右クリックして [Enterprise Server へのサブミット] を選択して、この JCL を実行します。

3) [コンソール] タブに JOB 実行ログと JOB 番号が表示されますので、リンクをクリックします。



- 4) この JOB 番号にかかわるスプール一覧が表示されます。

[COND CODE] には [0008] が返却されていますが、これは初回実行時、ステップ 1 で削除該当ファイルが見つからないためです。問題ありません。

JOB01006	Name: VSAMWRT2	Status: Output Hold				
Release	Class: A	Priority: 00				
Update	User: JESUSER	COND: 00008				
JCLCM0188I JOB01006 VSAMWRT2 JOB STARTED 14:58:44 JCLCM0182I JOB01006 VSAMWRT2 JOB ENDED COND CODE 0008 14:58:44						
Status	Class	DD Name	Step	Nbr.	Proc Step	Records
Details	Hold	A	JESYSMSG	0		76
Details	Ready	A	SYSPRINT	1	DEFVSAM1	18
Details	Ready	A	SYSOUT	2	SORT1	12
Details	Ready	A	SYSOUT	3	APPL1	1

- 8) [JESYSMSG] の内容を確認すると、各ステップの COND CODE 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。

```
JCLCM0191I STEP ENDED VERIFY1 - COND CODE 0000
JCLCM0182I JOB ENDED - COND CODE 0008
```

- 9) 右クリックで [前へ戻る] を選択し、スプール一覧から DEFVSAM1 ステップの [SYSPRINT] をクリックして内容を確認します。

```
Micro Focus MFJAMS Utility Version ED2.3.2_HF1F
Copyright (C) 1997-2015 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]
```

- 10) 右クリックで [前へ戻る] を選択し、スプール一覧から SORTSTEP の [SYSOUT] をクリックしてソート内容を確認します。

```
Content-Type: text/plain

Micro Focus MFJSORT ユーティリティ 3.0.00

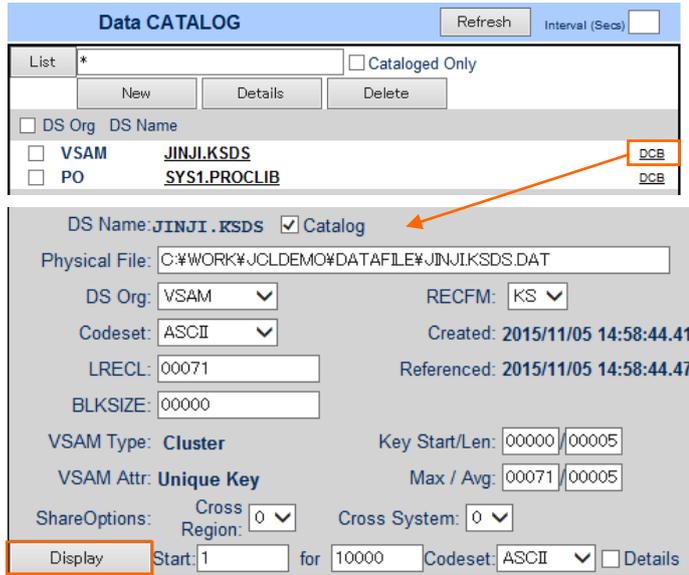
SORT FIELDS=(1,5,CH,A)

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
```

11) 右クリックで [前へ戻る] を選択し、スプルー一覧から他ステップに関しても確認してみてください。

12) 前項と同様の手順で、この JOB によってカタログされた情報を確認します。カタログ一覧で [List] ボタンをクリックすると、VSAM ファイル JINJI.KSDS がカタログされていることが確認できます。

右端の [DCB] をクリックすると登録情報が表示されます。



Data CATALOG Refresh Interval (Secs) []

List: * Cataloged Only

New Details Delete

DS Org DS Name

VSAM JINJI.KSDS DCB

PO SYS1.PROCLIB DCB

DS Name: JINJI.KSDS Catalog

Physical File: C:\WORK\JCLDEMO#DATAFILE#JINJI.KSDS.DAT

DS Org: VSAM RECFM: KS

Codeset: ASCII Created: 2015/11/05 14:58:44.41

LRECL: 00071 Referenced: 2015/11/05 14:58:44.47

BLKSIZE: 00000

VSAM Type: Cluster Key Start/Len: 00000/00005

VSAM Attr: Unique Key Max / Avg: 00071/00005

ShareOptions: Cross Region: 0 Cross System: 0

Display Start: 1 for 10000 Codeset: ASCII Details

[Display] ボタンをクリックすると、ファイ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        2-8,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
```

3.11 COBOL バッチプログラムのデバッグ

JCL から実行される COBOL プログラムをデバッグします。

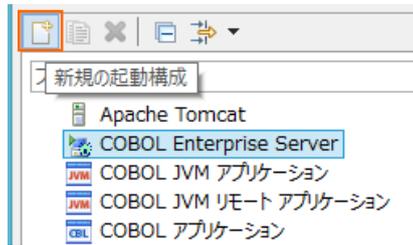
1) [実行] プルダウンメニューの [デバッグの構成] を選択します。



ソース ナビゲート(N) 検索 プロジェクト(P) **実行(R)**

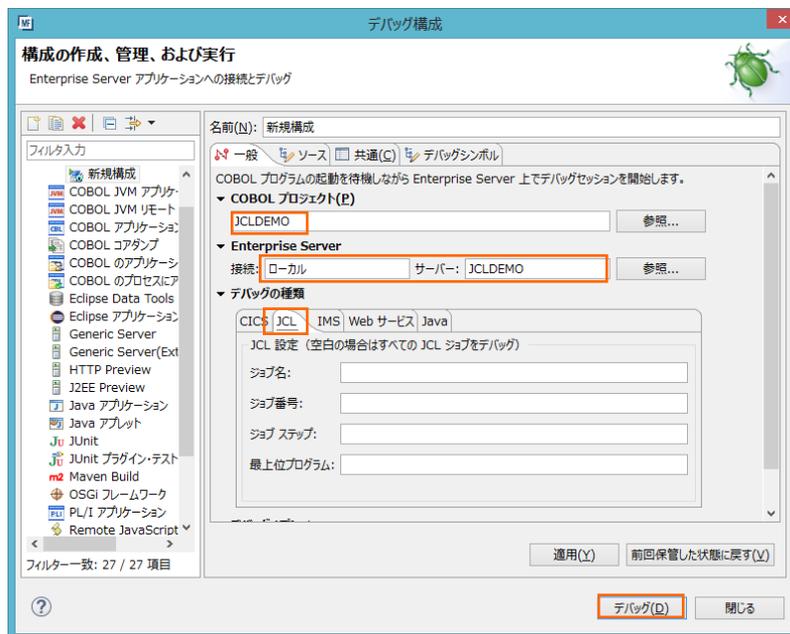
- 実行点をリセット
- 実行(R) Ctrl+F11
- デバッグ(D) F11
- 実行履歴(T) ▶
- 実行(S) ▶
- 実行構成(N)... ▶
- デバッグ履歴(H) ▶
- デバッグ(G) ▶
- デバッグの構成(B)...

- 2) 左側のメニューから [COBOL Enterprise Server] を選択して、左上の [新規の起動構成] アイコンをクリックします。

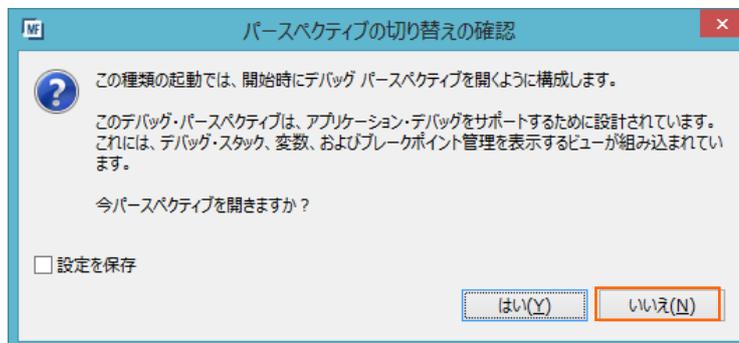


- 3) [COBOL プロジェクト] へ対象となる「JCLDEMO」を入力し、[Enterprise Server] へ実行させる「JCLDEMO」インスタンスを指定します。

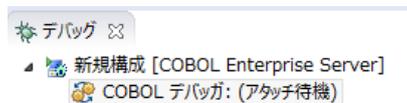
[デバッグの種類] は「JCL」タブを選択した状態で、[デバッグ] ボタンをクリックします。



- 4) パースペクティブの切り替え確認ウィンドウでは [いいえ] ボタンをクリックします。

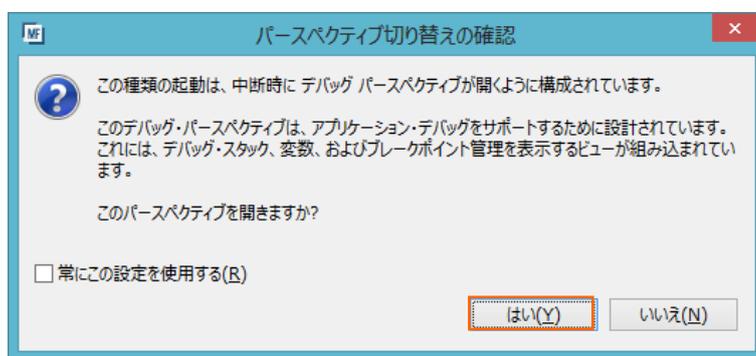


- 5) デバッグタブで [アタチ待機] 状態になったことを確認します。

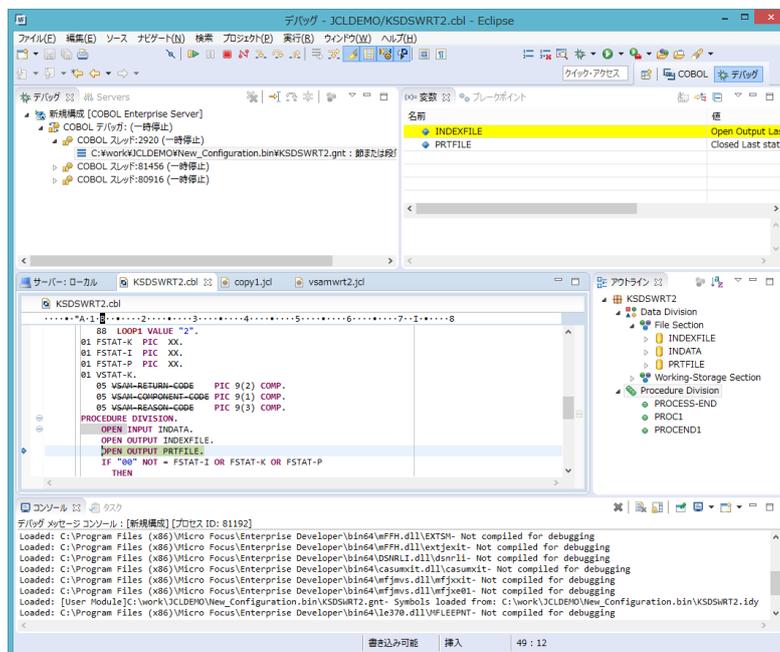


- 6) [COBOL エクスプローラー] 内の [vsamwrt2.jcl] を右クリックして [Enterprise Server へのサブミット] を選択して、JCL を実行します。

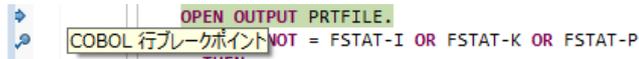
- 7) 再度、パースペクティブの切り替え確認ウィンドウが表示されますので、ここでは [はい] ボタンをクリックし、デバッグ用のパースペクティブを開きます。



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



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



10) 先に進める場合は画面上部の再開アイコンをクリックします。



11) デバッグを終了させるため、画面上部の終了アイコンをクリックします。

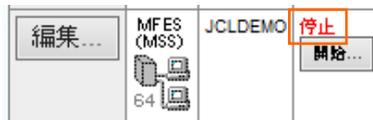


3.12 Enterprise Server インスタンスの停止

1) [JCLDEMO] インスタンスを停止します。



2) [JCLDEMO] インスタンスの停止を確認後、Eclipse を終了します。



WHAT'S NEXT

- リモート メインフレーム COBOL 開発 : JCL Eclipse 編
- 本チュートリアルで学習した技術の詳細については製品マニュアルをご参照ください。