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

2.1 本チュートリアルで使用したマシン OS：Windows 11 Pro

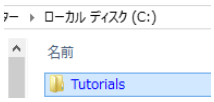
2.2 使用マシンに Enterprise Developer 9.0 for Eclipse がインストールされていること

3 チュートリアル手順の概要

3.1　チュートリアルの準備

例題プログラムに関連する資源を用意します。

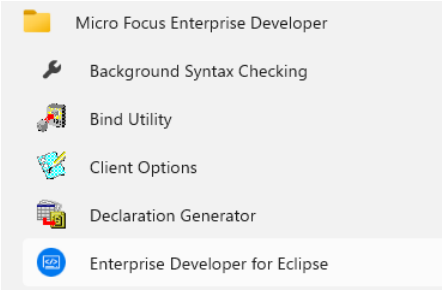
3.1.1　使用する例題プログラムは、キットに添付されている Tutorials.zip に圧縮されています。これを C:¥ 直下に解凍します。



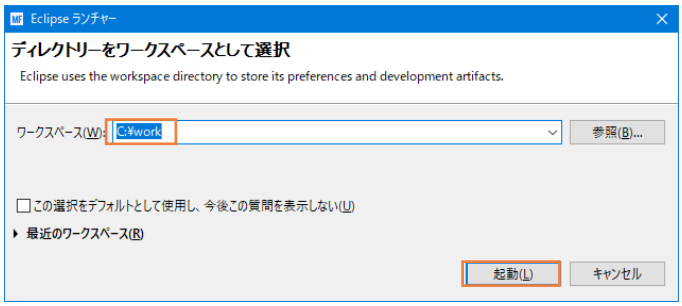
3.1.2　Eclipse のワークスペースで使用する「work」フォルダを C:¥ 直下に作成します。

3.2　Eclipse の起動

3.2.1　メニューから［Enterprise Developer for Eclipse］を起動します。

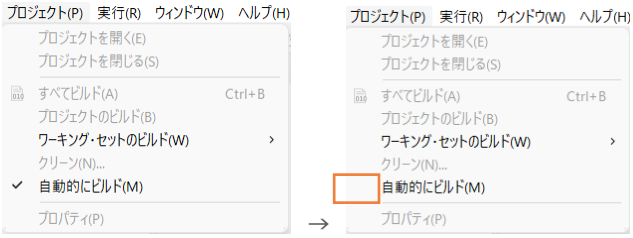


3.2.2　前項で作成した C:¥work をワークスペースへ指定して、［起動］ボタンをクリックします。

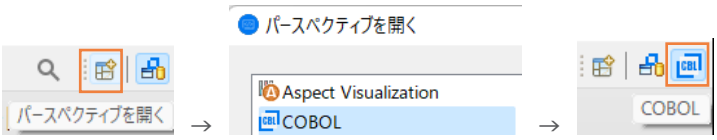


3.2.3　［ようこそ］タブでは［Open COBOL Perspective］をクリックして、COBOL パースペクティブを開きます。

3.2.4 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。

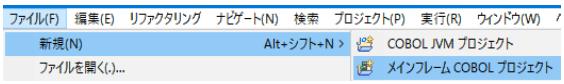


3.2.5 COBOL パースペクティブが開いていない場合は Eclipse 右上の [パースペクティブを開く] アイコンをクリックして表示後、[COBOL] を選択して [開く] ボタンをクリックします。

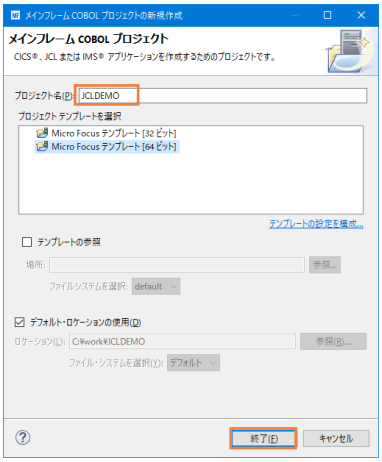


3.3 メインフレーム COBOL プロジェクトの作成

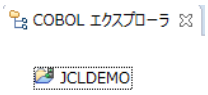
3.3.1 用意した例題ソースをインポートします。[ファイル] プルダウンメニューから [新規] > [メインフレーム COBOL プロジェクト] を選択します。



3.3.2 [プロジェクト名] は任意ですが、ここでは JCLDEMO を入力し、テンプレートは 64 ビットを選択して [終了] ボタンをクリックします。

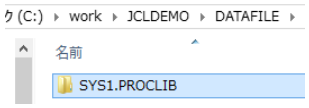


3.3.3 COBOL エクスプローラーに作成したプロジェクトが表示されます。

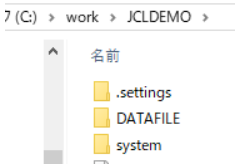


3.3.4 プロジェクトを作成したことにより C:¥work¥JCLDEMO フォルダが作成されています。このフォルダ配下に JES 機能で使用するフォルダを Windows エクスプローラーを使用してあらかじめ用意しておきます。

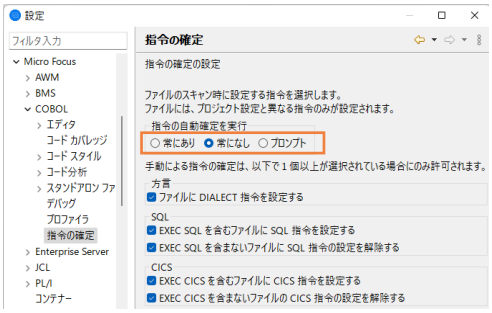
- カタログファイルやスプールファイルを配置するため DATAFILE フォルダを C:¥work¥JCLDEMO 配下へ作成します。
- プロシージャファイルを配置するため、プロシージャライブラリとして SYS1.PROCLIB フォルダを C:¥work¥JCLDEMO¥DATAFILE 配下へ作成します。



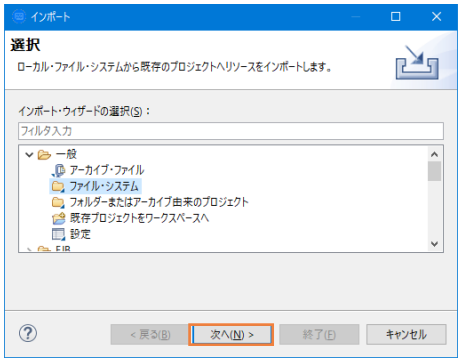
- 実行時に使用する system フォルダを C:¥work¥JCLDEMO 配下へ作成します。



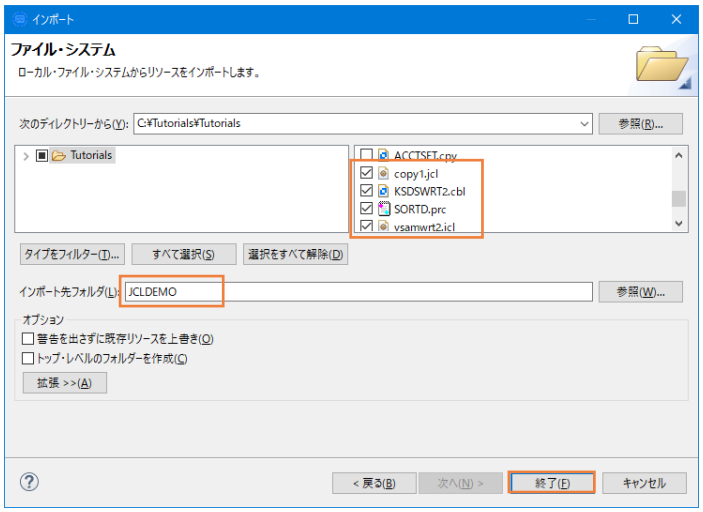
3.3.5 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。［ウィンドウ］プルダウンメニューの［設定］>［Micro Focus］>［COBOL］>［指令の確定］>［指令の自動確定を実行］で［常になし］を選択し、［適用して閉じる］ボタンをクリックします。



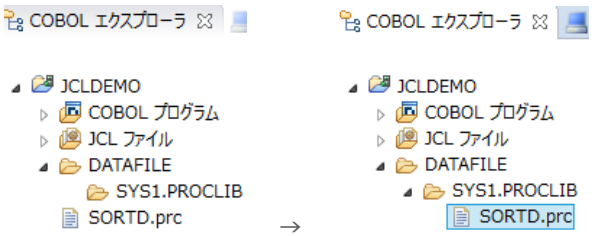
3.3.6 用意した例題プログラム類をインポートします。JCLDEMO プロジェクトを右クリックして［インポート］>［インポート］を選択し、インポートウィンドウにて［一般］>［ファイル・システム］を選択後［次へ］ボタンをクリックします。



3.3.7 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 ファイルにチェックをして［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。

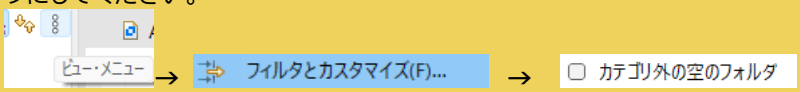


3.3.8 COBOL エクスプローラー内に表示されている JCLDEMO プロジェクトにインポートしたファイルが表示されていることを確認後、作成した SYS1.PROCLIB フォルダへ SORTD.prc ファイルをドラッグ＆ドロップして移動します。



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］>［フィルタとカスタマイズ］を選択後［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。

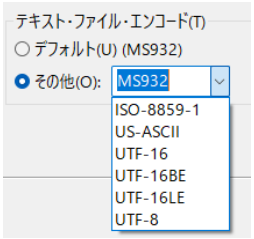


3.4 テキストファイルのエンコード指定

Eclipse ではワークスペースの設定として、テキストファイルのエンコードを指定できます。ソースファイル類のエンコードに沿って適切なエンコードを指定してください。この例題では MS932 を使用します。

3.4.1 Eclipse の［ウィンドウ］プルダウンメニューから［設定］を選択し、設定ウィンドウを表示します。

3.4.2 左側ペインで［一般］>［ワークスペース］を選択し、右側ペインの［テキスト・ファイル・エンコード］に MS932 が指定されていることを確認します。デフォルトが MS932 ではない場合は、その他を選択して MS932 を指定してください。



3.4.3　指定後は［適用して閉じる］ボタンをクリックします。

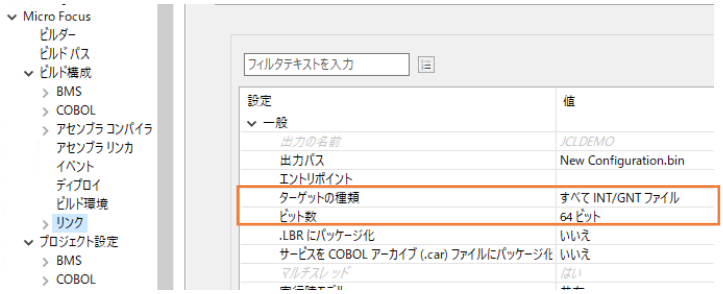
3.5　プロジェクトプロパティの設定

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。

3.5.1　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。

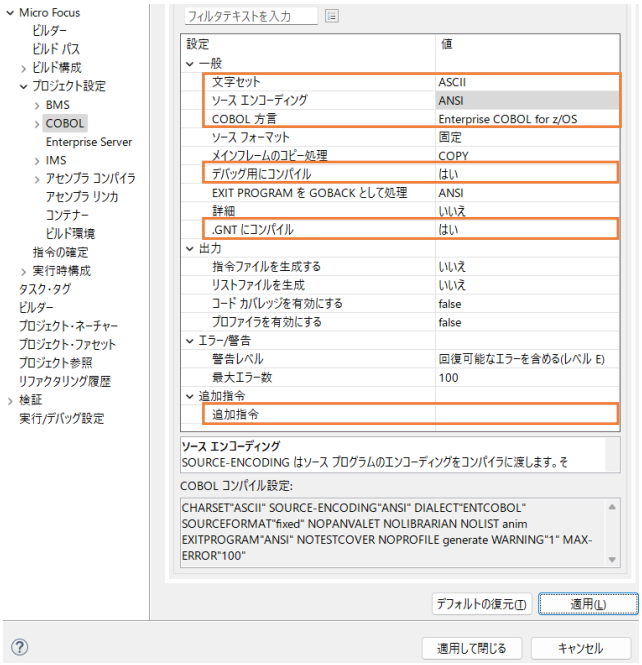
3.5.2　左側ツリービューの［Micro Focus］>［ビルド構成］>［リンク］を選択して、下記項目を指定します。指定後は［適用］　ボタンをクリックしてください。

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。



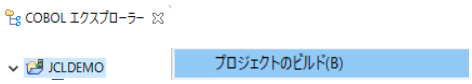
3.5.3　左側ツリービューの［Micro Focus］>［プロジェクト設定］>［COBOL］を選択して、下記項目を指定します。指定後は　［適用して閉じる］ボタンをクリックしてください。

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。
ソースエンコーディング	Eclipse に指定したエンコードと一致するエンコーディングを指定します。 ここでは MS932 を指定したため、ANSI を指定します。
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チェックをオンに指定します。
.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。

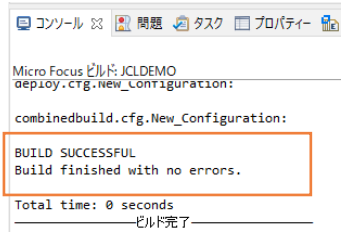


3.6 ビルドの実行

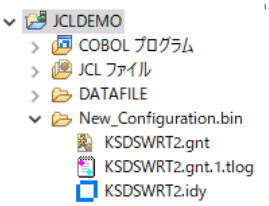
3.6.1 COBOL エクスプローラー内のプロジェクトを右クリックして［プロジェクトのビルド］を選択するとビルドが実行されます。



3.6.2 コンソールタブで成功を確認します。



3.6.3 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。



3.7 文字エンコーディングの設定

Enterprise Server インスタンスを運用、管理する Enterprise Server Common Web Administration(以降 ESCWA)では、スプールやデータ内容などに含まれる日本語を正しく表示させるために、事前に文字セットを所定のフォルダへ展開します。製品マニュアルの「リファレンス > コードセットの変換 > CCSID 変換テーブルのインストール > CCSID 変換テーブルをインストールするには」を参照しながら進めてください。

3.7.1 CCSID 変換テーブルをインストールします。

- 製品マニュアルにリンクされている下記の IBM CCSID 変換テーブルを、Web ブラウザから任意のフォルダへダウンロードします。アドレスは変更される可能性がありますので、製品マニュアルにてご確認ください。

  http://www.microfocus.com/docs/links.asp?vc=cdctables

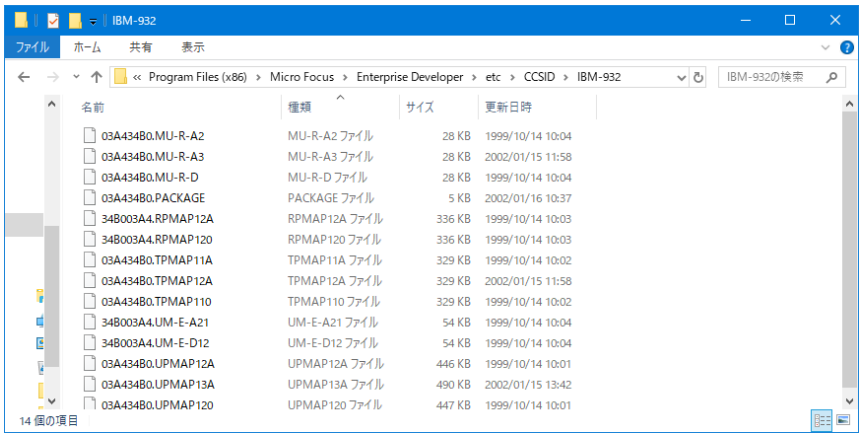
3.7.2 製品インストールフォルダ配下の etc フォルダに CCSID フォルダがない場合はこれを作成します。

例)C:¥Program Files (x86)¥Micro Focus¥Enterprise Developer¥etc¥CCSID

3.7.3 ダウンロードファイルに含まれている Package2.zip を展開します。

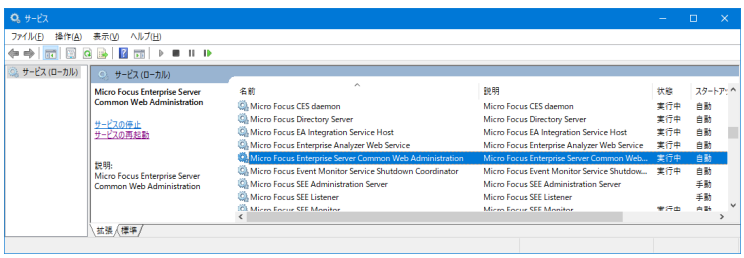
3.7.4 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

3.7.5 展開した IBM-932 フォルダを切り取り、作成した CCSID フォルダ配下へ貼り付け、14 ファイルが含まれていることを確認します。



詳細については、製品マニュアルの「ディプロイ > 構成および管理 > Enterprise Server の構成および管理 > Enterprise Server Common Web Administration > [Native] > [Directory Servers] > リージョンとサーバー > リージョン > エンタープライズ サーバー リージョンの文字エンコーディングのサポート」をご参照ください。

3.7.6 Windows サービスとして起動している Micro Focus™ Enterprise Server Common Web Administration を再起動し、インストールした CCSID をロードさせます。



3.8 Enterprise Server インスタンスの設定

Enterprise Developer は JES のエミュレーション機能を搭載している開発用 Enterprise Server インスタンスを内包しており、各開発者がこのインスタンスを占有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うことができます。本番環境にはコンパイラーなどを含まない実行環境製品 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。

3.8.1

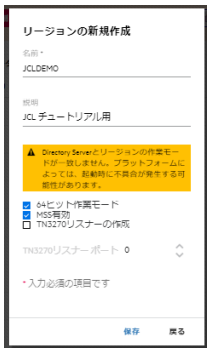
実行する開発用 Enterprise Server インスタンスを作成します。Eclipse の［サーバー エクスプローラー］タブの［ローカル］>［Default］を右クリックして［管理ページを開く］を選択します。Default に登録されているインスタンスが表示エラーになる場合は、Windows の Micro Focus™ Directory Server サービスが開始されているか確認し、停止している場合は開始してください。

3.8.2

ブラウザが立ち上がり ESCWA が表示されます。画面の中央にある［新規作成］ボタンをクリックします。

3.8.3

［リージョンの新規作成］項目の［名前］、［説明］は任意ですが、ここでは名前に JCLDEMO、説明に JCL チュートリアル用と入力します。 Eclipse の実行可能ファイルは 64 ビットを指定してコンパイルしたため、稼働させる Enterprise Server インスタンスも同様に［64 ビット作業モード］へチェックを入れます。これにより警告が表示されますが無視して先に進んでください。［MSS 有効］にチェックが入っていることを確認し、［TN3270 リスナーの作成］のチェックを外して［保存］ボタンをクリックします。



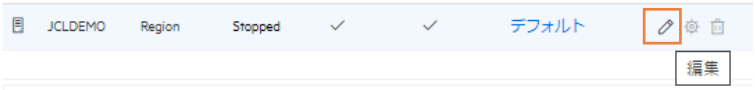
重要

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。

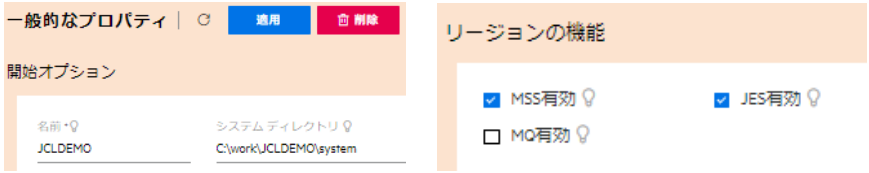
3.8.4

64 ビットアプリケーション稼働用の JCLDEMO インスタンスが作成され、一覧に表示されます。

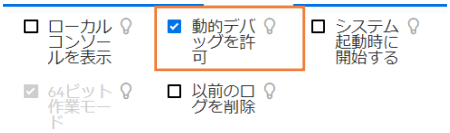
3.8.5 JCLDEMO インスタンスにカーソルを合わせ、[編集] アイコンをクリックします。



3.8.6 JCLDEMO インスタンスのログなどが出力される [システムディレクトリ] には前項で作成した system フォルダを指定して、[リージョンの機能] の [JES 有効] をチェックします。指定後は [適用] ボタンをクリックします。



3.8.7 表示画面の下にある [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。

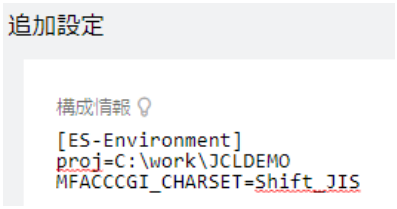


3.8.8 [追加設定] の [構成情報] 欄に、文字エンコーディングを指定する MFACCCGI_CHARSET 環境変数に IBM-932 を認識させるための値である Shift_JIS と、プロジェクトのパスを指定する環境変数を設定し、最後に [適用] ボタンをクリックします。

入力値）

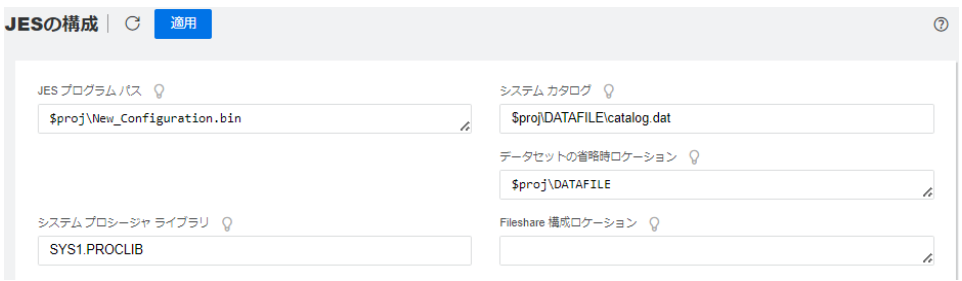
[ES-Environment]

proj=C:¥work¥JCLDEMO
MFACCCGI_CHARSET=Shift_JIS



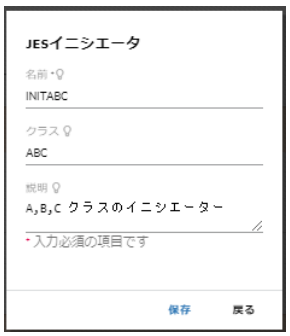
3.8.9 画面上部の [JES] プルダウンメニューから [構成] を選択し、表示される画面の各項目を設定します。構成情報に指定した proj 環境変数を使用して値を入力後、[適用] ボタンをクリックします。

項目名	説明
JES プログラム パス	COBOL アプリケーションの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	カタログファイルを出力するパスと、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。



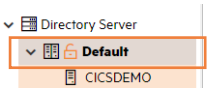
3.8.10 ［イニシエータ］の［新規作成］ボタンをクリックします。

3.8.11 下記画面のように入力して［保存］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。



3.8.12 セキュリティ観点から、Web リスナーのデフォルトステータスは［Disabled］になっています。安全を確認したうえで、［一般］プルダウンメニューから［リスナー］を選択し、表示された Web リスナーのステータスを［Stopped］へ変更後、［適用］ボタンをクリックします。

3.8.13 画面左側ペインの［Default］をクリックして一覧画面に戻ります。



重要

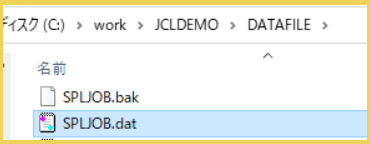
バージョン 7.0 から、JES 関連ファイルである SPLJOB.DAT のフォーマットが改善されています。そのため、旧バージョンのファイルを 7.0 以降で利用する場合は mfsplcnv コマンドを使用して新フォーマットにコンバートする必要があります。コンバートを実行すると、古いフォーマットのファイルは SPLJOB.bak として保存されます。

対象ファイルの特定には MFSYSCAT 環境変数を利用して、カタログファイルを指定します。

例）

set MFSYSCAT=C:¥work¥JCLDEMO¥DATAFILE¥catalog.dat

mfsplcnv -2

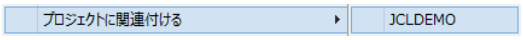


詳しくは製品マニュアルをご参照ください。

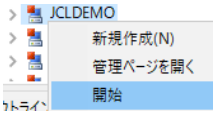
3.9 Enterprise Server インスタンスの開始と確認

3.9.1 Eclipse に戻り、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［Default］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。

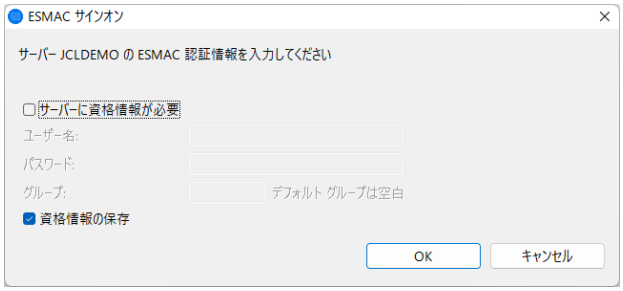
3.9.2 サーバーエクスプローラー内の JCLDEMO インスタンスを右クリックし、［プロジェクトに関連付ける］＞［JCLDEMO］を選択します。これにより JCLDEMO プロジェクトから実行されるアプリケーションは JCLDEMO インスタンスで処理されることになります。



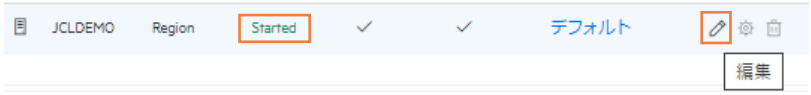
3.9.3 JCLDEMO インスタンスを右クリックして［開始］を選択します。



3.9.4 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。

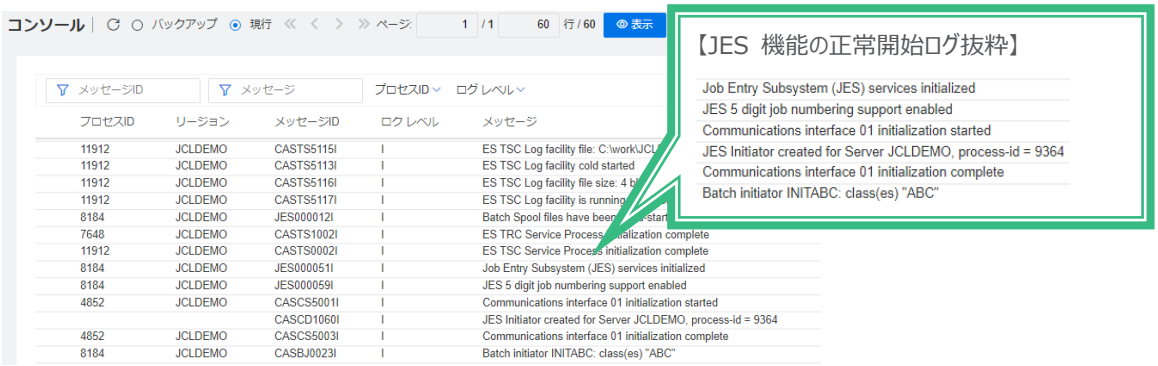


3.9.5 ESCWA へ移動して開始状態であることを確認後、［編集］アイコンをクリックします。



3.9.6 画面上部の［モニター］プルダウンメニューから［ログ］＞［コンソールログ］を選択し、正常に開始されたことを確認します。

ログレベルが I はインフォメーション、S や E の場合はエラー表示されます。



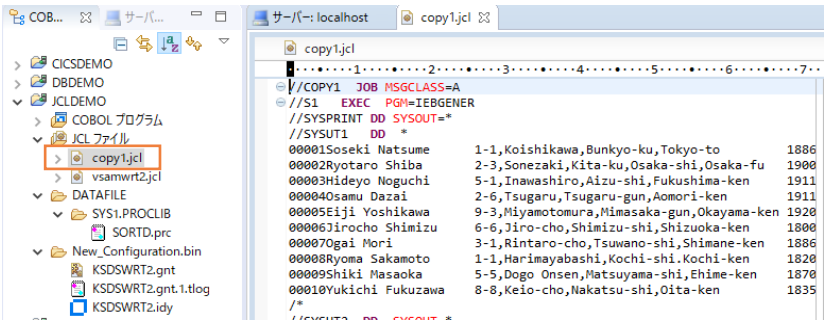
注意

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。

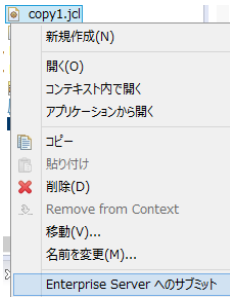
3.10 JCL の実行

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。

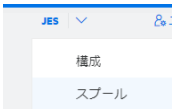
3.10.1 COBOL エクスプローラー内にある JCLDEMO プロジェクト配下の copy1.jcl をダブルクリックし、エディタで内容を確認します。この JCL は IEBGENER ユーティリティを使用して、JCL 内に書かれたインラインデータを SYSOUT に書き出しています。



3.10.2 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。

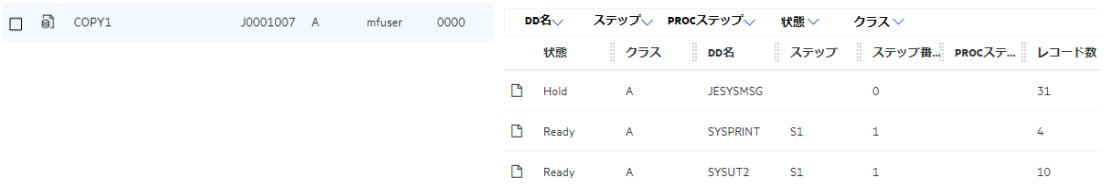


3.10.3 ESCWA からスプールを確認します。JCLDEMO インスタンスを選択後、［JES］プルダウンメニューから［スプール］を選択します。



3.10.4 フィルタ機能で［完了］が指定されていることを確認後、［リスト］ボタンをクリックして一覧を表示します。

3.10.5 実行した JOB 番号のスプールをダブルクリックして内容を表示します。



3.10.6 先頭の［JESYSMEG］をダブルクリックしてジョブログを確認すると、正常に終了していることが確認できます。

```
---> 14:25:30 JCLCM0191I STEP ENDED     S1 - COND CODE 0000

---> 14:25:30 JCLCM0182I JOB  ENDED    - COND CODE 0000
```

3.10.7 右上にある［戻る］ボタンをクリックしてスプール一覧に戻り、[SYSPRINT］をダブルクリックすると、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。

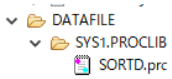
```
      Micro Focus  MFJGENER Utility Version ED8.0_005
      Copyright (C) Micro Focus 1997-2020. All rights reserved.

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

3.10.8 右上にある［戻る］ボタンをクリックしてスプール一覧に戻り、［SYSUT2］をダブルクリックすると、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          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,Miyamotomura,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 プロシージャライブラリの作成

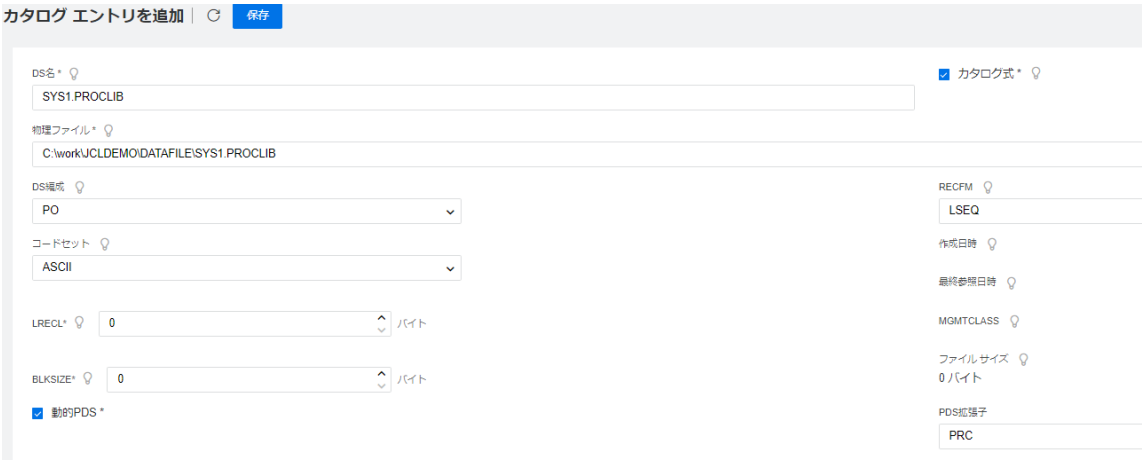
プロシージャを使用する JCL を実行するために、プロシージャライブラリを作成します。Enterprise Server インスタンスではプロシージャを区分データセットのメンバーとして配置します。前項で作成したプロシージャライブラリとなるフォルダをカタログします。



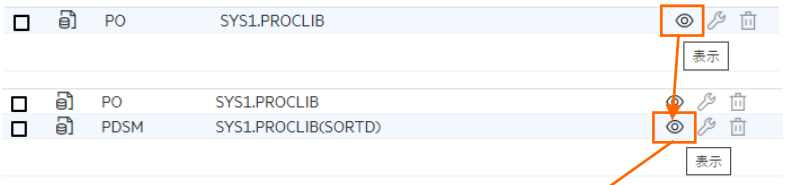
3.11.1 ESCWA へ移動して JCLDEMO インスタンスの画面上部にある［JES］プルダウンメニューの［カタログ］を選択し、[リスト］ボタンをクリックすると、現時点では何も登録されていないことがわかります。新しくカタログするために［新規作成］ボタンをクリックします。

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

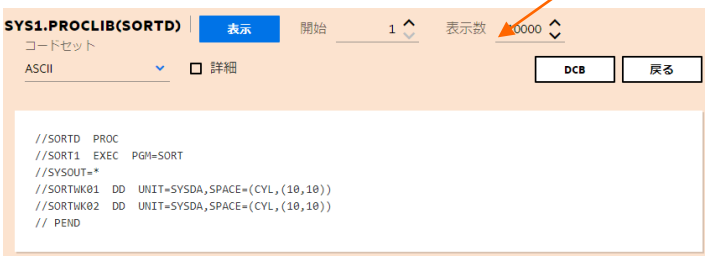
項目名	説明
DS 名	SYS1.PROCLIB を入力します。
物理ファイル	物理パスを指定します。ここでは前項で作成したフォルダパスを入力します。
DS 編成	区分データセットである PO を選択します。
RECFM	行順編成ファイルである LSEQ を選択します。
動的 PDS	プロシージャファイルをフォルダ配下に保持する動的 PDS の場合にオンにします。ここではオンを指定します。
PDS 拡張	プロシージャファイル拡張子を指定します。ここでは PRC を入力します。



3.11.3　カタログ一覧に戻り［リスト］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。



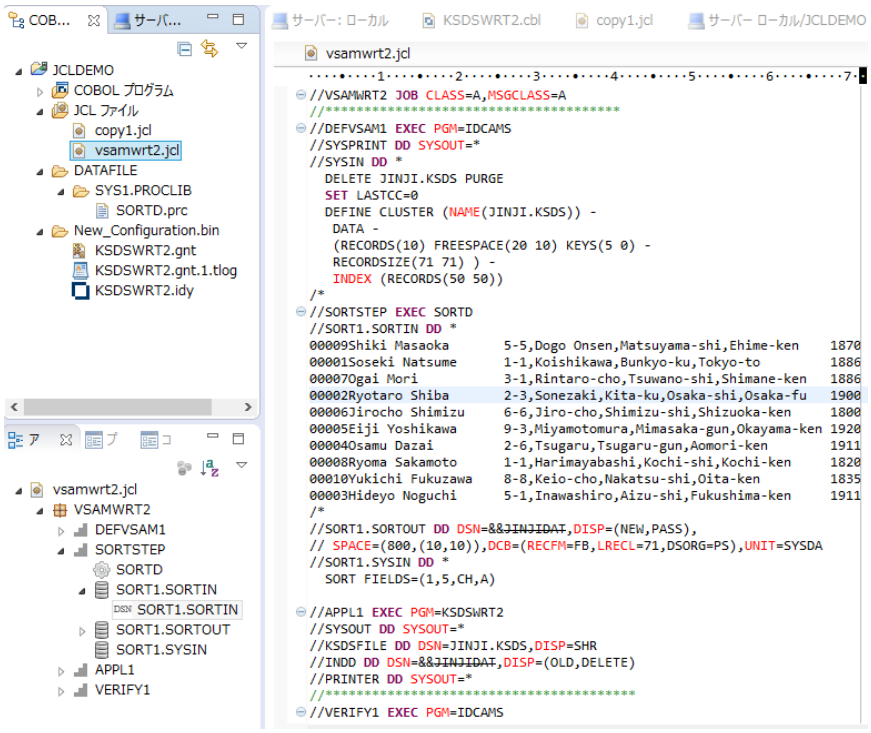
［SYS1.PROCLIB(SORTD)］の［表示］アイコンをクリックすると内容が確認できます。



3.12 COBOL バッチプログラムの実行

Eclipse に戻り、COBOL プログラムを含む JCL を実行してみます。

3.12.1 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

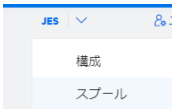
アプリケーション KSDSWRT2 を呼び出しステップ 2 でソートされたデータをステップ 1 で定義した VSAM クラスターに書き込みを行います。同時に書き込まれたデータを DD=PRINTER へ出力します。

④ ステップ 4:VERIFY1

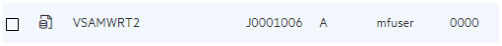
出力内容確認のため、IDCAMS の REPRO で内容を出力します。

3.12.2 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、この JCL を実行します。

3.12.3 ESCWA で JCLDEMO インスタンスを選択後、[JES] プルダウンメニューから［スプール］を選択します。



3.12.4 実行した JOB 番号のスプールをダブルクリックして内容を表示します。



DD名	ステップ	PROCステップ	状態	クラス		
状態	クラス	DD名	ステップ	ステップ番...	PROCステ...	レコード数
Hold	A	JESYSMSG		0		76
Ready	A	SYSPRINT	DEFVSAM1	1		18
Ready	A	SYSOUT	SORTSTEP	2	SORT1	12
Ready	A	SYSOUT	APPL1	3		1
Ready	A	PRINTER	APPL1	3		10
Ready	A	SYSPRINT	VERIFY1	4		41

初回はステップ１で削除するファイルが見つからないために［COND CODE］へ［0008］が返却されますが、次回以降は［0000］が返却されます。

3.12.5 ［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。

```
 ---> 13:23:50 JCLCM0191I STEP ENDED     STEP04 - COND CODE 0000

 ---> 13:23:50 JCLCM0182I JOB  ENDED    - COND CODE 0000
```

3.12.6 右上にある［戻る］ボタンをクリックしてスプール一覧に戻り、DEFVSAM1 ステップの［SYSPRINT］をダブルクリックして内容を確認します。

```
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]
```

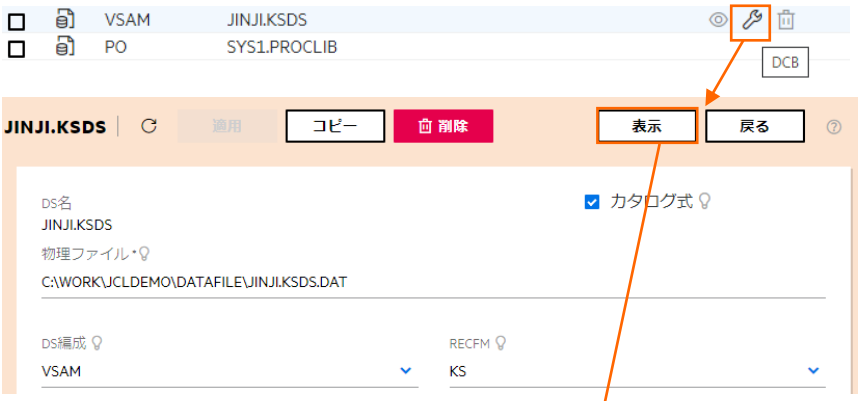
3.12.7 右上にある［戻る］ボタンをクリックしてスプール一覧に戻り、SORTSTEP の［SYSOUT］をクリックしてソート内容を確認します。

```
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
```

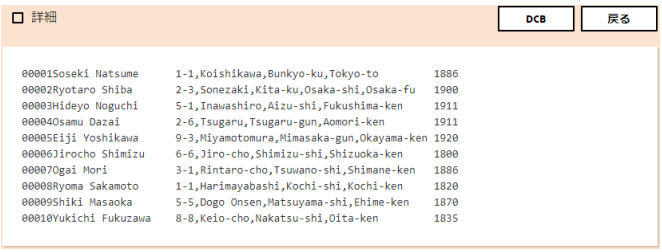
3.12.8 右上にある［戻る］ボタンをクリックしてスプール一覧に戻り、他ステップに関しても確認してみてください。

3.12.9 ESCWA へ移動し、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［リスト］ボタンをクリックすると、VSAM ファイルの JINJI.KSDS が新たにカタログされていることが確認できます。

このカタログ情報にカーソルを合わせ、[DCB] アイコンをクリックすると登録情報が表示されますので、内容を確認します。



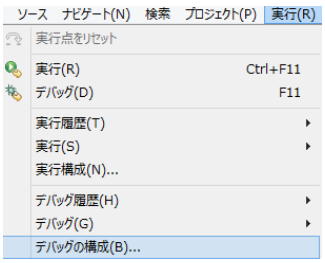
また、一覧の［表示］アイコンもしくは DCB 情報の［表示］ボタンをクリックすると、ファイルの内容が表示されます。



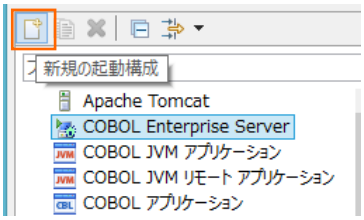
3.13 COBOL バッチプログラムのデバッグ

Eclipse に戻り、JCL から実行される COBOL プログラムをデバッグします。

3.13.1 ［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。

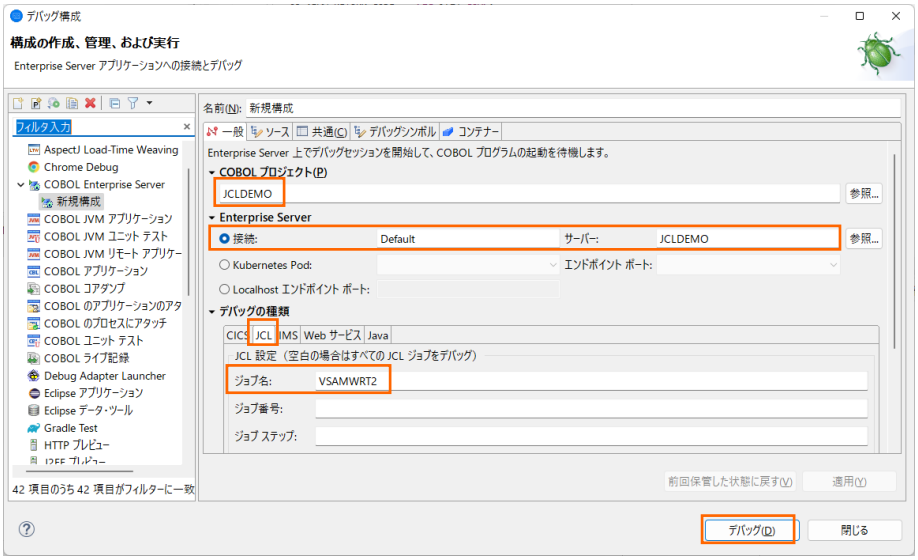


3.13.2 左側のメニューから［COBOL Enterprise Server］を選択して、左上の［新規の起動構成］アイコンをクリックします。

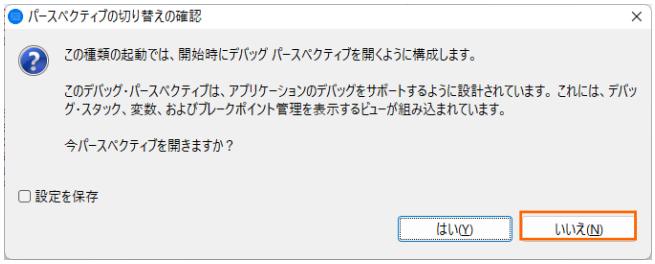


3.13.3 [COBOL プロジェクト] へ対象となる JCLDEMO プロジェクトを入力し、[Enterprise Server] へ実行させる JCLDEMO インスタンスを指定します。

[デバッグの種類] は「JCL」タブを選択した状態で、[ジョブ名] に VSAMWRT2 を入力し、[デバッグ] ボタンをクリックします。



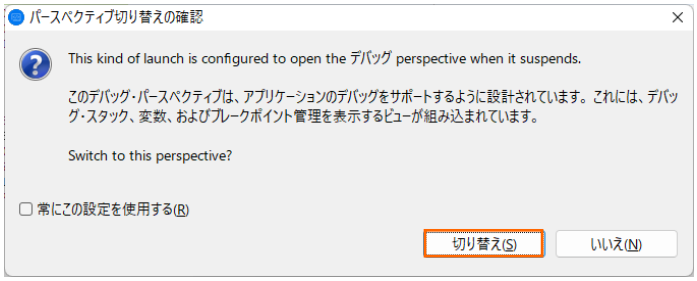
3.13.4 パースペクティブの切り替え確認ウィンドウが表示されますが、COBOL エクスプローラーから JCL を実行するため、ここでは［いいえ］ボタンをクリックします。



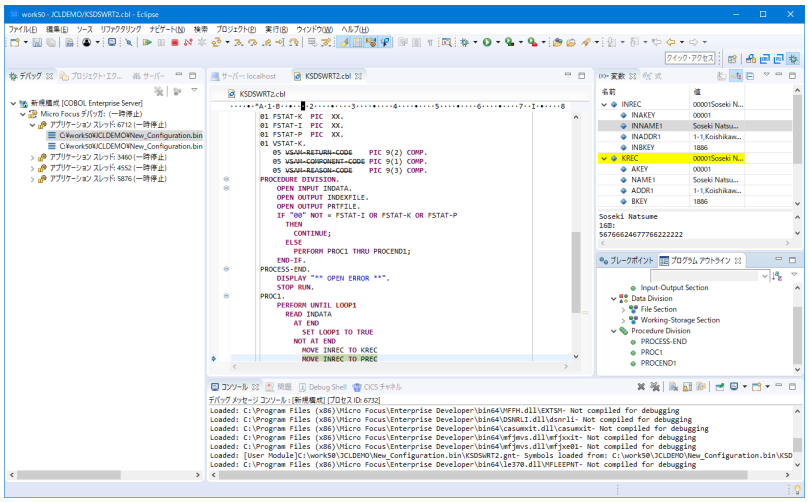
3.13.5 デバッグタブで［アタッチ待機］状態になったことを確認します。

3.13.6 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。

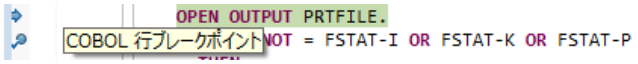
3.13.7 再度、パースペクティブの切り替え確認ウィンドウが表示されますので、ここでは［切り替え］ボタンをクリックし、デバッグ用のパースペクティブを開きます。



3.13.8 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] プ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。

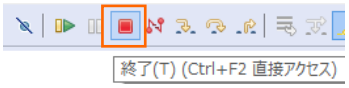


3.13.9 希望のステップの左端をダブルクリックすることにより、ブレークポイントを設定することも可能です。



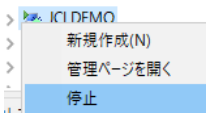
3.13.10 先に進める場合は画面上部の再開アイコンをクリックします。

3.13.11 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。



3.14 Enterprise Server インスタンスの停止

3.14.1 COBOL パースペクティブへ戻り、JCLDEMO インスタンスを停止します。



3.14.2 ESCWA から JCLDEMO インスタンスの停止を確認後、Eclipse を終了します。

4　免責事項

本チュートリアルの例題ソースコードは機能説明を目的としたサンプルであり、無謬性を保証するものではありません。例題ソースコードは弊社に断りなくご利用いただけますが、本チュートリアルに関わる全てを対象として、二次的著作物に引用する場合は著作権法の精神に基づき適切な扱いを行ってください。

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。