

# 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

1. 本チュートリアルで使用したマシン OS : Windows 11 Pro
2. 使用マシンに Enterprise Developer 11J 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. テキストファイルのエンコード指定
5. プロジェクトプロパティの設定
6. ビルドの実行
7. 文字エンコーディングの設定
8. Enterprise Server インスタンスの設定
9. Enterprise Server インスタンスの開始と確認
10. JCL の実行
11. プロシージャライブラリの作成
12. COBOL バッチプログラムの実行
13. COBOL バッチプログラムのデバッグ
14. Enterprise Server インスタンスの停止

### 4 免責事項

### 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。



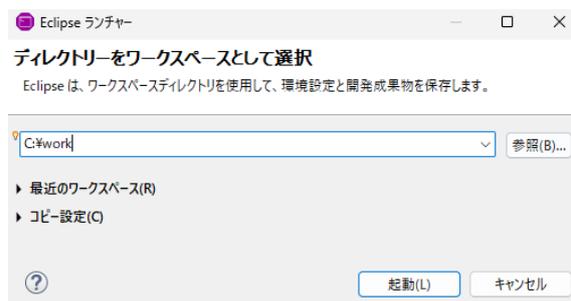
- 2) Eclipse のワークスペースで使用する work フォルダを C:¥直下に作成します。

### 3.2 Eclipse の起動

- 1) メニューから [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、[起動] ボタンをクリックします。



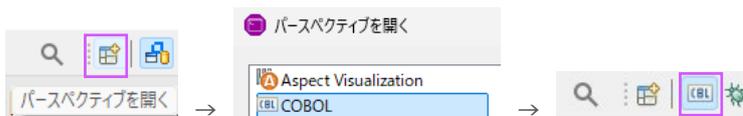
- 3) [ようこそ] タブでは [Open COBOL Perspective] をクリックして、COBOL パースペクティブを開きます。



- 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。

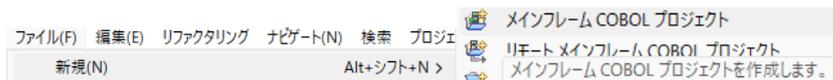


- 5) COBOL パースペクティブが開いていない場合は Eclipse 右上の [パースペクティブを開く] アイコンをクリックして表示後、[COBOL] を選択して [開く] ボタンをクリックします。



### 3.3 メインフレーム COBOL プロジェクトの作成

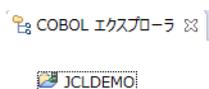
- 1) 用意した例題ソースをインポートします。[ファイル] プルダウンメニューから [新規] > [メインフレーム COBOL プロジェクト] を選択します。



- 2) [プロジェクト名] は任意ですが、ここでは JCLDEMO を入力し、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。

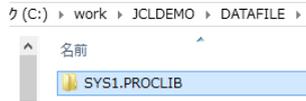


- 3) COBOL エクスプローラーに作成したプロジェクトが表示されます。

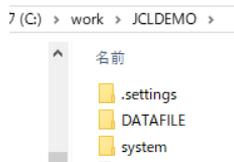


- 4) プロジェクトを作成したことにより C:\work\JCLDEMO フォルダが作成されています。このフォルダ配下に JES 機能で使用するフォルダを、Windows エクスプローラーを使用してあらかじめ用意しておきます。

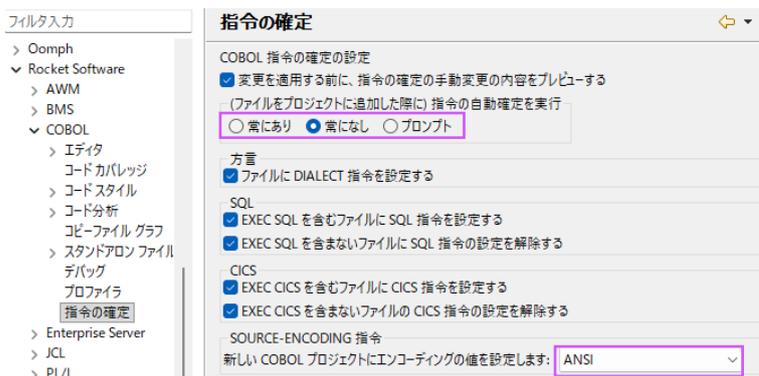
- ・ カタログファイルやスプールファイルを配置するため DATAFILE フォルダを C:\work\JCLDEMO 配下へ作成します。
- ・ プロシージャファイルを配置するため、プロシージャライブラリとして SYS1.PROCLIB フォルダを C:\work\JCLDEMO\DATAFILE 配下へ作成します。



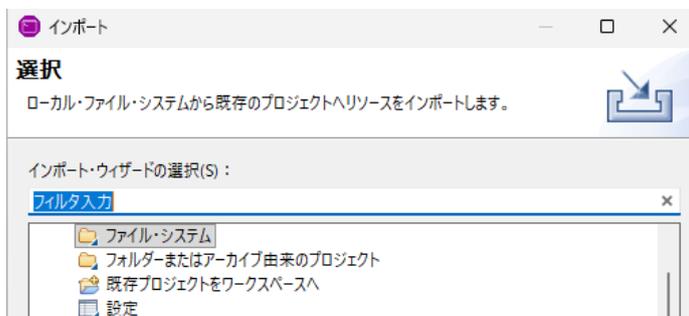
- ・ 実行時に使用する system フォルダを C:\work\JCLDEMO 配下へ作成します。



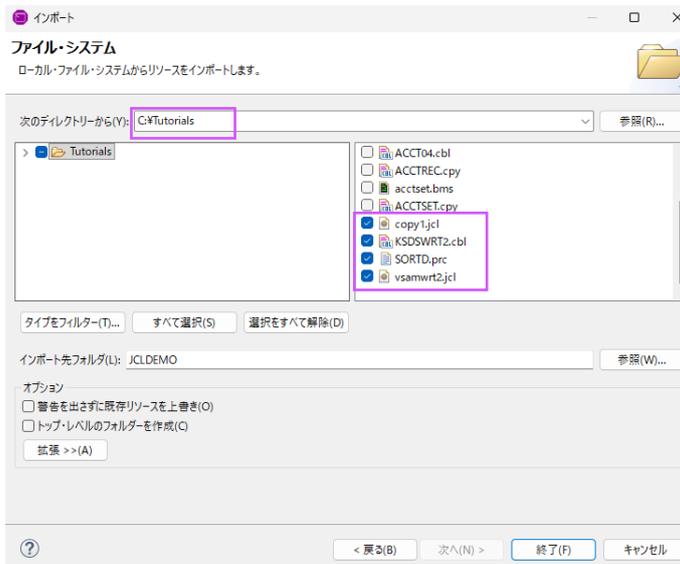
- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ウィンドウ] プルダウンメニューの [設定] > [Rocket Software] > [COBOL] > [指令の確定] > [COBOL 指令の確定の設定] では [常になし] を選択し、[SOURCE-ENCODING 指令] では [ANSI] を選択後、[適用して閉じる] ボタンをクリックします。



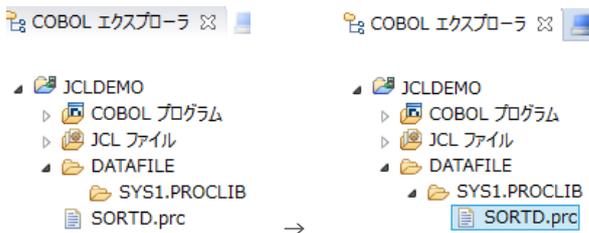
- 6) 用意した例題プログラム類をインポートします。JCLDEMO プロジェクトを右クリックして [インポート] > [インポート] を選択し、インポートウィンドウにて [一般] > [ファイル・システム] を選択後 [次へ] ボタンをクリックします。



- 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。

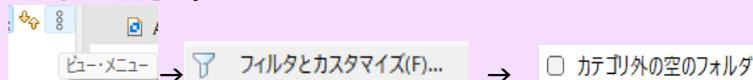


- 8) COBOL エクスプローラー内に表示されている JCLDEMO プロジェクトにインポートしたファイルが表示されていることを確認後、作成した SYS1.PROCLIB フォルダへ SORTD.prc ファイルをドラッグ&ドロップして移動します。



### 注意

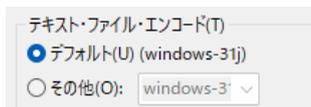
フォルダ内が空のため作成したフォルダが表示されない場合があります。その際は COBOL エクスプローラー 右上の [ビュー・メニュー] > [フィルタとカスタマイズ] を選択後 [カテゴリ外の空のフォルダ] のチェックをオフにしてください。



## 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 Windows-31j を使用します。

- 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。
- 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 または Windows-31j が指定されていることを確認します。「デフォルト」がこの値ではない場合は、「その他」を選択して、これらを指定してください。



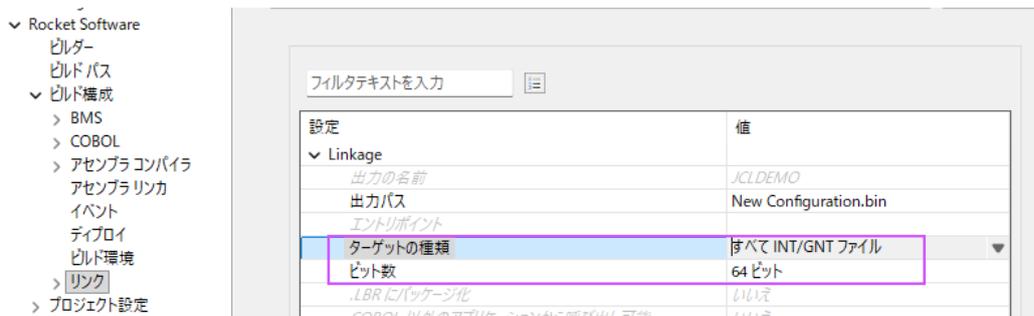
- 3) 指定後は [適用して閉じる] ボタンをクリックします。

### 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。

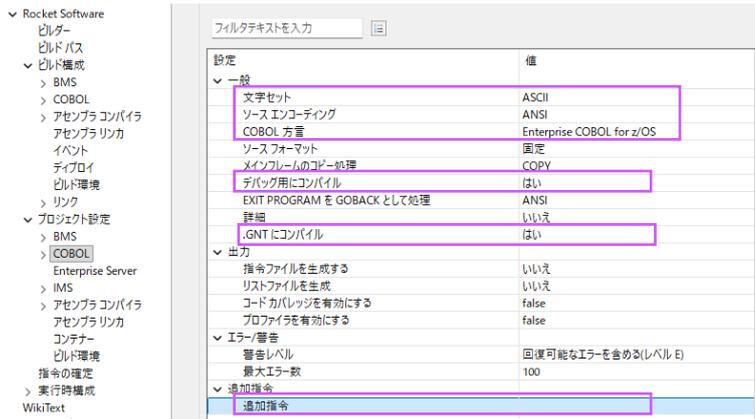
- 1) COBOL エクスプローラー内の JCLDEMO プロジェクトを右クリックして [プロパティ] を選択します。
- 2) 左側ツリービューの [Rocket Software] > [ビルド構成] > [リンク] を選択して、下記項目を指定します。指定後は [適用] ボタンをクリックしてください。

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 [すべて INT/GNT ファイル] を選択します。
ビット数	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。



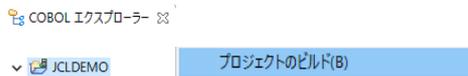
- 3) 左側ツリービューの [Rocket Software] > [プロジェクト設定] > [COBOL] を選択して、下記項目を指定します。指定後は [適用して閉じる] ボタンをクリックしてください。

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。MS932 または Windows-31j と一致するように 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。

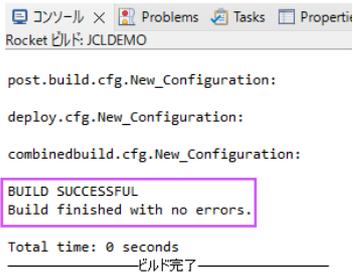


### 3.6 ビルドの実行

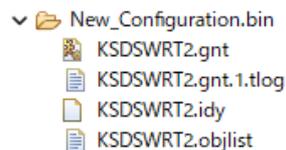
- 1) COBOL エクスプローラー内のプロジェクトを右クリックして [プロジェクトのビルド] を選択するとビルドが実行されます。



- 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。



### 3.7 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。

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。

<https://download.boulder.ibm.com/ibmdl/pub/software/dw/java/cdctables.zip>

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)\Rocket Software\Enterprise Developer\etc\ccsid

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。

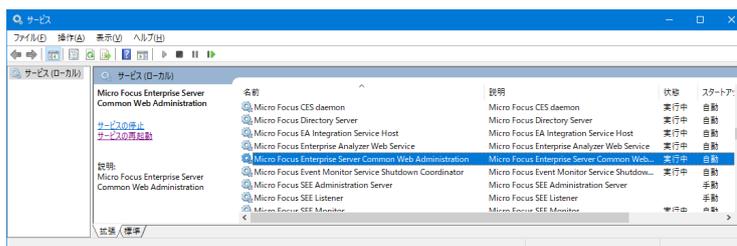
4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

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。

名前	種類	サイズ
03A434B0.MU-R-A2	MU-R-A2 ファイル	28 KB
03A434B0.MU-R-A3	MU-R-A3 ファイル	28 KB
03A434B0.MU-R-D	MU-R-D ファイル	28 KB
03A434B0.PACKAGE	PACKAGE ファイル	5 KB
03A434B0.TPMAP11A	TPMAP11A ファイル	329 KB
03A434B0.TPMAP12A	TPMAP12A ファイル	329 KB
03A434B0.TPMAP110	TPMAP110 ファイル	329 KB
03A434B0.UPMAP12A	UPMAP12A ファイル	446 KB
03A434B0.UPMAP13A	UPMAP13A ファイル	490 KB
03A434B0.UPMAP120	UPMAP120 ファイル	447 KB
34B003A4.RPMAP12A	RPMAP12A ファイル	336 KB
34B003A4.RPMAP120	RPMAP120 ファイル	336 KB
34B003A4.UM-E-A21	UM-E-A21 ファイル	54 KB
34B003A4.UM-E-D12	UM-E-D12 ファイル	54 KB

詳細については、製品マニュアルの「デプロイ > 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。

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。

- 1) ESCWA は、製品が提供する VSAM 外部セキュリティマネージャー (ESM) モジュールによるセキュリティがデフォルトで有効になっており、すべての処理において実行ユーザーの認証が行われます。

まずはデフォルトユーザーと初期パスワードを取得するために、スタートメニューから [Enterprise Developer] を選択し、[Enterprise Developer コマンドプロンプト] を起動します。



Enterprise Developer コマンドプロンプトから次のコマンドを実行して、デフォルトユーザーと初期パスワードを取得します。

```
mfsecretsadmin read microfocus/temp/admin
```

```
C:\Users\%tarot%\Documents>mfsecretsadmin read microfocus/temp/admin
["mfUser": "SYSAD", "mfPassword": "CKfa+xWt"]
```

上記の場合、SYSAD がデフォルトユーザー、CKfa+xWt が初期パスワードです。

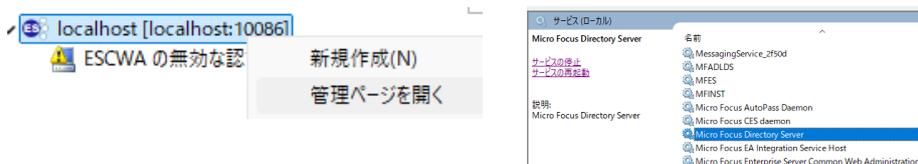
この情報は ESCWA のログオン時に使用しますので、記憶しておいてください。

デフォルトセキュリティを無効にすることもできますが、安全を確認後に実施してください。

詳しくは製品マニュアルの [ここからはじめよう] > [Getting Started] にある [デフォルトセキュリティの構成] チュートリアルをご参照ください。

- 2) 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 [localhost] を右クリックして [管理ページを開く] を選択します。表示されていない場合は、Windows サービスの [Micro Focus Directory Server] が開始されているかを確認し、停止している場合は開始してください。



- 3) Web ブラウザが立ち上がり、ユーザー認証を求められます。

パスワードを変更してログオンします。

[パスワード変更] をクリックし、前項で取得したデフォルトユーザーと初期パスワード、新しいパスワードを入力して [サブミット] をクリックします。

変更後のパスワードはご自身の責任で管理してください。

パスワード変更後、ESCWA がタイムアウトした場合は新しいパスワードを使用してログオンしてください。

**Enterprise Server Administration**

Rocket Software Enterprise Serverでは、インストール後に基本的なセキュリティ機能がデフォルトで有効になっています。

[詳細情報](#)

ユーザー名

パスワード

パスワード変更 ログオン

詳細情報は、次のセキュリティ マネージャを使用  
して検証されます: VSAM ESM

Rocket software

**Enterprise Server Administration**

Rocket Software Enterprise Serverでは、インストール後に基本的なセキュリティ機能がデフォルトで有効になっています。

[詳細情報](#)

ユーザー名

パスワード

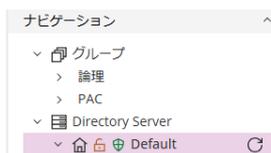
新しいパスワード\*  パスワードの確認\*

キャンセル サブミット

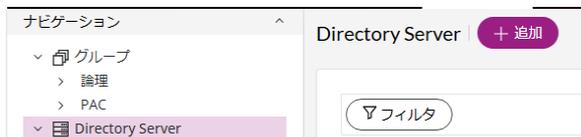
詳細情報は、次のセキュリティ マネージャを使用  
して検証されます: VSAM ESM

Rocket software

- 4) メニューで [オペレーション] を選択後、[ナビゲーション] に [Default] という名前の接続が表示されます。表示されない場合は新規に作成します。表示されている場合はこの項目はスキップしてください。



左側ペインで [Directory Server] をクリックし、右側ペインの [追加] ボタンをクリックします。



追加ウィンドウの [名前] は任意ですが、ここでは Default を、  
[ホスト] は ホスト名またはマシンの IP アドレスを、  
[ポート] は Micro Focus Directory Server の  
デフォルトポートである 86 を入力して [保存] ボタンをクリックします。

**Directory Server**

名前\*

ホスト\*

ポート\*

説明

\* 入力必須の項目です

保存 戻る

- 5) Enterprise Server インスタンスを追加します。  
[Default] をクリックし、右側ペインの [新規作成] ボタンをクリックします。



- 6) [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] ヘ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。

#### リージョンの新規作成

Directory Serverとリージョンの作業モードが一致しません。プラットフォームによっては、起動時に不具合が発生する可能性があります。

名前\*

説明

64ビット作業モード

MSS有効

TN3270リスナーの作成

TN3270リスナー ポート

\*入力必須の項目です 保存 戻る

#### 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。

- 7) 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。

ア...	名前	タ...	ステータス	64ビット	MSS有効	セキュ...	アクション
	JCLDEMO	Region	Stopped	✓	✓	デフォルト	✎ ⚙️ 🗑️

- 8) 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。

	JCLDEMO	Region	Stopped	✓	✓	デフォルト	✎ ⚙️ 🗑️
							編集

- 9) 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。

開始オプション		リージョンの機能	
名前* <input type="text" value="JCLDEMO"/>	システムディレクトリ <input type="text" value="C:\work\JCLDEMO\system"/>	<input checked="" type="checkbox"/> MSS有効	<input checked="" type="checkbox"/> JES有効
		<input type="checkbox"/> IMS有効	<input type="checkbox"/> MQ有効

- 10) 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。指定後は [適用] ボタンをクリックします。

ローカル コンソールを表示

動的デバッグを許可

64ビット作業モード

以前のログを削除

システム起動時に開始する

- 11) [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

### 追加設定

構成情報 ⓘ

```
[ES-Environment]
proj=C:\work\JCLDEMO
MFACCCGI_CHARSET=Shift_JIS
```

- 12) 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。

**JESの構成** 適用

---

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 ⓘ

- 13) [イニシエータ] の [新規作成] ボタンをクリックします。

イニシエータ | \* 新規作成

- 14) 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。

**JESイニシエータ**

名前\* ⓘ

クラス ⓘ

説明 ⓘ

\* 入力必須の項目です 保存 戻る

- 15) 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。



- 16) 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

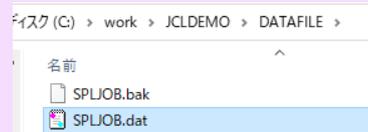
バージョン 7.0 から、JES 関連ファイルである SPLJOB.DAT のフォーマットが改善されています。そのため、旧バージョンのファイルを 7.0 以降で利用する場合は mfsplcnv コマンドを使用して新フォーマットにコンバートする必要があります。コンバートを実行すると、古いフォーマットのファイルは SPLJOB.bak として保存されます。

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
```

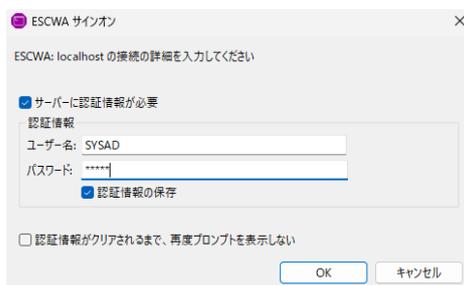
```
mfsplcnv -2
```



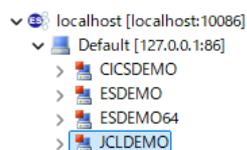
詳しくは製品マニュアルをご参照ください。

## 3.9 Enterprise Server インスタンスの開始と確認

- 1) Eclipse に戻り、サーバーエクスプローラー内の [localhost] をクリックすると、ESCWA への認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。



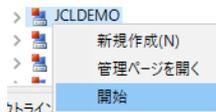
- 2) [Default] を展開して作成した JCLDEMO インスタンスが表示されていることを確認します。表示されていない場合は [Default] を右クリックし、[更新] を選択してリフレッシュしてください。



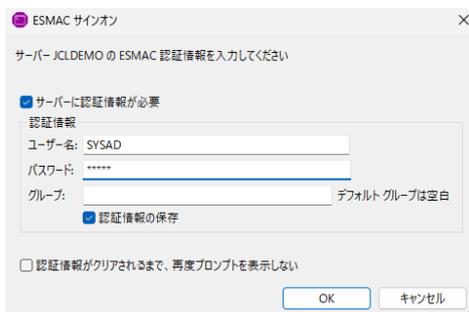
- 3) サーバーエクスプローラー内の JCLDEMO インスタンスを右クリックし、[プロジェクトに関連付ける] > [JCLDEMO] を選択します。これにより JCLDEMO プロジェクトから実行されるアプリケーションは JCLDEMO インスタンスで処理されることになります。



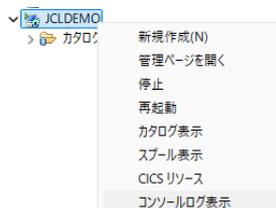
- 4) JCLDEMO インスタンスを右クリックして [開始] を選択します。



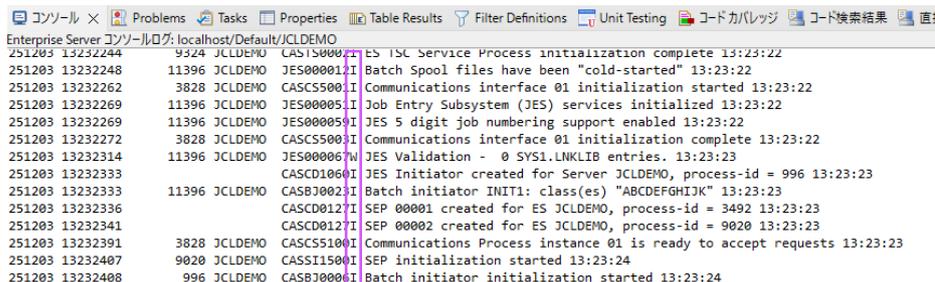
- 5) 開始ユーザーの認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。



- 6) 開始後、JCLDEMO インスタンスを右クリックして [コンソールログ表示] を選択すると、[コンソール] タブに開始ログが表示されます。



メッセージコードの最終桁のログレベルが I はインフォメーション、W は警告、S や E の場合はエラーです。



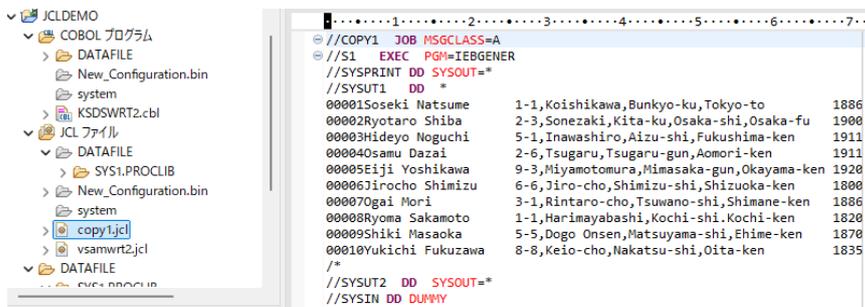
### 注意

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。

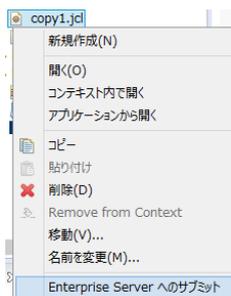
### 3.10 JCL の実行

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。

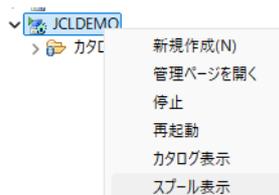
- 1) COBOL エクスプローラー内にある JCLDEMO プロジェクト配下の copy1.jcl をダブルクリックし、エディタで内容を確認します。この JCL は IEBGENER ユーティリティを使用して、JCL 内に書かれたインラインデータを SYSOUT に書き出しています。



- 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) 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、[スプール表示] を選択します。



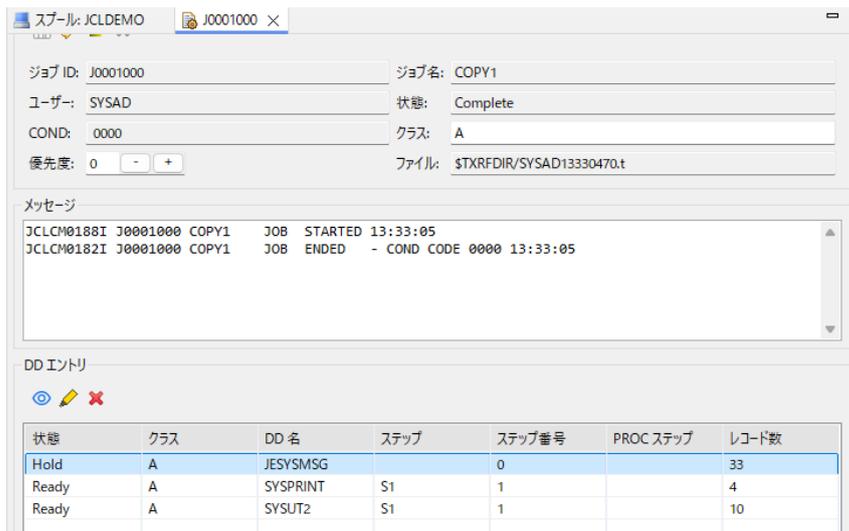
実行された JCL の結果を確認すると、条件コードはゼロで正常終了したことがわかります。

該当行をダブルクリックして詳細を表示します。

スプール: JCLDEMO × J0001000

名前	ジョブ ID	クラス	ユーザー	条件コード
COPY1	J0001000	A	SYSAD	0000

DD エントリをダブルクリックすると内容が確認できます。



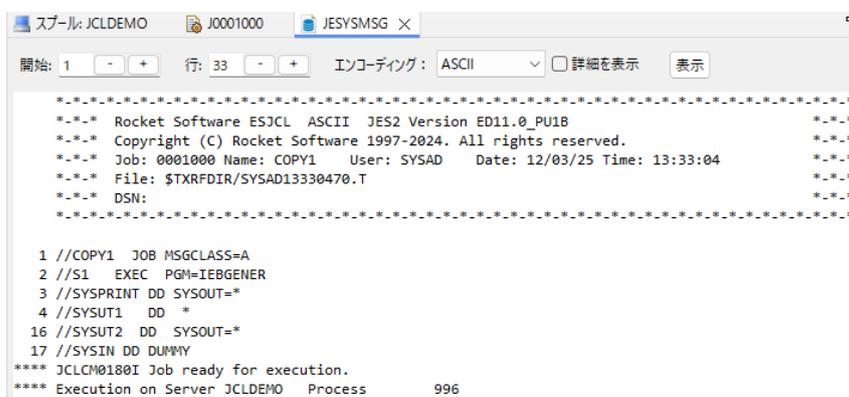
Job ID: J0001000      ジョブ名: COPY1  
 ユーザー: SYSAD      状態: Complete  
 COND: 0000      クラス: A  
 優先度: 0      ファイル: \$TXRFDIR/SYSAD13330470.t

メッセージ

```
JCLCM0188I J0001000 COPY1 JOB STARTED 13:33:05
JCLCM0182I J0001000 COPY1 JOB ENDED - COND CODE 0000 13:33:05
```

DD エントリ

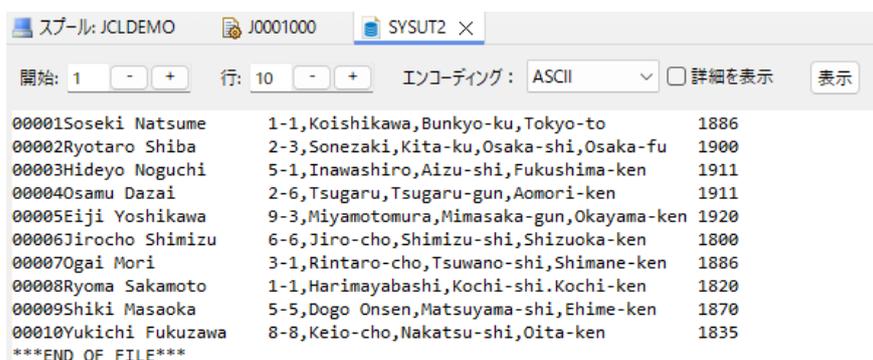
状態	クラス	DD 名	ステップ	ステップ番号	PROC ステップ	レコード数
Hold	A	JESYSMSG		0		33
Ready	A	SYSPRINT	S1	1		4
Ready	A	SYSUT2	S1	1		10



開始: 1      行: 33      エンコーディング: ASCII       詳細を表示      表示

```
*****
***** Rocket Software ESJCL ASCII JES2 Version ED11.0_PU1B *****
***** Copyright (C) Rocket Software 1997-2024. All rights reserved. *****
***** Job: 0001000 Name: COPY1 User: SYSAD Date: 12/03/25 Time: 13:33:04 *****
***** File: $TXRFDIR/SYSAD13330470.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 JCLDEMO Process 996
```



開始: 1      行: 10      エンコーディング: ASCII       詳細を表示      表示

```
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
00007Gai Mori      3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken      1886
00008Ryoma Sakamoto      1-1,Harimayabashi,Kochi-shi,Kochi-ken      1820
00009Shiki Masaoka      5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken      1870
00010Yukichi Fukuzawa      8-8,Keio-cho,Nakatsu-shi,Oita-ken      1835
***END OF FILE***
```

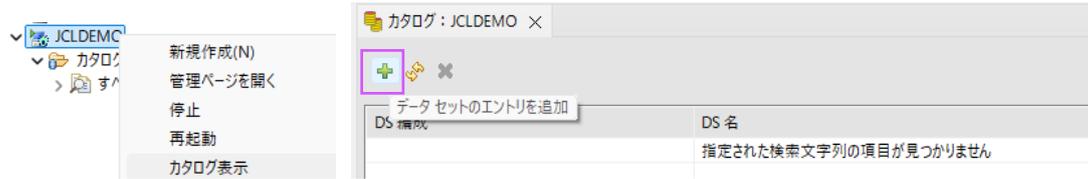
### 3.11 プロシージャライブラリの作成

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。

- ▼ DATAFILE
- ▼ SYS1.PROCLIB
- SORTD.prc

- 1) 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、[カタログ表示] を選択します。現時点では何も登録されていないことがわかります。

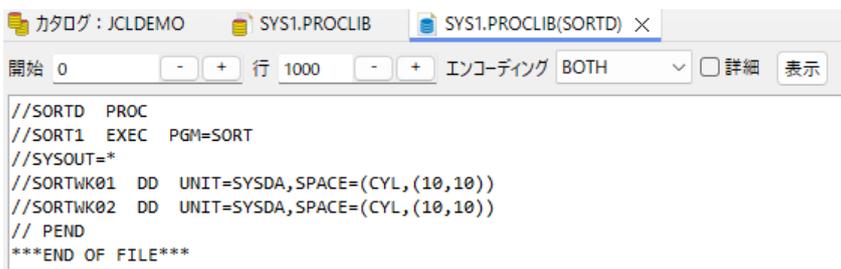
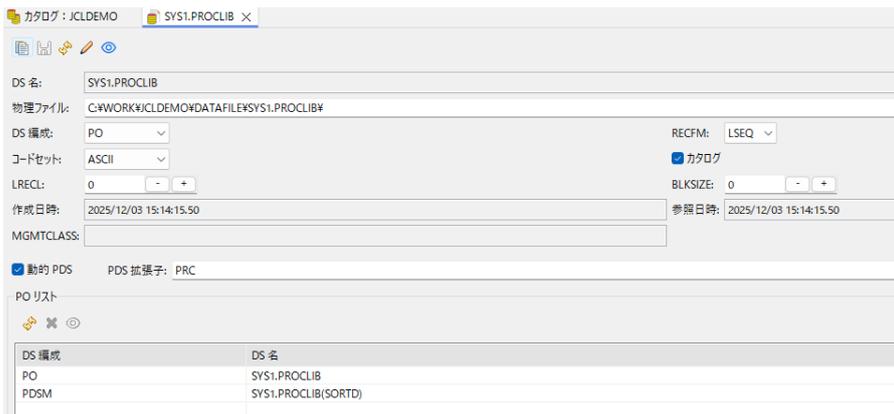
新しくカタログするために [データセットのエントリを追加] アイコンをクリックします。



- 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 [OK] 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。
コードセット	EBCDIC もしくは ASCII を指定します。ここでは ASCII を選択します。
カタログ	チェックを入れます。
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。

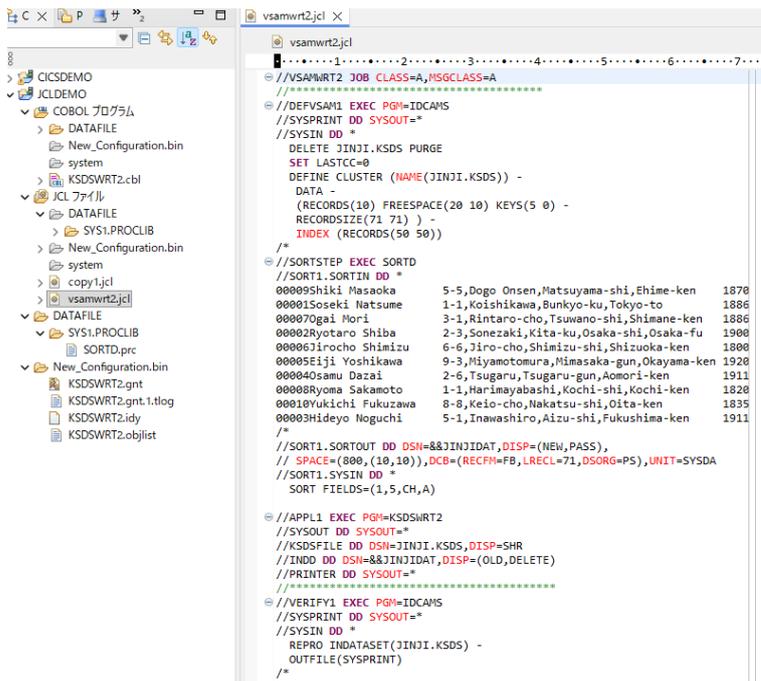
- 3) 登録した内容が表示されます。[DS 構成] の PDSM をダブルクリックすると内容を確認できます。



### 3.12 COBOL バッチプログラムの実行

Eclipse から COBOL プログラムを含む JCL を実行します。

- 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、この JCL を実行します。
  - 3) 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、[スプール表示] を選択します。  
実行された JCL の結果を確認すると、条件コードが 0008 で終了したことがわかります。これは、初回ではステップ 1 で削除するファイルが見つからないためであり、次回以降は 0000 が返却されます。  
該当行をダブルクリックして詳細を表示します。[DD エントリ] の [JESYSMSG] をダブルクリックします。

スプール: JCLDEMO ×

名前	ジョブ ID	クラス	ユーザー	条件コード
COPY1	J0001000	A	SYSAD	0000
VSAMWRT2	J0001001	A	SYSAD	0008

- 4) [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。

```

15:23:32 JCLCM0188I JOB   STARTED

15:23:32 JCLCM0190I STEP STARTED   DEFVSAM1
15:23:32 JCLCM0303I Program IDCAMS (MFJAMS) loaded from library SYS1.LOADLIB
15:23:32 JCLCM0199I Program MFJAMS is COBOL          ASCII Big-Endian  NOAMODE.
Y2025.S1203.S152332.J0001001.D00001.SYSPRINT      SYSPRINT
C:¥WORK¥JCLDEMO¥DATAFILE¥Y2025.S*32.J0001001.D00001.SYSPRINT.DAT  SPOOLED
Y2025.S1203.S152332.J0001001.D00002.SYSIN          SYSIN
C:¥WORK¥JCLDEMO¥DATAFILE¥Y2025.S*52332.J0001001.D00002.SYSIN.DAT  DELETED
--> 15:23:32 JCLCM0191I STEP ENDED   DEFVSAM1 - COND CODE 0008

```

- 5) プロシージャを使用した SORTSTEP の [SYSOUT] をダブルクリックしてソート内容を確認します。

```

Rocket Software 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: Rocket Software MFJSORT ユーティリティ終了
*****END OF FILE**

```

- 6) [DD エントリ] の他ステップについても確認してみてください。

- 7) 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。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。

カタログ : JCLDEMO ×

DS 構成	DS 名
VSAM	JINJI.KSDS
PO	SYS1.PROCLIB

該当行をダブルクリックして DCB 情報を確認します。

カタログ : JCLDEMO JINJI.KSDS ×

DS 名: JINJI.KSDS

物理ファイル: C:\WORK\JCLDEMO\DATAFILE\JINJI.KSDS.DAT

DS 構成: VSAM RECFM: KS

コードセット: ASCII  カタログ

LRECL: 71 BLKSIZE: 0

作成日時: 2025/12/03 15:23:32.41 参照日時: 2025/12/03 15:23:32.81

MGMTCLASS:

VSAM 型: Cluster VSAM 属性: Unique Key

サーバー間: 0 システム間: 0

キー先頭: 0 キーの長さ: 5

キー最大: 71 キー平均: 5

表示アイコンをクリックすると、ファイルのデータが表示されます。

カタログ : JCLDEMO JINJI.KSDS ×

表示

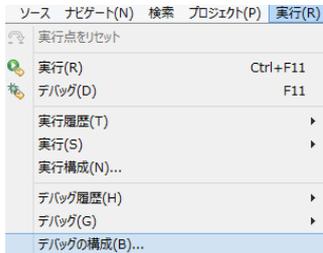
開始 0 行 1000 エンコーディング ASCII  詳細 表示

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

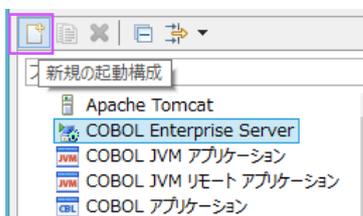
### 3.13 COBOL バッチプログラムのデバッグ

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。



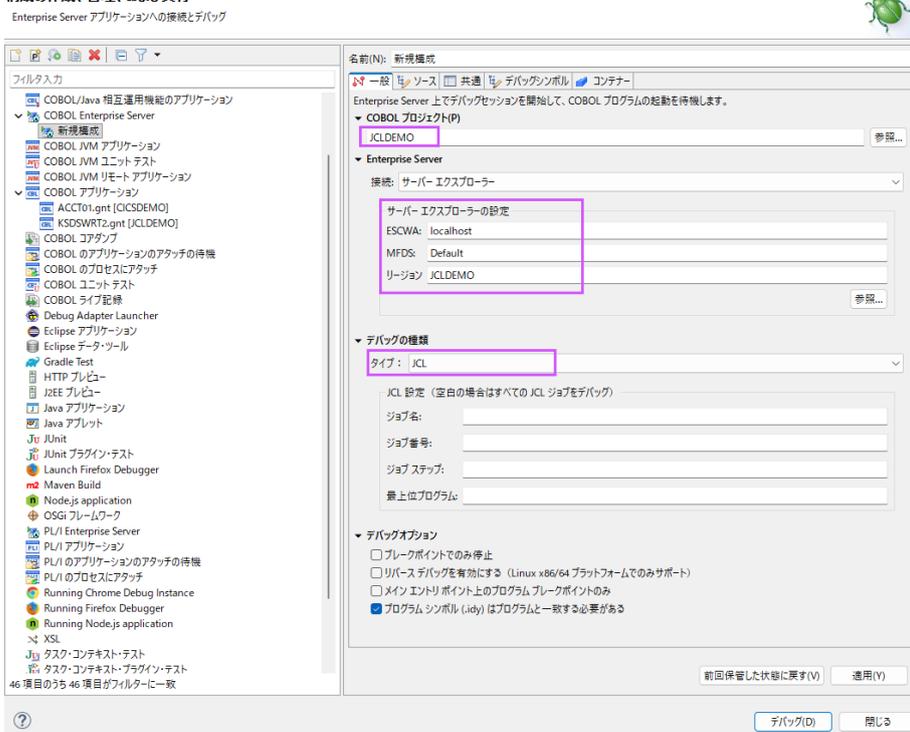
- 2) 左側のメニューから [COBOL Enterprise Server] を選択して、左上の [新規の起動構成] アイコンをクリックします。



- 3) [COBOL プロジェクト] へ対象となる JCLDEMO 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。

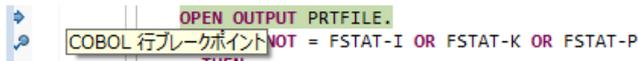
[デバッグの種類] は「JCL」を選択した状態で [デバッグ] ボタンをクリックします。

構成の作成、管理、および実行





- 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。

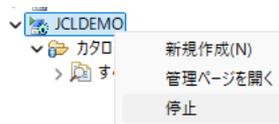


- 11) デバッグを終了させるため、画面上部の終了アイコンをクリックします。



### 3.14 Enterprise Server インスタンスの停止

- 1) COBOL パースパクティブへ戻り、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。

## 4 免責事項

本チュートリアル of 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。