

# Enterprise Developer チュートリアル

## メインフレーム PL/I 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 PL/I プロジェクトの作成、PL/I ソースのコンパイル、JCL の実行、デバッグを行い、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 PL/I プロジェクトのインポート
4. プロジェクトプロパティの確認
5. ビルドの実行
6. 文字エンコーディングの設定
7. Enterprise Serve インスタンスの設定
8. Enterprise Serve インスタンス開始と確認
9. JCL の実行
10. PL/I ソースのデバッグ
11. 終了処理

### 4. 免責事項

### 3.1 チュートリアルの準備

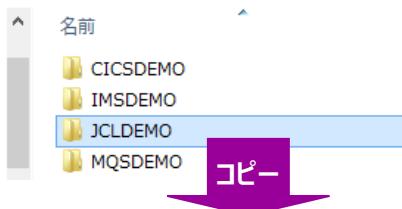
例題プログラムに関連するリソースを用意します。

- 1) Eclipse のワークスペースで使用する C:¥work¥PLI フォルダを作成します。
- 2) Windows のエクスプローラーを使用して、下記のパスに配置されている例題プログラムの JCLDEMO フォルダを、作成した C:¥work¥PLI へコピーします。

例)

C:¥Users¥Public¥Documents¥Rocket Software¥Enterprise Developer¥Samples¥PLI-Eclipse¥JCLDEMO

Enterprise Developer > Samples > PLI-Eclipse >



コピー先)C:¥work¥PLI¥JCLDEMO

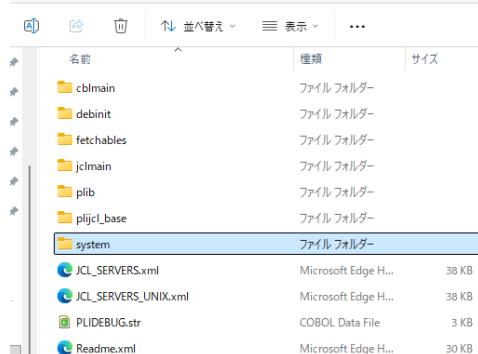
ディスク (C) > work > PLI

名前

JCLDEMO

- 3) C:¥work¥PLI¥JCLDEMO フォルダ配下に、実行時に使用する system フォルダをあらかじめ作成しておきます。

PC > Windows (C:) > work > PLI > JCLDEMO >



### 3.2 Eclipse の起動

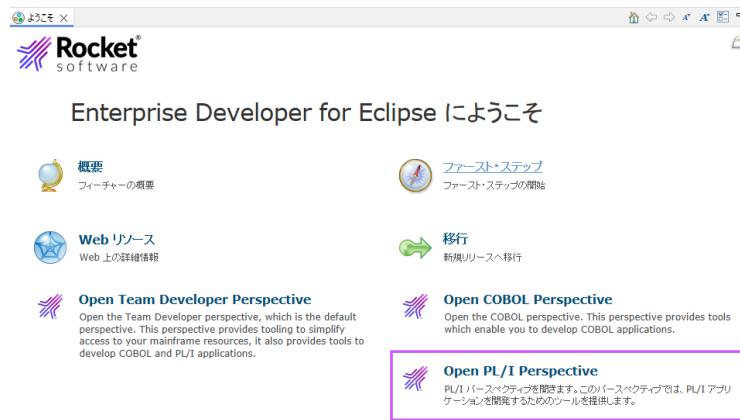
- 1) Enterprise Developer for Eclipse を起動します。



- 2) 前項で作成した C:¥work¥PLI 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。



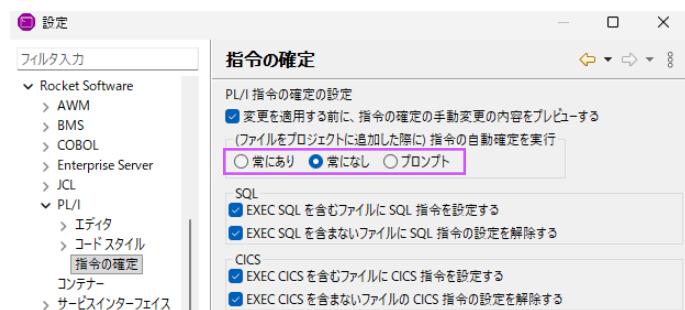
- 3) [ようこそ] タブが表示されますので、[Open PL/I Perspective] をクリックして、PL/I パースペクティブを開きます。



- 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。

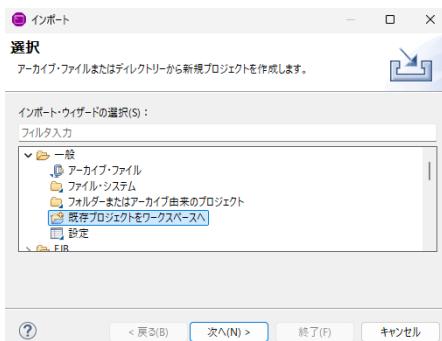


- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ウィンドウ] プルダウンメニューの [設定] > [Rocket Software] > [PL/I] > [指令の確定] > [指令の自動確定を実行] で [常なし] を選択し、[適用して閉じる] ボタンをクリックします。

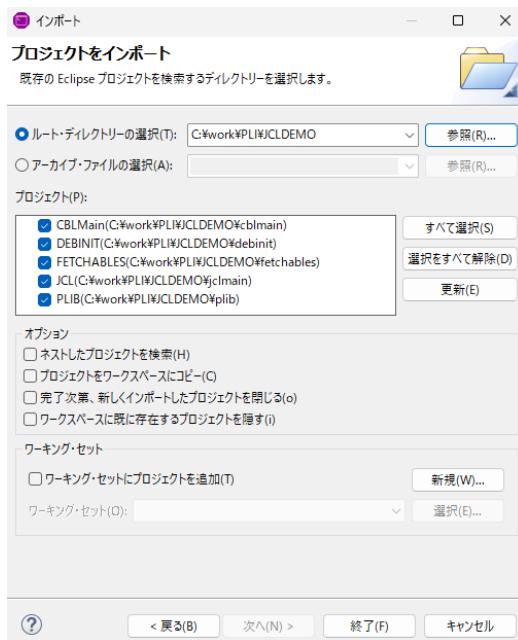


### 3.3 メインフレーム PL/I プロジェクトのインポート

- 1) 用意した例題プロジェクトをインポートします。[ファイル] プルダウンメニューから [インポート] を選択し、インポートウィンドウにて [一般] > [既存プロジェクトをワークスペースへ] を選択後 [次へ] ボタンをクリックします。



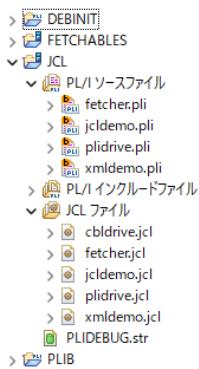
- 2) [ルート・ディレクトリの選択] へ C:\work\PLI\JCLDEMO を指定すると、このフォルダに含まれるプロジェクトが表示されます。チェックをオンにした状態で [終了] ボタンをクリックします。



- 3) PL/I エクスプローラーにインポートしたプロジェクトが表示されます。



- 4) JCL プロジェクトを展開すると PL/I ソースや JCL などが確認できます。



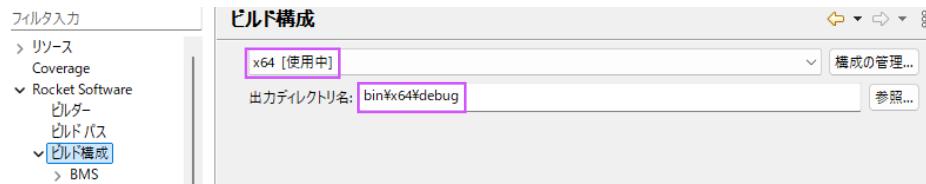
### 3.4 プロジェクトプロパティの確認

プロジェクトの設定値を確認していきます。

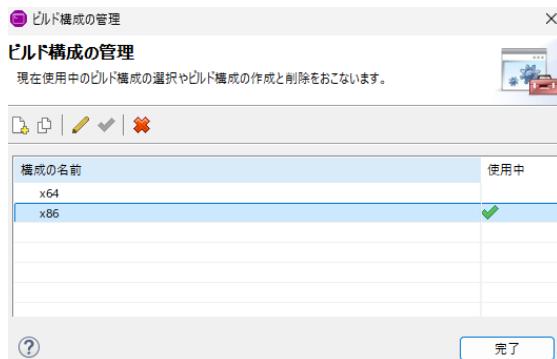
- 1) JCL プロジェクトを右クリックして [プロパティ] を選択するとプロパティウィンドウが表示されます。

- ① [Rocket Software] > [ビルド構成] を表示すると、bin¥x64¥debug フォルダへ64ビット稼働の実行ファイルが output されることが確認できます。

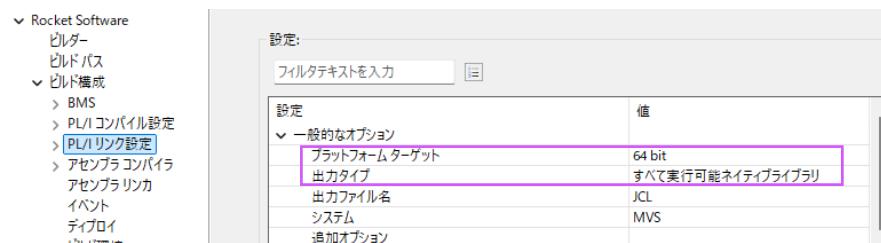
32 ビット稼働にする場合は [構成の管理] ボタンをクリックして x86 を選択してください。



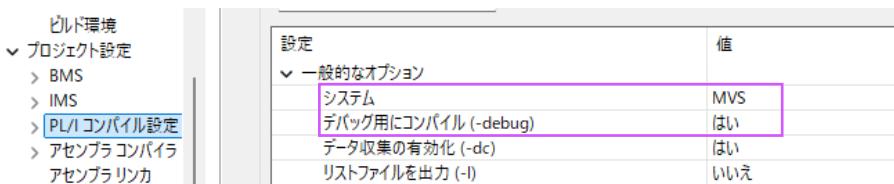
32 ビット稼働の場合)



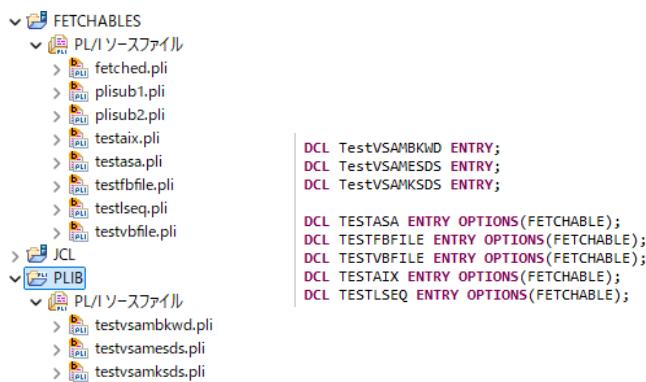
- ② [Rocket Software] > [ビルド構成] > [PL/I リンク設定] を選択して内容を確認すると、64 ビット稼働する実行可能ネイティブライブラリを実行ファイルタイプとして生成することがわかります。



- 2) [Rocket Software] > [プロジェクト設定] > [PL/I コンパイル設定] を選択して内容を確認すると、例題の内容に沿って、[システム] には MVS が設定されており、デバッグ実行用ファイルを生成することがわかります。確認後、[適用して閉じる] ボタンをクリックします。



- 3) jcldemo.pli プログラムから Fetchable と PLIB プロジェクトに含まれる実行モジュールを CALL するため、2 つのプロジェクトのプロパティも確認してください。



### 3.5 ビルドの実行

- 1) DLL を生成するために、Windows SDK が必要になります。リンクエラーを避けるために、これがインストールされているかご確認ください。また、複数の SDK や Microsoft Build Tools がインストールされている場合には、製品が使用するバージョンを指定することもできます。

使用可能な SDK と Microsoft Build Tools の確認コマンド)

```
cblms -L
c:\>cblms -L
Rocket (R) COBOL - Configuration Utility for the Microsoft Build Tools & SDK
11.0.0.88 (C) 1984-2025 Rocket Software, Inc. or its affiliates.

Windows SDK
Id バージョン 場所
0] 10.0.26100.0 c:\Program Files (x86)\Windows Kits\10

Microsoft Build Tools
Id バージョン 場所
```

最新バージョンに設定するコマンド)

```
cblms -U
c:\>cblms -U
Rocket (R) COBOL - Configuration Utility for the Microsoft Build Tools & SDK
11.0.0.88 (C) 1984-2025 Rocket Software, Inc. or its affiliates.

Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.26100.0
エラー: 有効な Build Tools が見つかりません
```

特定のバージョンを指定するコマンド例):以降は-L で表示された番号を指定します。

Windows SDK を指定する場合)cblms -US:0

Build Tools を指定する場合)cblms -UB:0

```
c:\>cblms -US:0
Rocket (R) COBOL - Configuration Utility for the Microsoft Build Tools & SDK
11.0.0.88 (C) 1984-2025 Rocket Software, Inc. or its affiliates.

Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.26100.0
```

使用するバージョンを表示するコマンド)

```
cblms -Q
c:\>cblms -Q
Rocket (R) COBOL - Configuration Utility for the Microsoft Build Tools & SDK
11.0.0.88 (C) 1984-2025 Rocket Software, Inc. or its affiliates.

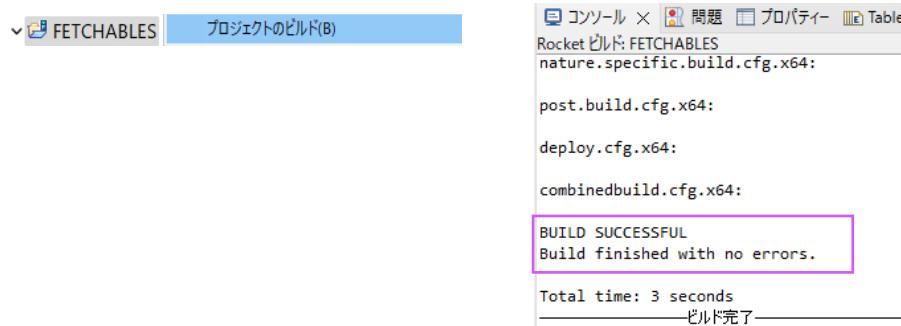
Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.26100.0

Microsoft Build Tools
場所 = C:\Program Files (x86)\Rocket Software\Enterprise Developer\Microsoft
バージョン = 14.40.33807
```

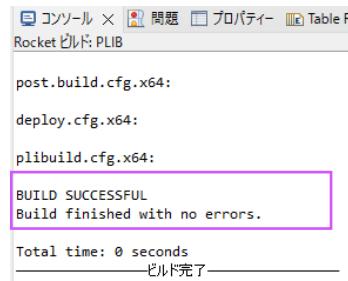
利用可能なオプションを表示するコマンド)

```
cblms -H
```

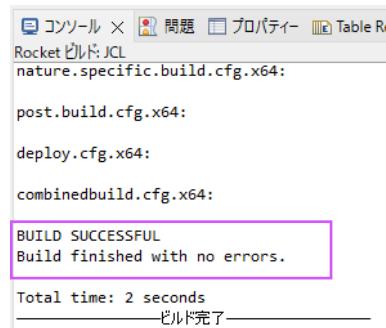
- 2) PL/I エクスプローラー内の FETCHABLES プロジェクトを右クリックして [プロジェクトのビルド] を選択し、コンソールタブでビルドの成功を確認します。



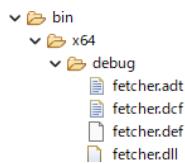
- 3) PL/I エクスプローラー内の PLIB プロジェクトを右クリックして [プロジェクトのビルド] を選択し、コンソールタブでビルドの成功を確認します。



- 4) PL/I エクスプローラー内の JCL プロジェクトを右クリックして [プロジェクトのビルド] を選択し、コンソールタブでビルドの成功を確認します。



- 5) 各プロジェクトの bin/x64/debug フォルダ配下に目的の実行可能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 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ドロードします。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。

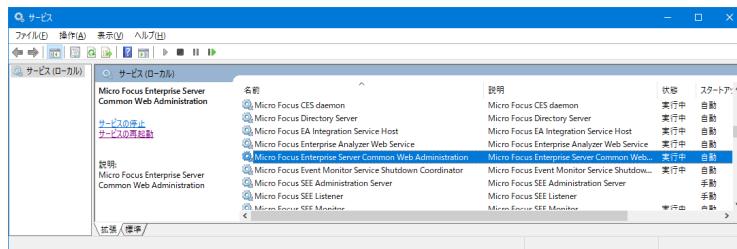
- 4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

- 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。

- 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。



### 3.7 Enterprise Server インスタンスの設定

Enterprise Server インスタンスには PL/I を実行するためのエンジンが搭載されており、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) 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"}]
```

上記例の場合、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。

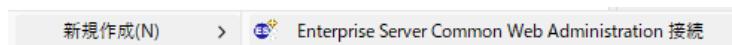
- 2) Enterprise Server インスタンスを作成します。

Eclipse に戻り、サーバーエクスプローラー内に [localhost] が表示されていない場合は、接続を作成します。存在する場合はこの項目はスキップしてください。

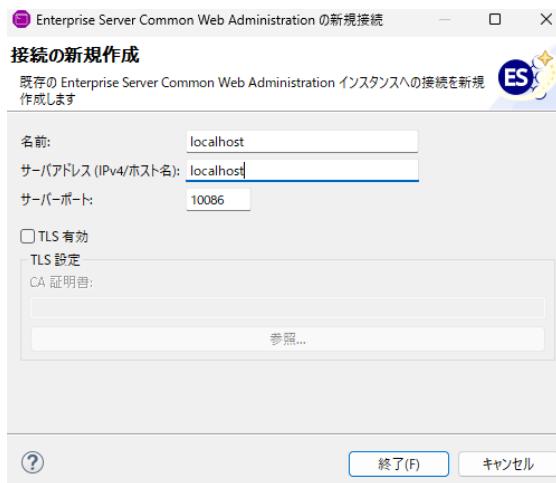


ESCWA インスタンス未定義

サーバーエクスプローラー内で右クリックし、[新規作成] > [Enterprise Server Common Web Administration 接続] を選択します。



[名前] と [サーバーアドレス] に localhost を入力し、デフォルトポートである 10086 が指定されていることを確認後、[終了] ボタンをクリックします。



[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。



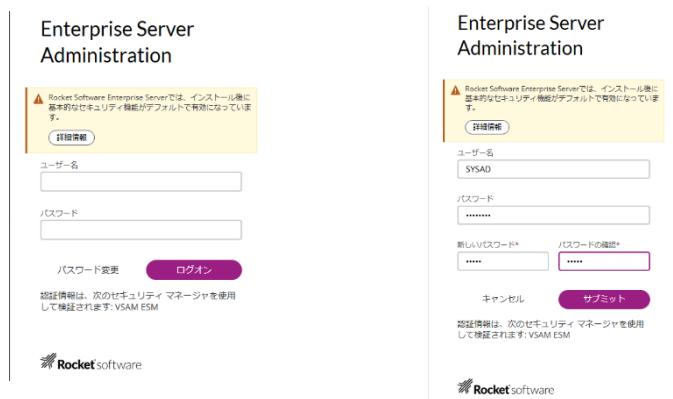
3) Web ブラウザが立ち上がり、ユーザー認証を求められます。

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、ESCAWA 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。

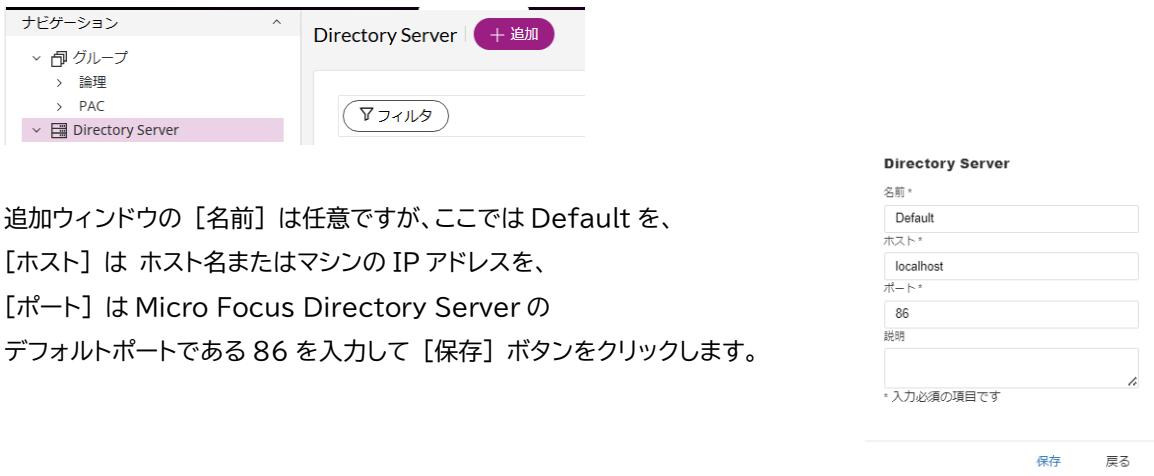


4) メニューで [オペレーション] を選択後、[ナビゲーション] に [Default] という名前の接続が表示されます。表示されない場合は新規に作成します。表示されている場合はこの項目はスキップしてください。





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。



5) 例題のフォルダには Enterprise Server インスタンスのサンプルが含まれており、これをインポートします。

C:\work\PLI\JCLDEMO\JCL\_SERVERS.xml がインポート対象のファイルです。

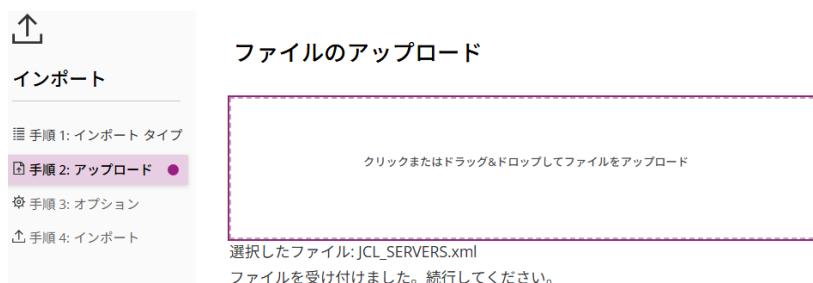
ESCWA の左側ペインで Default を選択し、右側ペインの [インポート] をクリックします。



表示されたインポートウィンドウでは、[XML] を選択して [次へ] をクリックします。



[ファイルのアップロード] の四角い枠内をクリックして XML ファイルを指定し、[ファイルを受け付けました] メッセージを確認後、[次へ] ボタンをクリックします。



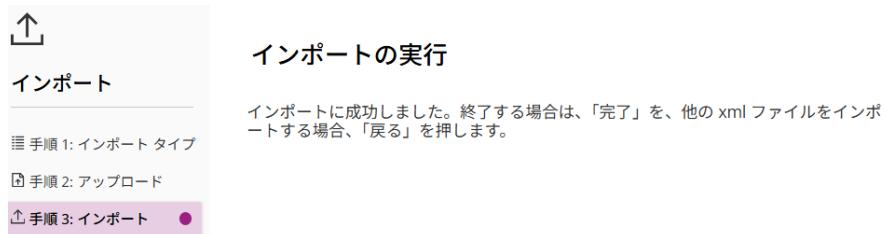
手順 3 では [次へ] をクリックします。



手順 4 では [インポート] をクリックします。



インポートの完了が通知されますので、[完了] ボタンをクリックします。



- 6) 32 ビットアプリケーション稼働用の PLIJCL と、64 ビットアプリケーション稼働用の PLIJCL64 インスタンスが追加されます。Eclipse では 64 ビットアプリケーション稼働用にコンパイルしましたので、ビット数が一致する PLIJCL64 インスタンスを使用します。

ア...	名前	タ...	ステータス	64ビット	MSS有効	セキュ...	アクション	
■	PLIJCL	Region	Stopped	✓	デフォルト			
■	PLIJCL64	Region	Stopped	✓	✓	デフォルト		



### 重要

アプリケーション稼働ビット数 = 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) 設定を変更するため、名前をダブルクリック、または [編集] アイコンをクリックします。

■	PLIJCL64	Region	Stopped	✓	✓	デフォルト		
<input type="button" value="編集"/>								

- 8) [開始オプション] の [システムディレクトリ] には前項で作成した system フォルダを指定します。このフォルダにインスタンスのログなどが出力されます。

## 開始オプション

\* 入力必須の項目です

名前*	システム ディレクトリ
PLIJCL64	C:\work\PLI\JCLDEMO\system

- 9) [開始オプション] の [ローカルコンソールを表示] チェックボックスをオフに、[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。

<input type="checkbox"/> ローカル コンソールを表示	<input checked="" type="checkbox"/> 動的デバッグを許可
<input checked="" type="checkbox"/> 64ビット作業モード	<input type="checkbox"/> 以前のログを削除
<input type="checkbox"/> システム起動時に開始する	

- 10) [リージョンの機能] では [MSS 有効], [JES 有効] のチェックがオンであることを確認します。

### リージョンの機能

<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

- 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 環境変数へ 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。

変更前；

```
[ES-Environment]
JBASE=C:\Users\Public\Documents\Rocket Software\Enterprise
Developer\Samples\PLI-VS or PLI-Eclipse"\JCLDEMO
JDEMO=$JBASE\jclmain
JFETCH=$JBASE\fetchables
JDEBINIT=$JBASE\debinit
JCBLBASE=$JBASE\cblmain
```

変更後；

### 追加設定

```
[ES-Environment]
JBASE=C:\work\PLI\JCLDEMO
MFACCCGI_CHARSET=Shift_JIS
JDEMO=$JBASE\jclmain
JFETCH=$JBASE\fetchables
JDEBINIT=$JBASE\debinit
JCBLBASE=$JBASE\cblmain
```

構成情報

```
[ES-Environment]
JBASE=C:\work\PLI\JCLDEMO
MFACCCGI_CHARSET=Shift_JIS
JDEMO=$JBASE\jclmain
JFETCH=$JBASE\fetchables
JDEBINIT=$JBASE\debinit
JCBLBASE=$JBASE\cblmain
```

- 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確認します。

項目名	説明
JES プログラム パス	PL/I アプリケーションの実行可能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オプション	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。ここでは指定しません。



- 13) [イニシエータ] 項目でイニシエータ定義を確認します。A~9までのクラスを受入れるイニシエータが設定されています。



- 14) 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 では、パフォーマンス向上の観点から 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。

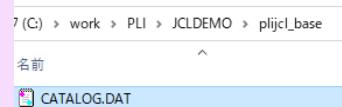
対象ファイルの特定には MFSYSCAT 環境変数を利用して、カタログファイルを指定します。

例)

```
set MFSYSCAT=C:\work\PLI\JCLDEMO\plijcl_base\catalog.dat
```

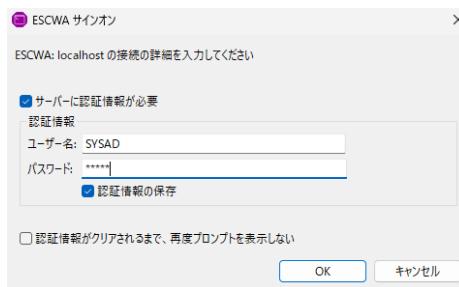
```
mfsplcnv -2
```

詳しくは製品マニュアルをご参照ください。

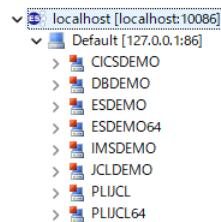


## 3.8 Enterprise Server インスタンスの開始と確認

- 1) Eclipse に戻り、[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。



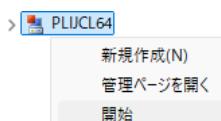
- 2) [Default] を展開して PLIJCL64 インスタンスが表示されていることを確認します。表示されていない場合は [Default] を右クリックし、[更新] を選択してリフレッシュしてください。



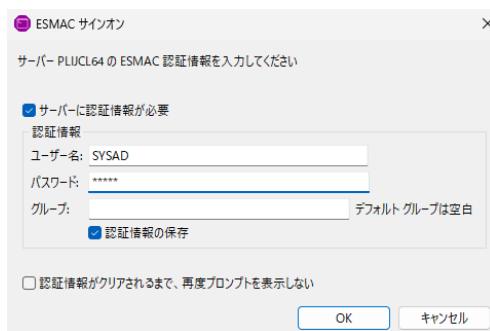
- 3) サーバーエクスプローラー内の PLIJCL64 インスタンスを右クリックし、[プロジェクトに関連付ける] > [JCL] を選択します。これにより Eclipse 内の JCL プロジェクトから実行される JCL は PLIJCL64 インスタンスで処理されることになります。



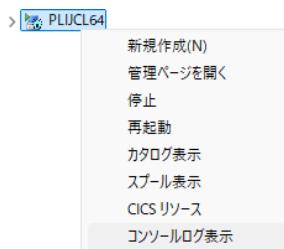
- 4) PLIJCL64 インスタンスを右クリックして [開始] を選択します。



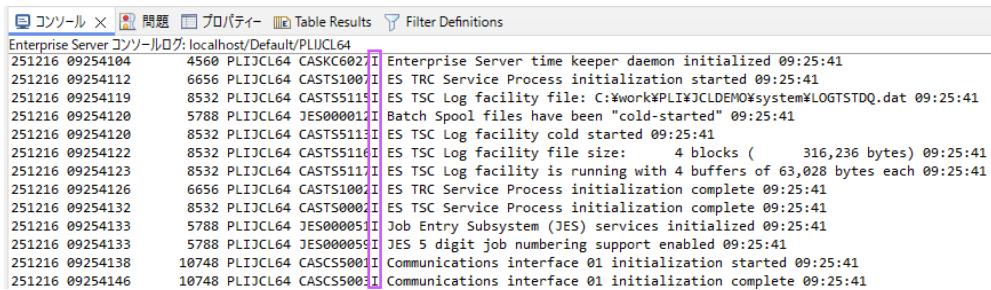
- 5) 開始ユーザーの認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。



- 6) 開始後、PLIJCL64 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。



```

Enterprise Server コンソールログ: localhost/Default/PLIJCL64
251216 09254104 4560 PLIJCL64 CASKC6027I Enterprise Server time keeper daemon initialized 09:25:41
251216 09254112 6656 PLIJCL64 CASTS1007I ES TRC Service Process initialization started 09:25:41
251216 09254119 8532 PLIJCL64 CASTS5115I ES TSC Log facility file: C:\work\PLI\JCLDEMO\system\LOGTSTDQ.dat 09:25:41
251216 09254120 5788 PLIJCL64 JES000012I Batch Spool files have been "cold-started" 09:25:41
251216 09254120 8532 PLIJCL64 CASTS5113I ES TSC Log facility cold started 09:25:41
251216 09254122 8532 PLIJCL64 CASTS5116I ES TSC Log facility file size: 4 blocks ( 316,236 bytes) 09:25:41
251216 09254123 8532 PLIJCL64 CASTS5117I ES TSC Log facility is running with 4 buffers of 63,028 bytes each 09:25:41
251216 09254126 6656 PLIJCL64 CASTS1002I ES TRC Service Process initialization complete 09:25:41
251216 09254132 8532 PLIJCL64 CASTS0002I ES TSC Service Process initialization complete 09:25:41
251216 09254133 5788 PLIJCL64 JES000051I Job Entry Subsystem (JES) services initialized 09:25:41
251216 09254133 5788 PLIJCL64 JES000059I JES 5 digit job numbering support enabled 09:25:41
251216 09254138 10748 PLIJCL64 CASC5001I Communications interface 01 initialization started 09:25:41
251216 09254146 10748 PLIJCL64 CASC5001I Communications interface 01 initialization complete 09:25:41

```



### 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。

## 3.9 JCL の実行

現在インスタンスが稼働しているので、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) Eclipse の PL/I エクスプローラー内に存在する jcldemo.jcl をダブルクリックして内容を表示します。IDCAM などのユーティリティを使用してファイルを操作したのち、JCLDEMO プログラムを実行していることがわかります。

```

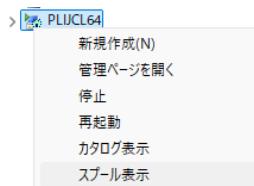
//*****
//* Run the JCLDEMO Program
//*****
//STEP100 EXEC PGM=JCLDEMO
//SYSOUT DD SYSOUT=*,HOLD=Y
//SYSPRINT DD SYSOUT=*,HOLD=Y,DCB=(RECFM=LSEQ)
//B1079256 DD DISP=(,CATLG),SPACE=(CYL,(5,5),RLSE),
//           DCB=(RECFM=FBA,LRECL=137,BLKSIZE=0),
//           DSN=SYSAD.STREAM.TEST

```

- 2) PL/I エクスプローラーから jcldemo.jcl を選択して右クリック後、[Enterprise Server へサブミット] を選択すると、前項で関連付けた PLIJCL64 インスタンスで、この JCL が実行されます。

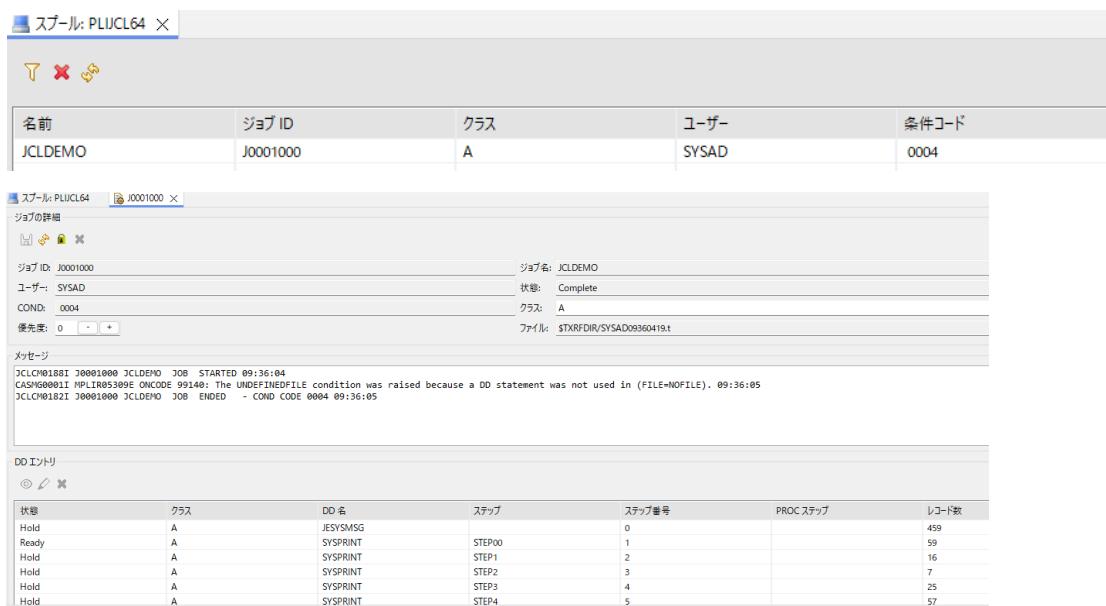


- 3) サーバーエクスプローラー内の PLIJCL64 インスタンスを右クリックし、[スプール表示] を選択します。

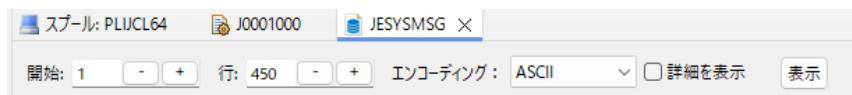


実行された JCL の結果を確認すると、条件コードは 0004 で終了したことがわかります。

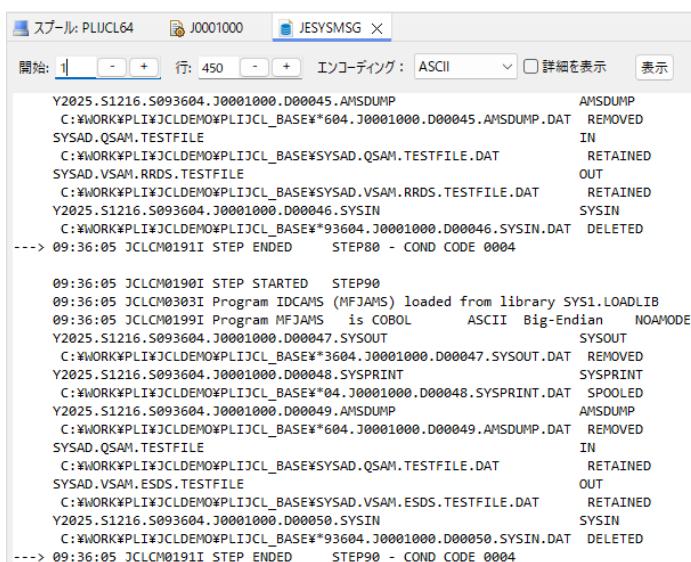
該当行をダブルクリックして詳細を表示します。



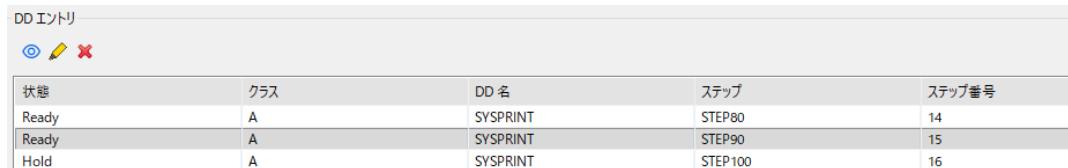
DD エントリの JESYSMSG をダブルクリックして内容を確認するため、表示行を 450 に変更し、[表示] ボタンをクリックします。



JESYSMSG のステップ名 STEP60 から STEP090 でリターンコードに 0004 が返却されていることがわかります。



- 4) リターンコード 0004 の理由を探るため、DD エントリの STEP90 の [SYSPRINT] をダブルクリックします。最終行にワーニングが発生しており、JCL で指定した 100 件のレコードを下回ったため発生した警告と判断できます。



DD エントリ				
状態	クラス	DD 名	ステップ	ステップ番号
Ready	A	SYSPRINT	STEP80	14
Ready	A	SYSPRINT	STEP90	15
Hold	A	SYSPRINT	STEP100	16

```

REPRO INFILE(IN) OUTFILE(OUT) COUNT(100)
JCLAM0134I(00) - 00000009 Records processed.
JCLAM0194W(04) - Number of records read was less than COUNT(00000100).

```

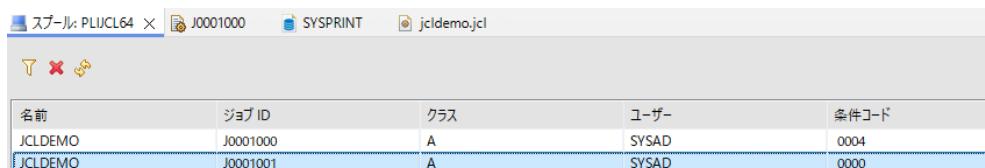
- 5) Eclipse で Jcldemo.jcl の STEP60 から STEP090 に記述されている COUNT(100)を COUNT(5)へ修正して保存し、JCL を再実行します。

```

//SYSIN DD *
  REPRO INFILE(IN) OUTFILE(OUT) COUNT(5)
/*

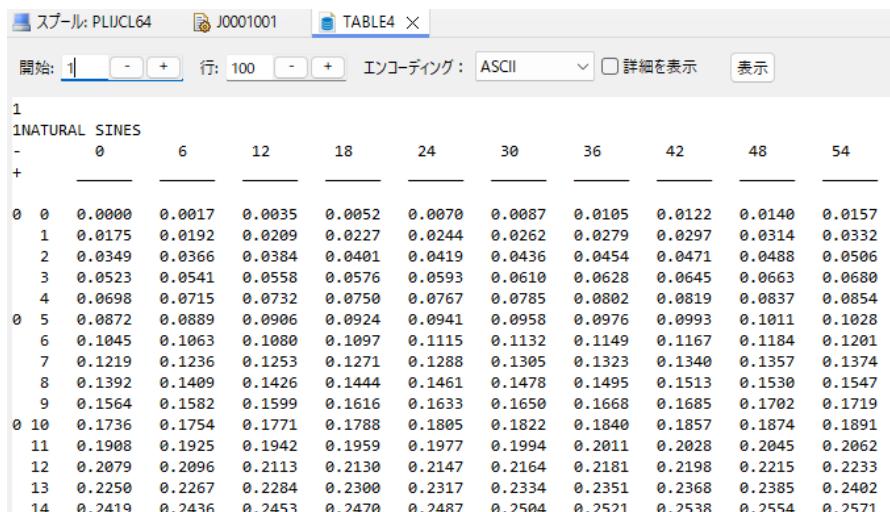
```

- 6) 再度スプールを確認すると、正常に終了していることがわかります。



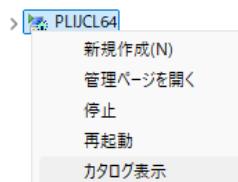
名前	ジョブ ID	クラス	ユーザー	条件コード
JCLDEMO	J0001000	A	SYSAD	0004
JCLDEMO	J0001001	A	SYSAD	0000

- 7) STEP100 では jcldemo.pli ソースから出力された内容が参照できますので、ソースコードと合わせて確認してみてください。



1NATURAL SINES										
-	0	6	12	18	24	30	36	42	48	54
0	0.0000	0.0017	0.0035	0.0052	0.0070	0.0087	0.0105	0.0122	0.0140	0.0157
1	0.0175	0.0192	0.0209	0.0227	0.0244	0.0262	0.0279	0.0297	0.0314	0.0332
2	0.0349	0.0366	0.0384	0.0401	0.0419	0.0436	0.0454	0.0471	0.0488	0.0506
3	0.0523	0.0541	0.0558	0.0576	0.0593	0.0610	0.0628	0.0645	0.0663	0.0680
4	0.0698	0.0715	0.0732	0.0750	0.0767	0.0785	0.0802	0.0819	0.0837	0.0854
5	0.0872	0.0889	0.0906	0.0924	0.0941	0.0958	0.0976	0.0993	0.1011	0.1028
6	0.1045	0.1063	0.1080	0.1097	0.1115	0.1132	0.1149	0.1167	0.1184	0.1201
7	0.1219	0.1236	0.1253	0.1271	0.1288	0.1305	0.1323	0.1340	0.1357	0.1374
8	0.1392	0.1409	0.1426	0.1444	0.1461	0.1478	0.1495	0.1513	0.1530	0.1547
9	0.1564	0.1582	0.1599	0.1616	0.1633	0.1650	0.1668	0.1685	0.1702	0.1719
10	0.1736	0.1754	0.1771	0.1788	0.1805	0.1822	0.1840	0.1857	0.1874	0.1891
11	0.1908	0.1925	0.1942	0.1959	0.1977	0.1994	0.2011	0.2028	0.2045	0.2062
12	0.2079	0.2096	0.2113	0.2130	0.2147	0.2164	0.2181	0.2198	0.2215	0.2233
13	0.2250	0.2267	0.2284	0.2300	0.2317	0.2334	0.2351	0.2368	0.2385	0.2402
14	0.2419	0.2436	0.2453	0.2470	0.2487	0.2504	0.2521	0.2538	0.2554	0.2571

- 7) サーバーエクスプローラー内の PLIJCL64 インスタンスを右クリックし、[カタログ表示] を選択すると、実行した JCL から作成されたカタログ情報が確認できます。



DS 編成	DS 名
VSAM	SYSAD.CLUSTER.AIX
VSAM	SYSAD.CLUSTER.BASE
VSAM	SYSAD.CLUSTER.BASE.DATA
VSAM	SYSAD.CLUSTER.BASE.INDEX
VSAM	SYSAD.CLUSTER.PATH
PS	SYSAD.QSAM.TESTFILE
UNKNOWN	SYSAD.STREAM.TEST
PS	SYSAD.TABLE5
PS	SYSAD.TABLE6
PS	SYSAD.VBFILE
PS	SYSAD.VBOUT
VSAM	SYSAD.VSAM.ESDS.TESTFILE
VSAM	SYSAD.VSAM.KSDS.TESTFILE
VSAM	SYSAD.VSAM.KSDS2.TESTFILE
VSAM	SYSAD.VSAM.RRDS.TESTFILE

- 8) 参照したいカタログ情報をダブルクリックすると DCB 情報が表示されます。変更も可能です。

カタログ : PLIJCL64		SYSAD.CLUSTER.BASE.DATA
DS 名:	SYSAD.CLUSTER.BASE.DATA	
物理ファイル:	C:\WORK\PLI\JCLDEMO\PLI_JCL\BASE\SYSAD.CLUSTER.BASE.DAT	
DS 編成:	VSAM	RECFM: KS
コードセット:	ASCII	<input checked="" type="checkbox"/> カタログ
LRECL:	80	BLKSIZE: 0
作成日時:	2025/12/16 09:53:07.94	参照日時: 2025/12/16 09:53:07.94
MGMTCLASS:		
VSAM 型:	Data	VSAM 属性: <input type="text"/>
サーバー間:	2	システム間: 3
キー先頭:	0	キーの長さ: 8
キー最大:	80	キー平均: 8

- 9) DCB 情報の表示アイコンをクリックするとデータが参照できます。

カタログ : PLIJCL64		SYSAD.CLUSTER.BASE.DATA	SYSAD.CLUSTER.BASE.DATA
開始	0	- + 行	1000 - + エンコーディング ASCII
<input type="checkbox"/> 詳細 <input type="checkbox"/> 表示			
RECORD01 AIX9 DATA-010101010101 RECORD02 AIX4 DATA-020202020202 RECORD03 AIX5 DATA-030303030303 RECORD04 AIX4 DATA-040404040404 RECORD05 AIX7 DATA-050505050505 RECORD06 AIX2 DATA-060606060606 ***END OF FILE***			

- 10) カタログ機能からカタログの新規作成や削除も可能です。

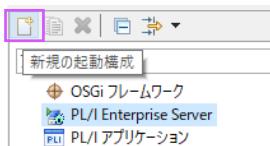
### 3.10 PL/I ソースのデバッグ

JCL から実行される PL/I プログラムをデバッグします。

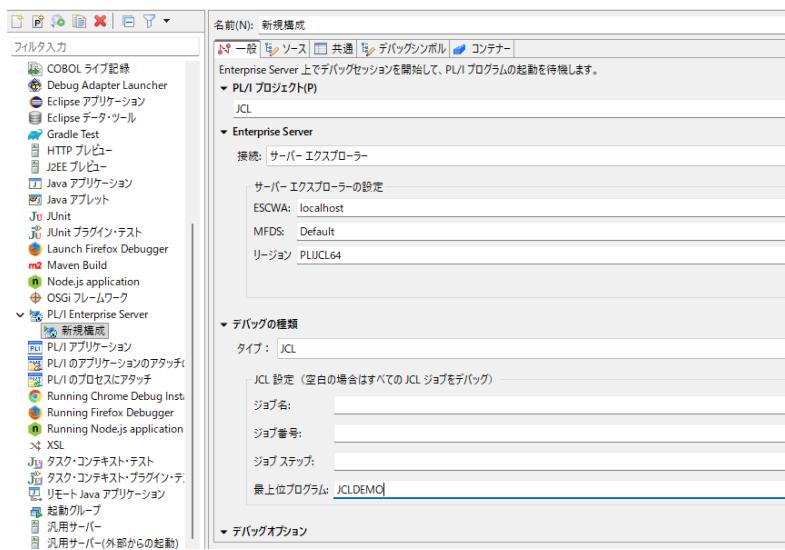
- 1) Eclipse の [実行] プルダウンメニューから [デバッグの構成] を選択します。



- 2) 左側のツリービューから [PL/I 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。

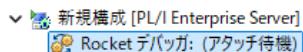


- 3) [PL/I プロジェクト] へ対象となる JCL プロジェクトを入力し、[Enterprise Server] へ実行せざる PLIJCL64 インスタンスを指定します。[デバッグの種類] は JCL を選択した状態で、[最上位プログラム] には JCLDEMO を指定し、[デバッグ] ボタンをクリックします。



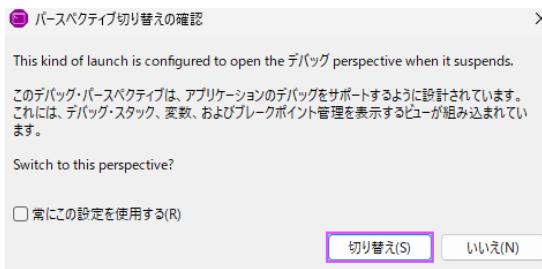
- 4) パースペクティブの切り替え確認ウィンドウが表示されますが、ここでは [いいえ] を選択します。

- 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。

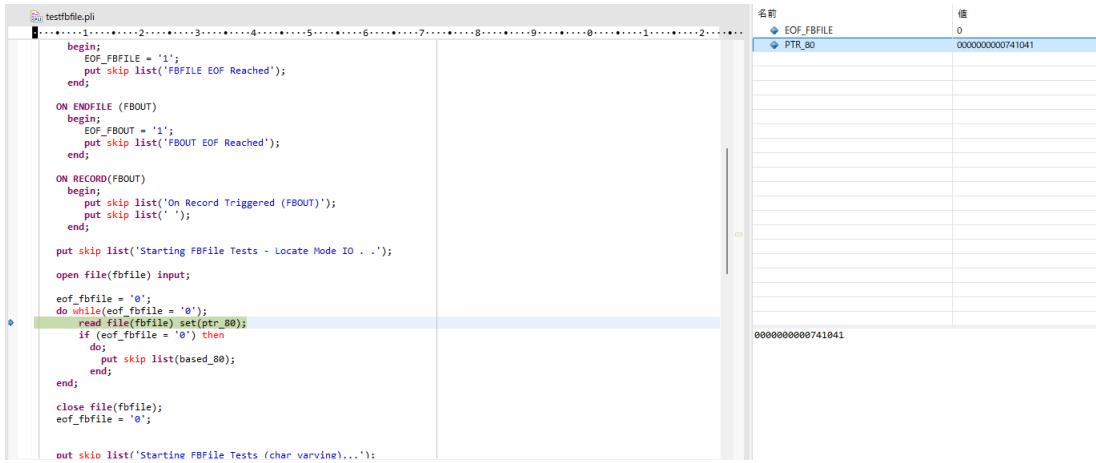


- 6) PL/I エクスプローラー内の jcldemo.jcl を右クリックして [Enterprise Server へのサブミット] を選択し、JCL を実行します。

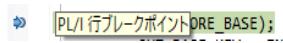
- 7) パースペクティブの切り替え確認ウィンドウが再度表示されますので、[切り替え] ボタンをクリックし、デバッグ用のパースペクティブを開きます。



- 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ます。[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。



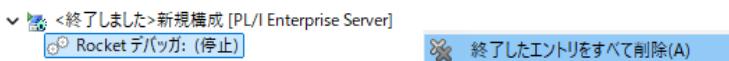
- 10) 先に進める場合は画面上部の再開アイコンをクリックします。



- 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。



- 12) デバッグが停止状態になったのを確認後、右クリックして [終了したエントリをすべて削除] を選択し、これを削除します。

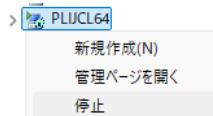


- 13) PL/I パースペクティブへ戻るには画面右上の PL/I アイコンをクリックします。



### 3.11 終了処理

- 1) サーバーエクスプローラー内で PLIJCL64 インスタンスを右クリックして [停止] を選択し、開始中のインスタンスを停止します。



- 2) PLIJCL64 インスタンスの停止状態を確認後に、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。