

# Enterprise Developer チュートリアル

## メインフレーム PL/I 開発 : CICS Eclipse 編

### 1. 目的

本チュートリアルでは、PL/I 言語で書かれた CICS 命令を含むソースをオープン環境へ移行後、Eclipse を使用してプロジェクトの作成、コンパイル、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. TN3270 エミュレーターがインストール済で稼働実績があること

### 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. BMS 画面定義の確認
6. CICS リソース定義の概念
7. ビルドの実行
8. 文字エンコーディングの設定
9. Enterprise Server インスタンスの設定
10. Enterprise Server インスタンス開始と確認
11. CICS の実行
12. PL/I ソースのデバッグ
13. 終了処理

### 4. 免責事項

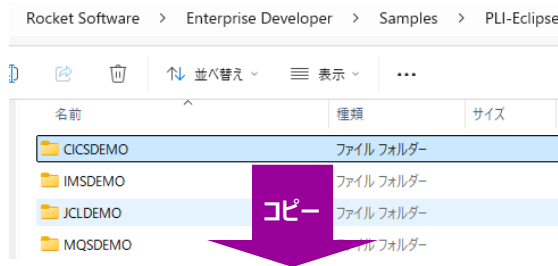
### 3.1 チュートリアルの準備

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) 下記のパスに配置されている例題プログラムの CICSDEMO フォルダを作成した C:\work\PLI へコピーします。

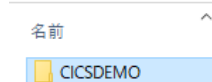
例)

C:\Users\Public\Documents\Rocket Software\Enterprise Developer\Samples\PLI-Eclipse\CICSDEMO

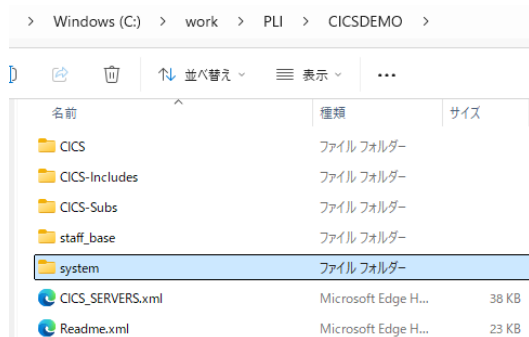


コピー先) C:\work\PLI\CICSDEMO

スク (C:) > work > PLI



- 3) C:\work\PLI\CICSDEMO フォルダ配下に実行時に使用する system 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。

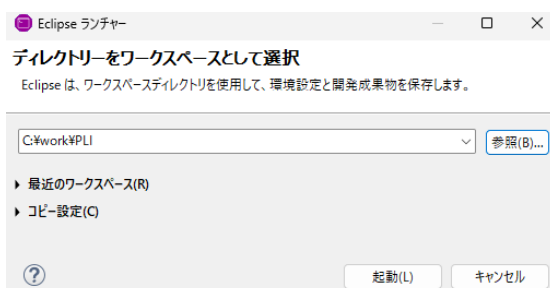


### 3.2 Eclipse の起動

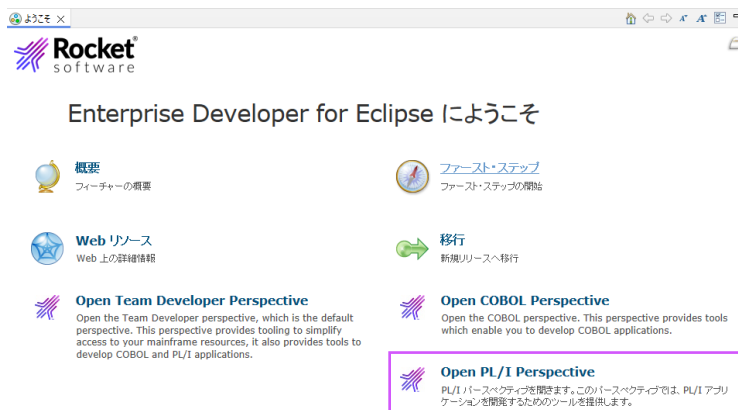
- 1) Enterprise Developer for Eclipse を起動します。



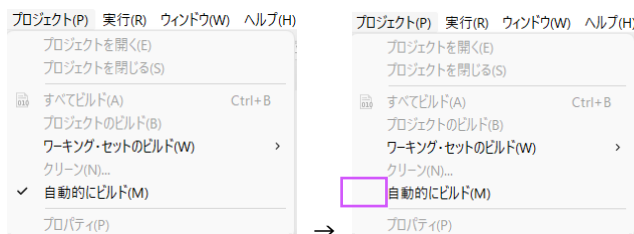
- 2) 前項で作成した C:\work\PLI をワークスペースへ指定して、[起動] ボタンをクリックします。



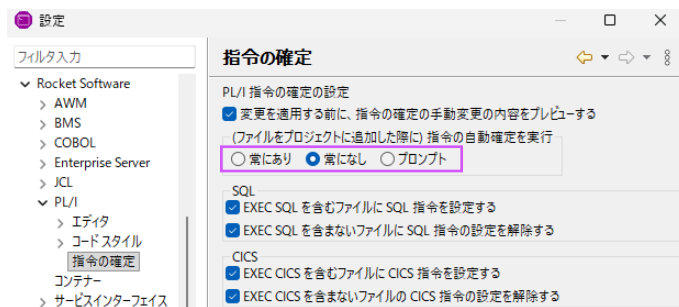
- 3) [ようこそ] タブが表示されますので、[Open PL/I Perspective] をクリックして、PL/I パースペクティブを開きます。



- 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。

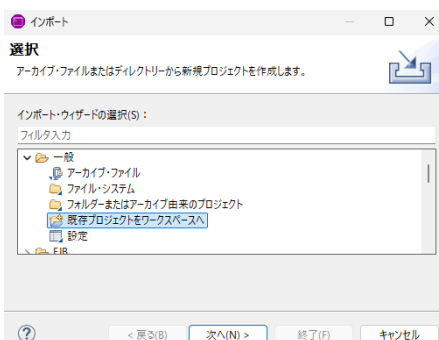


- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ウィンドウ] プルダウンメニューの [設定] > [Rocket Software] > [PL/I] > [指令の確定] > [指令の自動確定を実行] で [常になし] を選択し、[適用して閉じる] ボタンをクリックします。

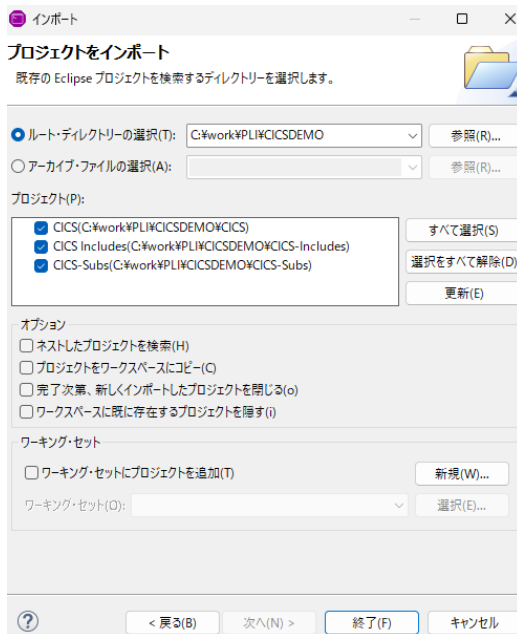


### 3.3 メインフレーム PL/I プロジェクトのインポート

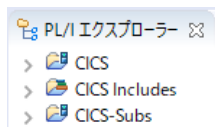
- 1) コピーしたプロジェクトをインポートします。[ファイル] プルダウンメニューから [インポート] を選択し、インポートウィンドウにて [一般] > [既存プロジェクトをワークスペースへ] を選択後 [次へ] ボタンをクリックします。



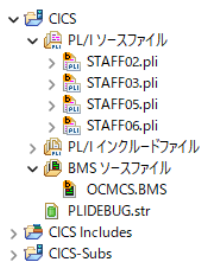
- 2) [ルート・ディレクトリの選択] へ C:\work\PLI\CICSDemo を指定すると、このフォルダに含まれるプロジェクトが表示されます。全てのチェックをオンにした状態で [終了] ボタンをクリックします。



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



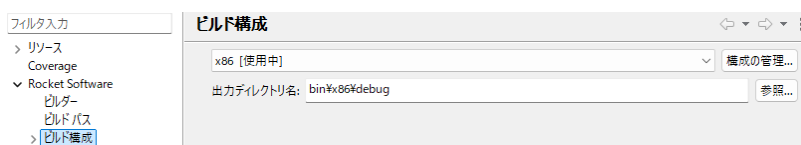
- 4) CICS プロジェクトを展開すると PL/I ソースなどが確認できます。



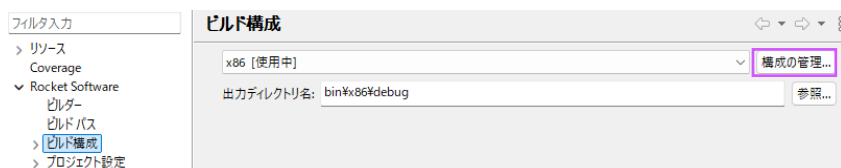
### 3.4 プロジェクトプロパティの確認

この例題では CICS-Subs プロジェクトで生成される lib オブジェクトを CICS プロジェクトがリンクして dll を生成します。プロジェクトの設定値を確認します。

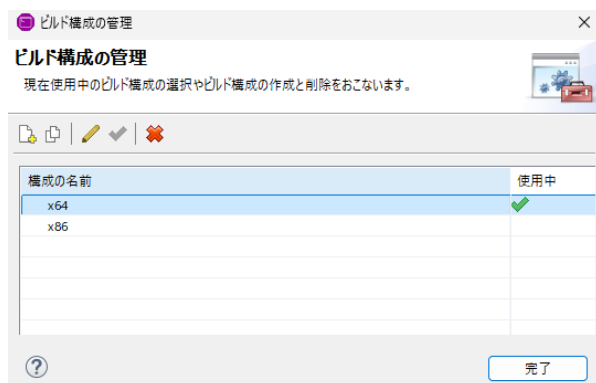
- 1) CICS-Subs プロジェクトを右クリックし、[プロパティ] を選択すると [プロパティ ウィンドウ] が表示されます。ここでは 32 ビット稼働の実行モジュールを生成するように指定されています。64 ビットを生成する場合は下記の手順に従ってください。



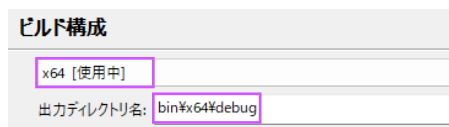
- ① [Rocket Software] > [ビルド構成] で [構成の管理] ボタンをクリックして構成管理ウィンドウを表示します。



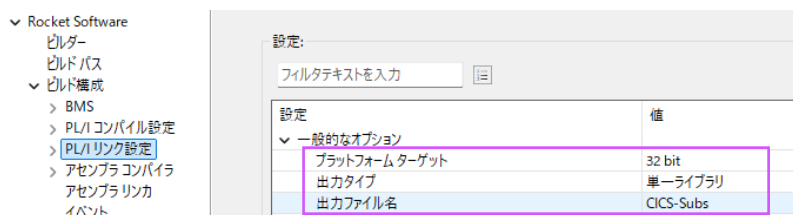
- ② [ビルドの構成管理] ウィンドウでは [x64] のチェックボックスをオンにして [完了] ボタンをクリックします。



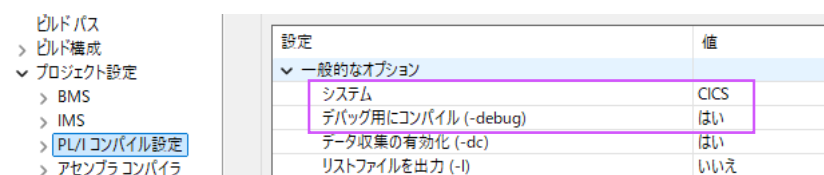
- ③ [Rocket Software] > [ビルド構成] ウィンドウへ戻り [x64] へ変更されたことと、プロジェクト配下の bin¥x64¥debug フォルダへ実行ファイルが出力されることを確認後 [適用] ボタンをクリックします。



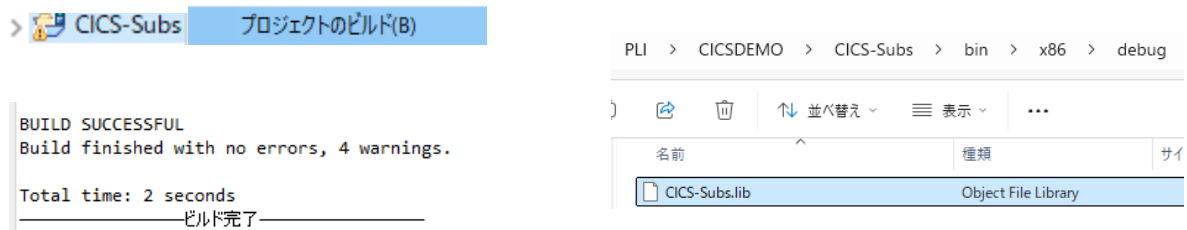
- 2) [Rocket Software] > [ビルド構成] > [PL/I リンク設定] を選択して内容を確認すると下記の値が設定されていますので、[出力タイプ] に 単一ライブラリ が、[出力ファイル名] には CICS-Subs が指定されていることを確認します。これにより 32 ビット稼働用の単一ライブラリである CICS-Subs.lib が生成されます。



- 3) [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 CICS が設定されており、デバッグ用のファイルを生成することがわかります。最後に [適用して閉じる] ボタンをクリックしてください。



- 4) 先に lib を作成するため、[PL/I エクスプローラー] タブで CICS-Subs プロジェクトを右クリックし [プロジェクトのビルド] を選択し、コンソールタブで成功を確認します。



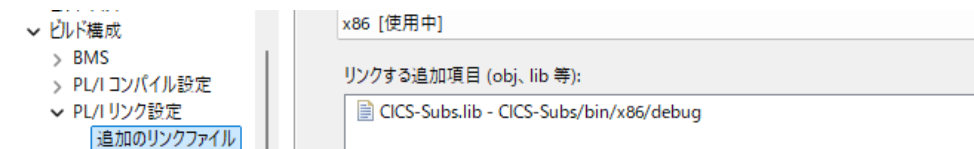
- 5) 次に CICS プロジェクトも同様に [プロパティ ウィンドウ] を表示します。  
CICS-Subs プロジェクトと同じ 32 ビット稼働が指定されています。CICS-Subs プロジェクトに 64 ビット稼働を指定した場合は CICS プロジェクトも同じビット数を指定します。



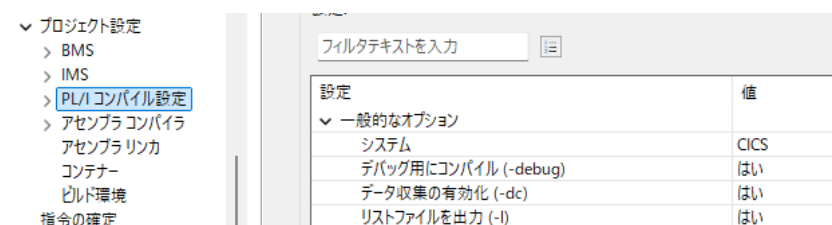
- 6) [Rocket Software] > [ビルド構成] > [PL/I リンク設定] を選択して内容を確認すると、CICS という名前で 32 ビット稼働する実行可能ネイティブライブラリとしてオブジェクトを生成することがわかります。



- 7) [Rocket Software] > [ビルド構成] > [PL/I リンク設定] > [追加のリンクファイル] を選択すると、「CICS-Subs.lib」をリンクするように指定していることがわかります。

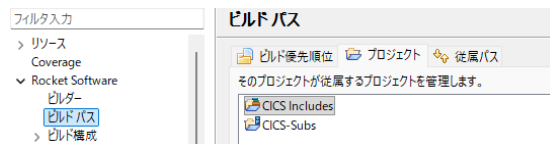


- 8) [Rocket Software] > [プロジェクト設定] > [PL/I コンパイル設定] を選択して内容を確認すると、例題の内容に沿って、「システム」には CICS が設定されており、デバッグ用のファイルも生成することがわかります。

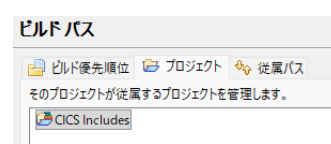


- 9) [Rocket Software] > [ビルドパス] の [プロジェクト] タブで従属プロジェクトを確認します。lib を生成している CICS-Subs プロジェクトと include ファイルが存在する CICS includes プロジェクトに従属していることが確認でき、これを基にコンパイルが実行されます。確認後は [適用して閉じる] ボタンをクリックしてください。

#### 【CICS プロジェクト指定】



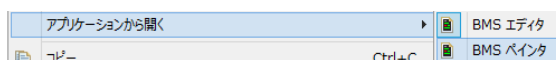
#### 【CICS-sub プロジェクト指定】



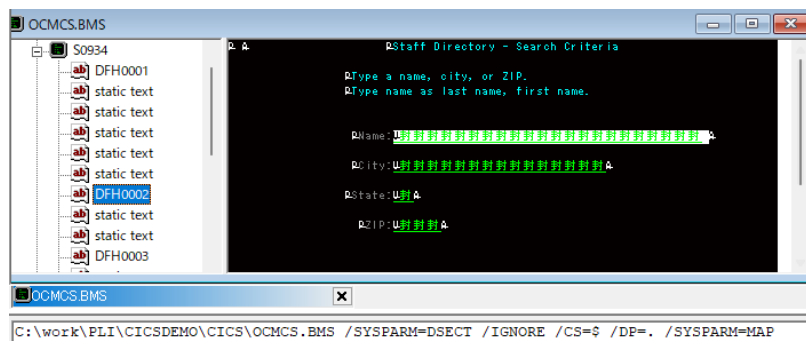
### 3.5 BMS 画面定義の確認

Enterprise DeveloperにはCICS 開発者のために BMS 画面を対話型で編集するユーティリティやビューアーが装備されています。

- 1) PL/I エクスプローラーのプロジェクト内に存在する BMS ソースファイルフォルダの OCMCS.BMS ファイルを右クリックして [アプリケーションから開く] > [BMS ペインタ] を選択します。



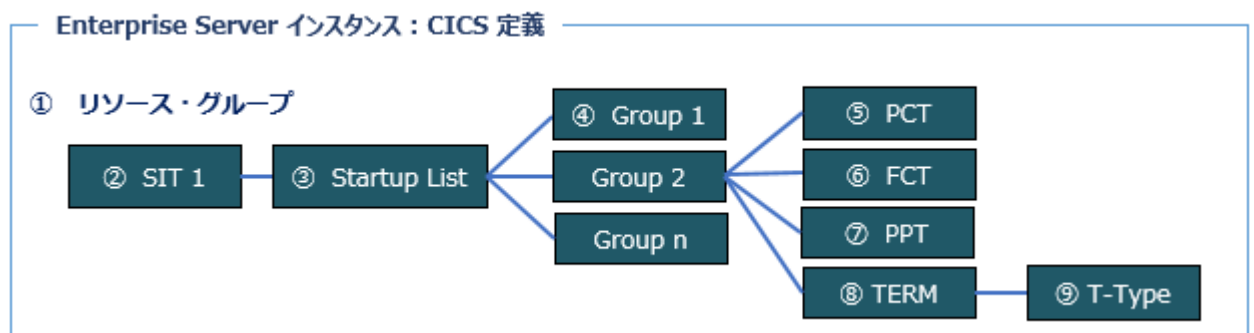
- 2) BMS ペインタウィンドウが表示され、画面定義内容をグラフィカルにメンテナンスできます。左側のツリービューでオブジェクトを選択すると右側のグラフィカルビュー内で対応するオブジェクトがハイライトされます。



- 3) [ファイル] プルダウンメニューの [終了] を選択して BMS ペインタウィンドウを終了します。

### 3.6 CICS リソース定義の概念

CICS ではアプリケーションで使用するソフトウェアやハードウェアの項目をリソースと呼び、Enterprise Server インスタンスの Mainframe Subsystem Support(MSS)は、このリソースを定義、制御、および監視するための機能を備えています。



項目名	説明
① リソース・グループ	SIT に関連する CICS リソースのセットを指します。
② SIT	CICS インスタンスの詳細設定を指定する、システム初期化テーブルを指します。
③ Startup List	CICS インスタンス起動時、自動的にロードされるグループ一覧を指します。
④ Group	PCT などの制御テーブルが所属するグループを指します。
⑤ PCT	CICS で使用するトランザクション制御テーブルを指します。
⑥ FCT	CICS で使用するファイル制御テーブルを指します。
⑦ PPT	CICS で使用するプログラム制御テーブルを指します。
⑧ TERM	端末定義を指します。
⑨ T-Type	端末タイプを指します。

### 3.7 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:\xa>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:\xa>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:\xa>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:\xa>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.

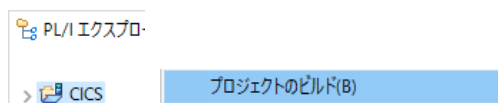
Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.26100.0

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

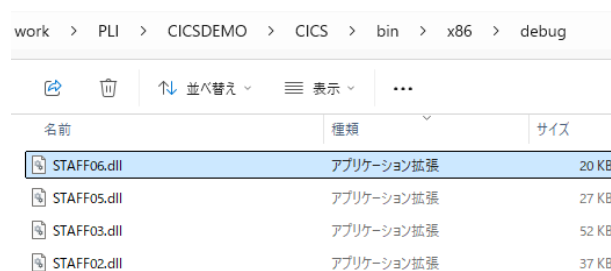
利用可能なオプションを表示するコマンド)

cblms -H

- 2) PL/I エクスプローラー内の CICS 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) CICS プロジェクトの bin\x86\debug フォルダ配下に目的の dll が作成されていることを確認してください。



### 3.8 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。

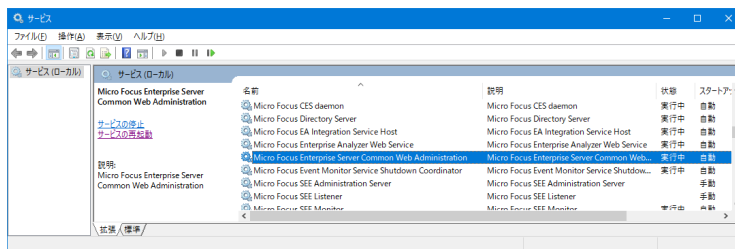
- 4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

- 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。

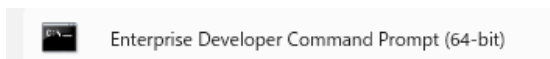


### 3.9 Enterprise Server インスタンスの設定

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。

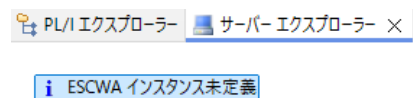
この情報は ESCWA のログオン時に使用しますので、記憶しておいてください。

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。

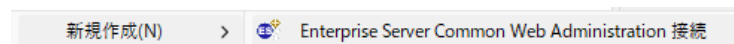
詳しくは製品マニュアルの【ここからはじめよう】>【Getting Started】にある【デフォルトセキュリティの構成】チュートリアルをご参照ください。

## 2) Enterprise Server インスタンスを作成します。

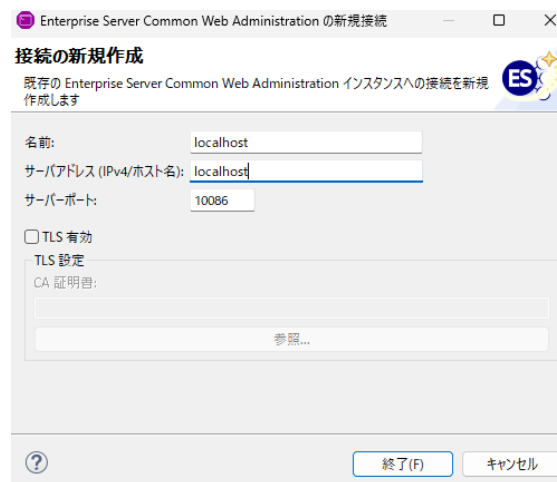
Eclipse に戻り、サーバーエクスプローラー内に【localhost】が表示されていない場合は、接続を作成します。存在する場合はこの項目はスキップしてください。



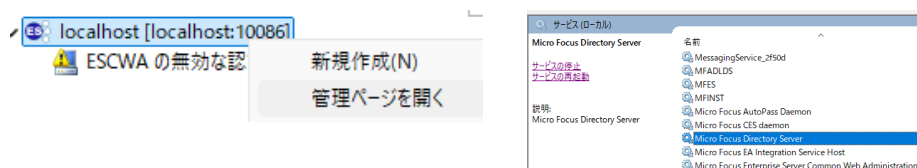
サーバーエクスプローラー内で右クリックし、【新規作成】>【Enterprise Server Common Web Administration 接続】を選択します。



【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。

TN3270 エミュレーターに設定されたコードページによって、自動的に半角英小文字が半角英大文字に変換されることがあります。この場合は、パスワードに含まれる半角英小文字と CICS サインオン時のパスワードが不一致となり、CICS にサインオンできません。

これを回避するために、新パスワードの英字はすべて大文字で入力してください。

良い新パスワードの例) SYSAD123

悪い新パスワードの例) sysad123

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、インストール時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](#)

認証情報は、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



Enterprise Server Administration

Rocket Software Enterprise Serverでは、インストール時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

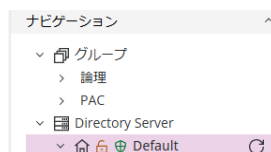
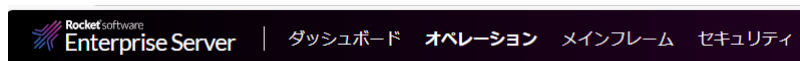
新しいパスワード\*  パスワードの確認\*

キャンセル [サブミット](#)

認証情報は、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



- 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。

ナビゲーション

- グループ
  - 論理
  - PAC
- Directory Server
  - Default

Directory Server [+ 追加](#)

**Directory Server**

名前\*

ホスト\*

ポート\*

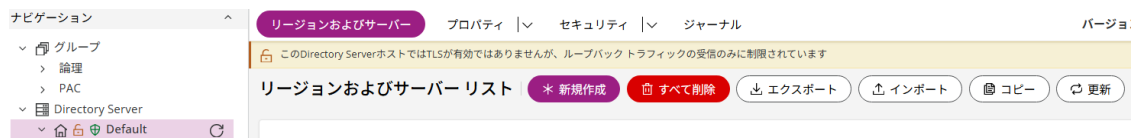
説明

\* 入力必須の項目です

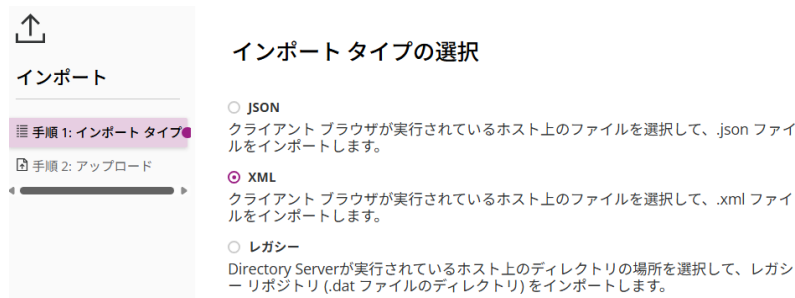
[保存](#) [戻る](#)

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。

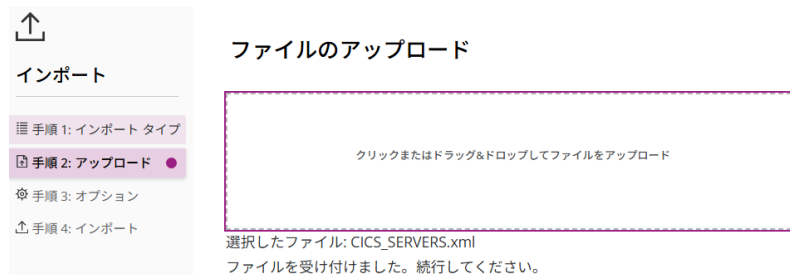
- 5) 例題のフォルダには Enterprise Server インスタンスのサンプルが含まれており、これをインポートします。  
 C:\¥work¥PLI¥CICSDEMO¥CICS\_SERVERS.xml がインポート対象のファイルです。  
 ESCWA の左側ペインで Default を選択し、右側ペインの [インポート] をクリックします。



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。



**インポート**

手順 1: インポート タイプ  
 手順 2: アップロード  
手順 3: インポート

### インポートの実行

インポートに成功しました。終了する場合は、「完了」を、他の xml ファイルをインポートする場合、「戻る」を押します。

- 6) 32 ビットアプリケーション稼働用の STAFF と、64 ビットアプリケーション稼働用の STAFF64 インスタンスが追加されます。Eclipse では 32 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 STAFF 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
	STAFF	Region	Stopped		✓	デフォルト	
	STAFF64	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。

ア...	名前	タ...	ステータス	64ビット	MSS有効	セキュ...	アクション
	STAFF	Region	Stopped		✓	デフォルト	
	STAFF64	Region	Stopped	✓	✓	デフォルト	

- 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。

一般的なプロパティ
適用
削除

#### 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

- 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。

一般的なプロパティ
適用
削除

#### 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

☐ 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

© Rocket Software, Inc. or its affiliates 1990–2025. All rights reserved. Rocket and the Rocket Software logos are registered trademarks of Rocket Software, Inc. Other product and service names might be trademarks of Rocket Software or its affiliates.

14

- 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

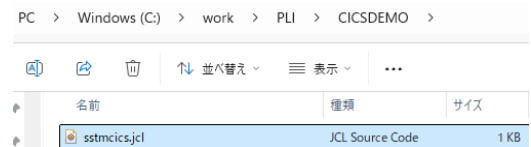
☒ MSS有効 ⓘ  
☒ JES有効 ⓘ  
☐ IMS有効 ⓘ  
☐ MQ有効 ⓘ

- 11) CICS から JCL を呼び出す場合は SSTM を有効にする必要があるため、次の初期化 JCL をテキストエディタなどで C:\work\PLI\CICSDemo に作成します。

名前) sstmccics.jcl

内容)

```
//SSTMJCL JOB 'CICS JOB',MSGCLASS=A
//CICS EXEC PGM=NOTUSED
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//TEMPJCL DD DSN=%%TEMPJCL,DISP=NEW
//OUTJCL DD DSN=CICS.SSTM.OUTJCL,DISP=(MOD,CATLG)
```



名前	種類	サイズ
sstmccics.jcl	JCL Source Code	1 KB

プログラム使用箇所の例)

```
EXEC CICS SPOOLOPEN OUTPUT
      TOKEN (WS-TOKEN)
```

```
EXEC CICS SPOOLWRITE
      TOKEN (WS-TOKEN)
      FROM (WS-JCL (WS-JIX))
```

- 12) [追加設定] の [構成情報] 欄に環境変数を設定します。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、前項で作成した JCL 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]
CICSDemo=C:\Users\Public\Documents\Rocket Software\Enterprise
Developer\Samples\PLI-VS or PLI-Eclipse\CICSDemo
ES_SSTM_CICS="$CICSDemo\sstmccics.jcl"
#MF_CHARSET=E
```

変更後;

```
[ES-Environment]
CICSDemo=C:\work\PLI\CICSDemo
ES_SSTM_CICS="$CICSDemo\sstmccics.jcl"
MFACCCGI_CHARSET=Shift_JIS
#MF_CHARSET=E
```

追加設定

構成情報 ⓘ

```
[ES-Environment]
CICSDemo=C:\work\PLI\CICSDemo
ES_SSTM_CICS="$CICSDemo\sstmccics.jcl"
MFACCCGI_CHARSET=Shift_JIS
#MF_CHARSET=E
```



#### 情報

ES\_SSTM\_CICS 環境変数:  
CICS 環境で JCL を使用する際に SSTM CICS 環境の初期化に使用される JCL の所在地を指定します。

### STAFF インスタンスの正常な SSTM 開始ログ)

```
JES000006I SSTM Startup Phase 1 (CICS) - processing "C:\work\PLI\CICSDemo\sstm\cics.jcl" 17:13:47
JES000007I SSTM Startup Phase 1 (CICS) - starting job J0001000 17:13:47
JCLCM0199I J0001000 SSTMJCL Program NOTUSED is COBOL VSC2 ASCII Big-Endian AMODE31. 17:13:47
JES000040I SSTM environment established successfully. (CICS) Job #: 0001000 17:13:47
```

- 13) 画面上部の【CICS】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
システム初期化テーブル(SIT)	CICS インスタンス設定の詳細が提供されるシステム初期化テーブルを指定します。
リソース定義ファイルパス	CICS リソース定義ファイルのパスを指定します。
トランザクションパス	実行される CICS プログラムの探索パスを指定します。
ファイルパス	FCT やプログラムからアクセスするファイルが存在するデフォルトパスを指定します。
マップパス	コンパイル済み BMS マップセットのパスを指定します。ここでは、MOD ファイルが生成されているパスを指定します。

**CICSの構成**
適用

システム初期化テーブル (SIT) ⓘ

STAFF

リソース定義ファイル パス ⓘ

\$CICSDemo\staff\_base

トランザクション パス ⓘ

\$CICSDemo\CICS\bin\x86\debug

ファイル パス ⓘ

\$CICSDemo\staff\_base

マップ パス ⓘ

\$CICSDemo\staff\_base\

☐ EZASOCKET サポート ⓘ

- 14) 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	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	カタログファイルのパスと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	JCL など指定するファイル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。

**JESの構成**
適用

JES プログラム パス ⓘ

\$CICSDemo\CICS\bin\x86\debug

システム カタログ ⓘ

\$CICSDemo\staff\_base\catalog.dat

データセットの省略時ロケーション ⓘ

\$CICSDemo\staff\_base\

システム プロシージャ ライブラリ ⓘ

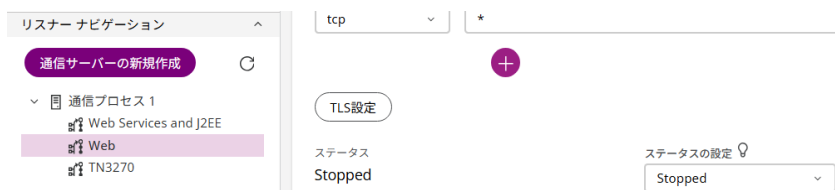
Fileshare 構成ロケーション ⓘ



- 15) 画面上部の【一般】プルダウンメニューから【リスナー】を選択し、TN3270 接続用のポート番号を確認します。例題では 5150 ポートを使用します。



- 16) 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】になっていることがあります。安全を確認したうえで、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。例題のリスナーが既に【Stopped】の場合はこれを確認します。



- 17) 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。

- ▼ Directory Server
- ▼ Default



### 重要

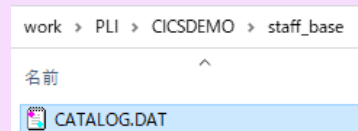
バージョン 7.0 では、パフォーマンス向上の観点から JES 関連ファイルである SPLJOB.DAT のフォーマットが改善されています。そのため、旧バージョンのファイルを 7.0 以降で利用する場合は mfsplcnv コマンドを使用して新フォーマットにコンバートする必要があります。コンバートを実行すると、古いフォーマットのファイルは SPLJOB.bak として保存されます。

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)

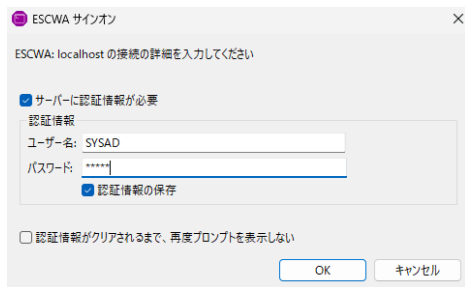
```
set MFSYSCAT= C:¥work¥PLI¥CICSDEMO¥staff_base¥catalog.dat
mfsplcnv -2
```

詳しくは製品マニュアルをご参照ください。

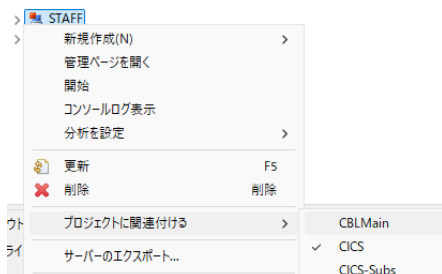


## 3.10 Enterprise Server インスタンスの開始と確認

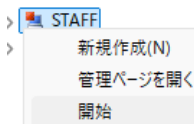
- 1) Eclipse に戻り、【localhost】をクリックすると ESCWA への認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して【OK】ボタンをクリックします。



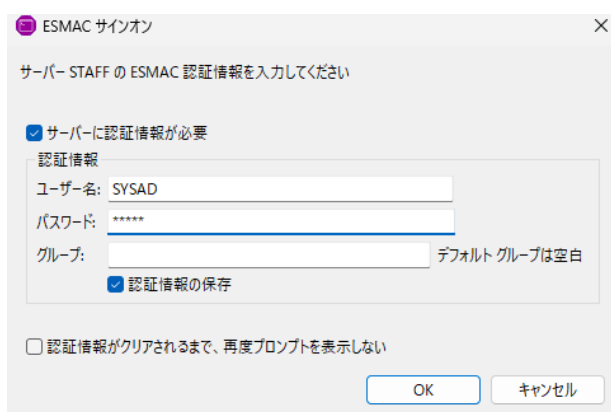
- 2) サーバーエクスプローラー内に STAFF インスタンスが表示されていることを確認します。表示されていない場合は [default] を右クリックし、[更新] を選択してリフレッシュしてください。
- 3) サーバーエクスプローラー内の STAFF インスタンスを右クリックし、[プロジェクトに関連付ける] > [CICS] を選択します。これにより Eclipse 内の CICS プロジェクトから実行されるプログラムは STAFF インスタンスで処理されることになります。



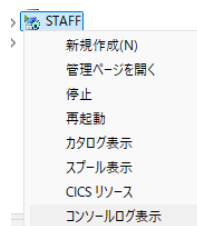
- 4) STAFF インスタンスを右クリックして [開始] を選択します。



- 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、STAFF 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。

コンソール × 問題 プロパティ Table Results Filter Definitions デバッグ				
Enterprise Server コンソールログ: localhost/Default/STAFF				
251217	17134775	9144	STAFF	CASFC0009I User (SYSAD) requested (OPEN) on file (DBAE) group (STAFF) from Tran (Unknown) Program
251217	17134776	9144	STAFF	CASSI5000I PLTPI Phase 1 - No PLT Specified 17:13:47
251217	17134777	9144	STAFF	CASSI5040I Active SEP memory strategy set to x'00000001', retain count 100 17:13:47
251217	17134909	976	STAFF	CASSI1600I SEP initialization completed successfully 17:13:49
251217	17134913	976	STAFF	ESFCA040I Caching disabled (cache size set to zero) 17:13:49
251217	17134918	14548	STAFF	CASB3000I Batch initiator started for job classes "ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789" 17:13:49
251217	17134929	976	STAFF	ESFEM130I ESM1: VSAM ESM Module initialized 17:13:49
251217	17134929	976	STAFF	ESFMI020I Loaded module vsam_esm for ESM (1) "VSAM ESM": VSAM ESM version 3.8.8 17:13:49
251217	17134931	14548	STAFF	ESFCA040I Caching disabled (cache size set to zero) 17:13:49
251217	17134932	976	STAFF	ESFPI010I External Security Manager version 3.8.7 initialized 17:13:49
251217	17134933	14548	STAFF	ESFEM130I ESM1: VSAM ESM Module initialized 17:13:49
251217	17134935	14548	STAFF	ESFMI020I Loaded module vsam_esm for ESM (1) "VSAM ESM": VSAM ESM version 3.8.8 17:13:49
251217	17134938	976	STAFF	JES000000I SSTH Startup Phase 2 (CICS) - starting job 17:13:49
251217	17134938	976	STAFF	JES000040I SSTH environment established successfully. (CICS) 17:13:49



## 注意

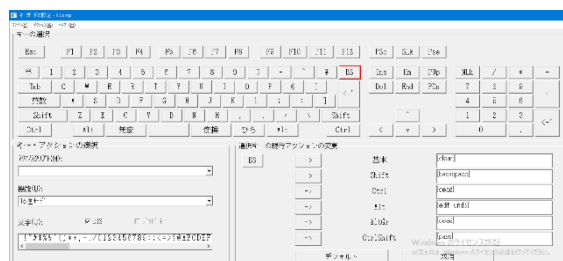
いくつかのサービス開始が失敗してもインスタンスは開始されますので、ログ内容を必ず確認してください。

## 3.11 CICS の実行

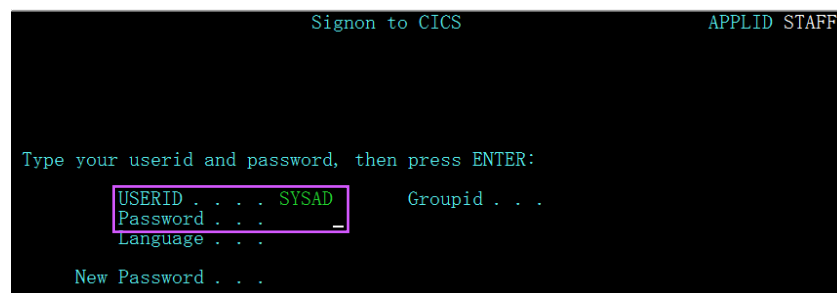
ご使用の TN3270 エミュレーターから、前項で確認した TN3270 リスナーポート(localhost:5150)へ接続します。

補足)クリアキーを使用するため、TN3270 エミュレーターのキーボード設定をご確認ください。

PCOMM の例)



- 1) 接続すると、例題の SIT では初期トランザクションに CESN が指定されているため、接続後 CICS サインオン画面が表示されます。[USERID]へ SYSAD を、[PASSWORD]へは前項で指定したパスワードを入力して実行キーを押します。



## SIT の設定)

CICSの構成 | **適用**

システム初期化テーブル (SIT) ⓘ  
STAFF

STAFF | **適用** | (コピー) | 削除

一般 | トレース ポイント | 基本マッピング

**一般**

説明 ⓘ  
SIT for testing Open PLI CICS

始動リスト ⓘ グループリスト ⓘ  
STAFF 開発用SIT ⓘ

☐ トランザクションクラスと優先度 ⓘ

WORK AREA (CWA) ⓘ 最小COMMAREA ⓘ  
512 0

SysID ⓘ 開始Tran ID ⓘ ローカルCCSID ⓘ  
\$IVP CESN 37

- 2) ログイン後、画面をクリアして PCT 名である TTEA を入力して実行キーを押すと、検索画面が表示されます。  
[Name] に WILSON と入力して実行キーを押下します。

```

Staff Directory - Search Criteria
Type a name, city, or ZIP.
Type name as last name, first name.

Name: WILSON_
City:
State: IL
ZIP:
    
```

## PCT の設定)

プログラム管理テーブル - TTEA (STAFF) | **適用** | インストール | コピー

**一般**

説明 ⓘ

TranClass ⓘ  
DFHTCL00

プログラム名 ⓘ リモート SysID ⓘ  
STAFF02

- 3) WILSON の情報が表示されます。表示されたデータは  
C:\¥work¥PLI¥CICSDEMO¥staff\_base¥DBA.dat ファイルから読み込まれたものです。

```

Staff Directory - Staff Information Summary

Name: WILSON, MICHAEL
Pref: MIKE

Designations: CLU, FIIC

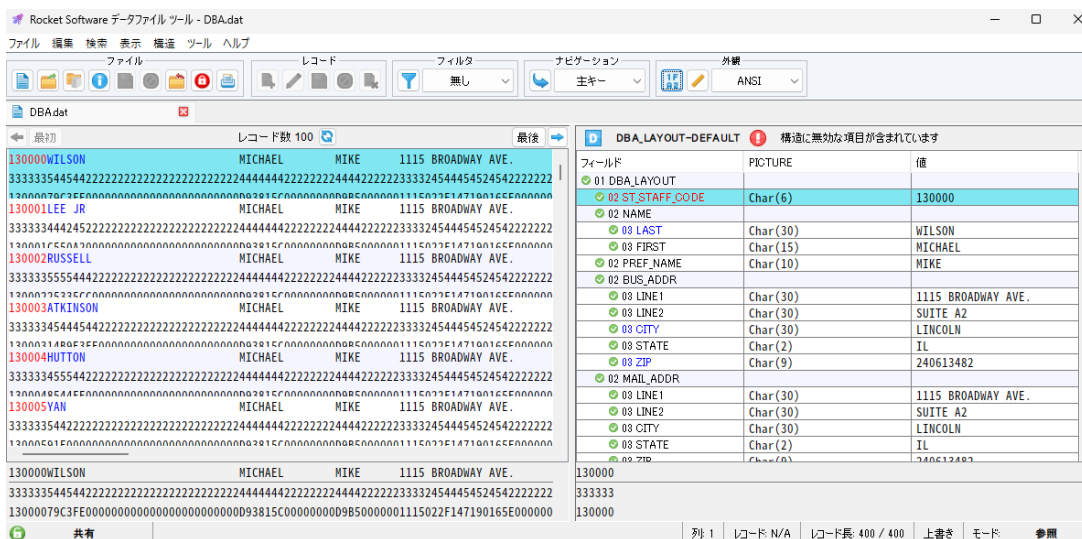
Mailing address: 1115 BROADWAY AVE.
                  SUITE A2
City: LINCOLN
State: IL ZIP: 24061-3482

Office address: 1115 BROADWAY AVE.
                  SUITE A2
City: LINCOLN
State: IL ZIP: 24061-3482

Phone: 000-000-0000
        000-000-0000

```

DBA.dat の内容)



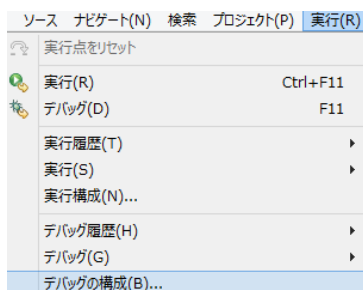
フィールド	PICTURE	値
01 DBA_LAYOUT		
02 ST_STAFF_CODE	Char(6)	130000
02 NAME		
03 LAST	Char(30)	WILSON
03 FIRST	Char(15)	MICHAEL
02 PREF_NAME	Char(10)	MIKE
02 BUS_ADDR		
03 LINE1	Char(30)	1115 BROADWAY AVE.
03 LINE2	Char(30)	SUITE A2
03 CITY	Char(30)	LINCOLN
03 STATE	Char(2)	IL
03 ZIP	Char(9)	240613482
02 MAIL_ADDR		
03 LINE1	Char(30)	1115 BROADWAY AVE.
03 LINE2	Char(30)	SUITE A2
03 CITY	Char(30)	LINCOLN
03 STATE	Char(2)	IL
02 ZIP	Char(9)	240613482

- 4) PCT の実行とファイルデータ読み込みまでを確認できましたので、TN3270 エミュレーターの接続を切断します。

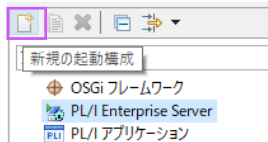
### 3.12 PL/I ソースのデバッグ

CICS から実行される PL/I プログラムをデバッグします。

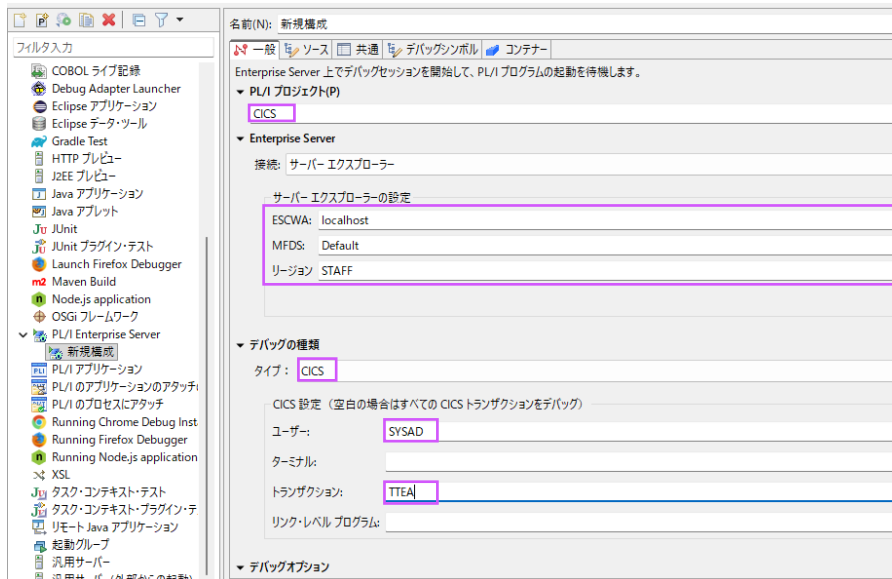
- 1) Eclipse の [実行] プルダウンメニューの [デバッグの構成] を選択します。



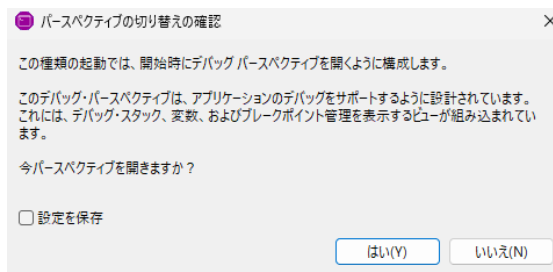
- 2) 左側のツリービューから [PL/I Enterprise Server] を選択して、左上の [新規の起動構成] アイコンをクリックします。



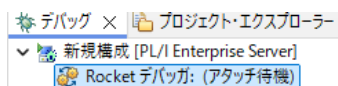
- 3) [PL/I プロジェクト] へ対象となる CICS プロジェクトを入力し、[PL/I Enterprise Server] へ実行させる STAFF インスタンスを指定します。[デバッグの種類] は CICS を選択し、[ユーザー] へは SYSAD を、[トランザクション] へは TTEA 入力した状態で、[デバッグ] ボタンをクリックします。



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

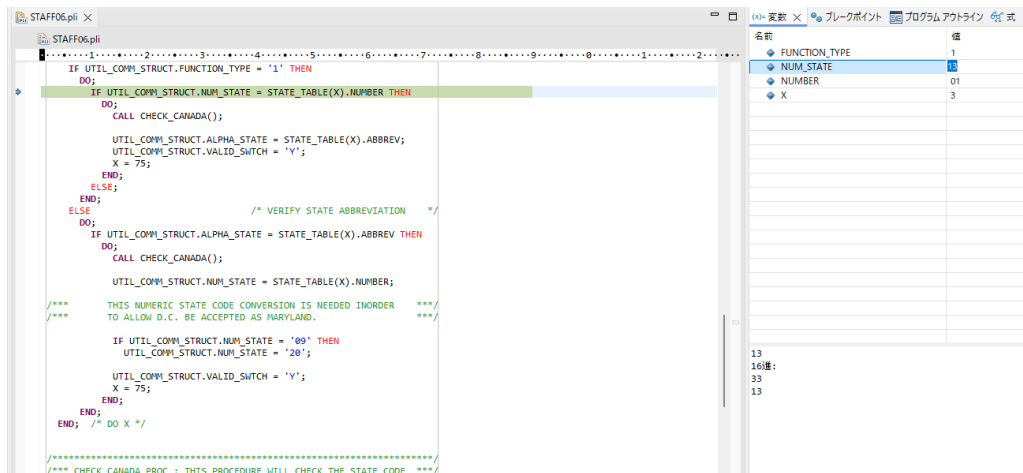


- 5) デバッグタブで [アタッチ待機] 状態になったことを確認します。

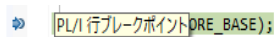


- 6) 前項と同様に TN3270 エミュレーターから接続後、ログインしてトランザクションを実行します。

- 7) 少し待つとデバッグセッションが開始され、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] や [式] タブでは使用している変数の値が確認できます。エミュレーターと画面を切り替えてデバッグします。



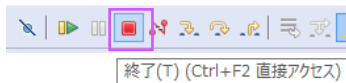
- 8) 希望のステップの左端をダブルクリックすることにより、ブレークポイントを設定することも可能です。



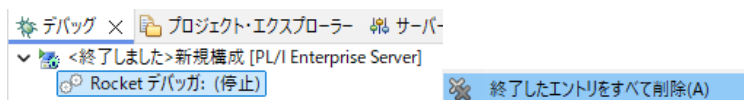
- 9) 先に進める場合は画面上部の再開アイコンをクリックします。



- 10) 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。



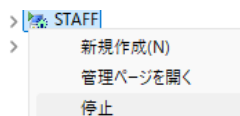
- 11) デバッガが停止状態になったのを確認後、右クリックして「終了したエントリをすべて削除」を選択し、これを削除します。



- 12) TN 3270 エミュレーターの接続を切断します。

### 3.13 終了処理

- 1) PL/I パースペクティブに戻り、サーバー エクスプローラー内で STAFF インスタンスを右クリックして「停止」を選択し、インスタンスを停止します。



- 2) STAFF 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。