

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

- Windows 開発環境に Enterprise Developer 9.0 for Eclipse がインストール済であること。
- 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

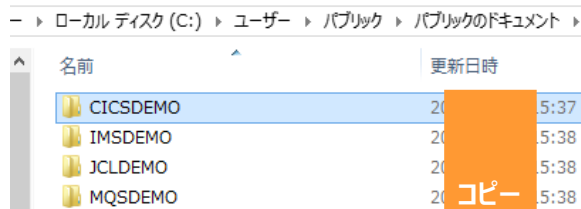
3.1 チュートリアルの準備

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。

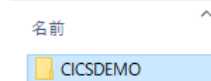
例)

C:¥Users¥Public¥Documents¥Micro Focus¥Enterprise Developer¥Samples¥PLI-Eclipse¥CICSDEMO

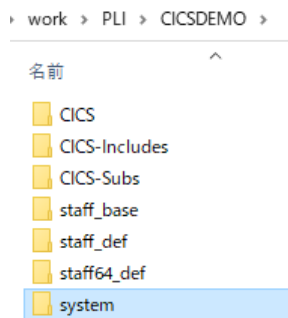


コピー先) C:¥work¥PLI¥CICSDEMO

スク (C:) > work > PLI

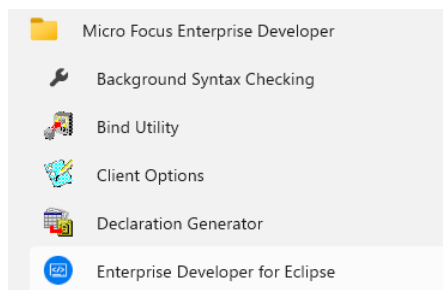


- 3) C:¥work¥PLI¥CICSDEMO フォルダ配下に、実行時に使用する system フォルダを Windows エクスプローラーを使用してあらかじめ用意しておきます。

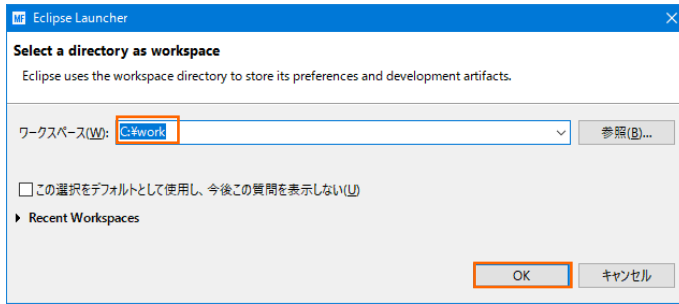


3.2 Eclipse の起動

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



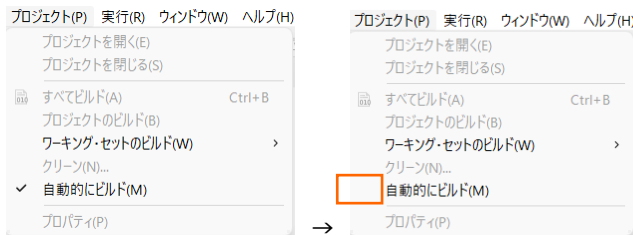
- 2) 前項で作成した C:¥work をワークスペースへ指定して、[OK] ボタンをクリックします。



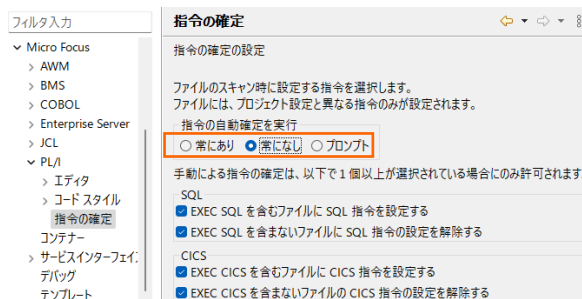
- 3) [ようこそ] タブが表示されますので、[Open PL/I Perspective] をクリックして、PL/I パースペクティブを開きます。



- 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。

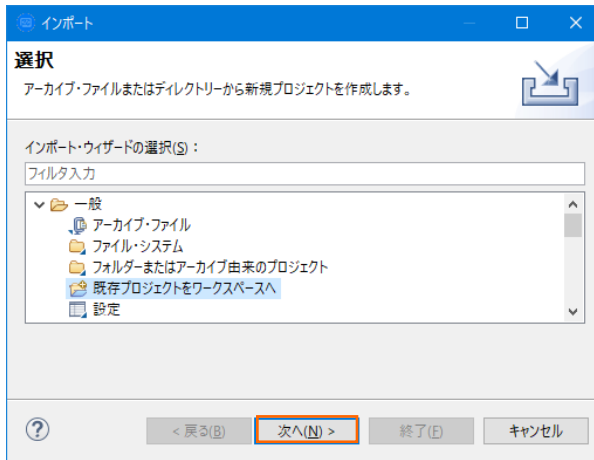


- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ウィンドウ] プルダウンメニューの [設定] > [Micro Focus] > [PL/I] > [指令の確定] > [指令の自動確定を実行] で [常になし] を選択し、[適用して閉じる] ボタンをクリックします。

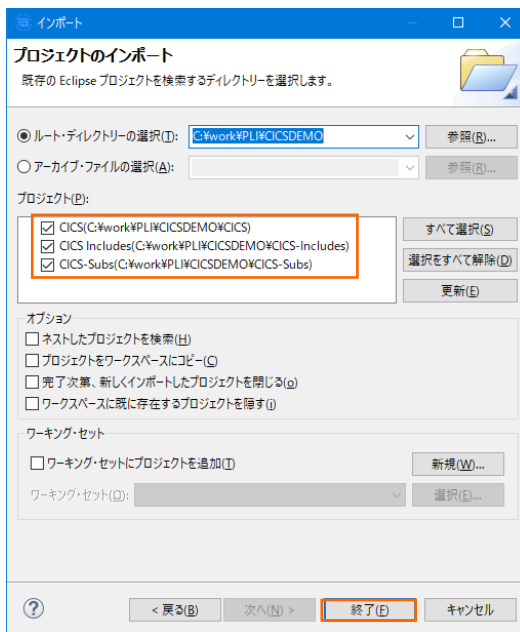


3.3 メインフレーム PL/I プロジェクトのインポート

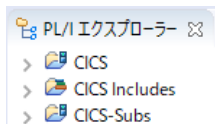
- 1) 用意した例題プロジェクトをインポートします。[ファイル] プルダウンメニューから [インポート] を選択し、インポートウィンドウにて [一般] > [既存プロジェクトをワークスペースへ] を選択後 [次へ] ボタンをクリックします。



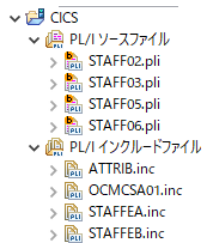
- 2) [ルート・ディレクトリの選択] へ C:¥work¥PLI¥CICSDemo を指定すると、このフォルダに含まれるプロジェクトが表示されます。全てのチェックをオンにした状態で [終了] ボタンをクリックします。



- 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。

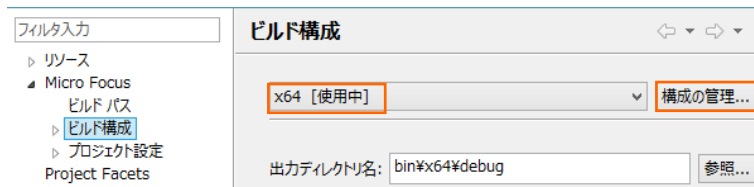


3.4 プロジェクトプロパティの確認

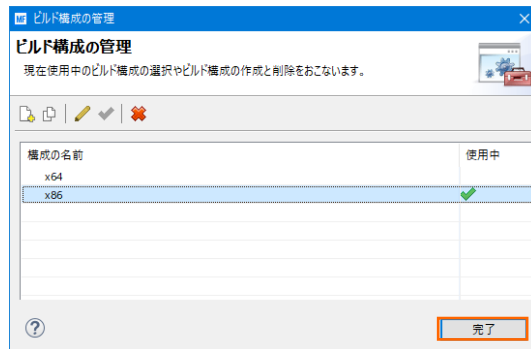
この例題では CICS-Subs プロジェクトで生成される lib オブジェクトを CICS プロジェクトがリンクして dll を生成します。プロジェクトの設定値を確認します。

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します。64 ビットが選択されている場合は下記の要領で変更してください。

- ① [Micro Focus] > [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。



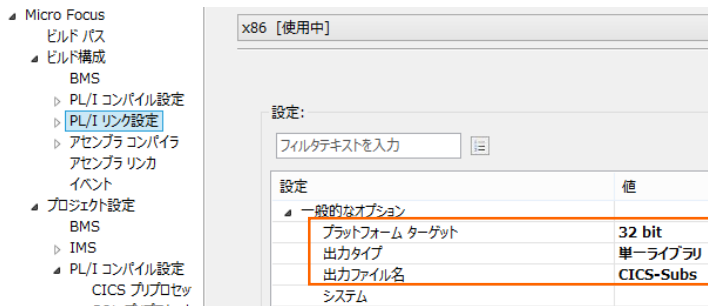
- ② [ビルドの構成管理] ウィンドウでは [x86] のチェックボックスをオンにして [完了] ボタンをクリックします。



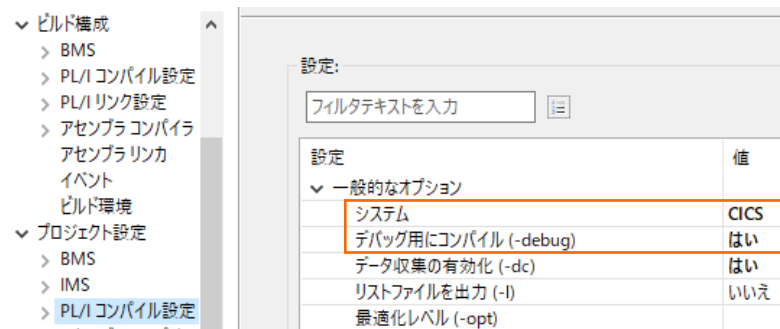
- ③ [Micro Focus] > [ビルド構成] ウィンドウへ戻り [x86] へ変更されたことと、プロジェクト配下の bin\debug フォルダへ実行ファイルが出力されることを確認後 [適用] ボタンをクリックします。



- 2) [Micro Focus] > [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) [Micro Focus] > [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(B)] を選択し、コンソールタブで成功を確認します。

> CICS-Subs **プロジェクトのビルド(B)**

combinedbuild.cfg.x86:

```
BUILD SUCCESSFUL
Build finished with no errors, 2 warnings.
```

> work > PLI > CICSDEMO > CICS-Subs > bin > debug

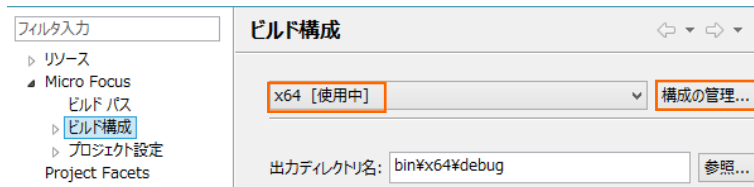
名前

CICS-Subs.lib

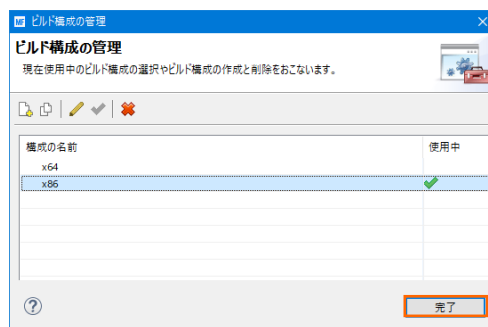
5) 次に CICS プロジェクトも同様に [プロパティ ウィンドウ] を表示します。

CICS-Subs プロジェクトと同じビット数にするために 32 ビット稼働とします。64 ビットが選択されている場合は下記の要領で変更してください。

- ① [Micro Focus] > [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。



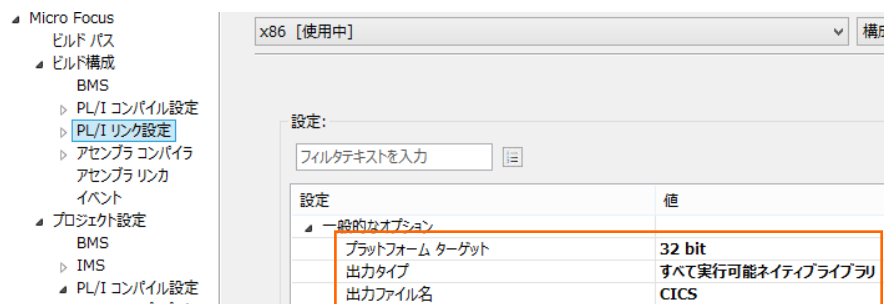
- ② [ビルドの構成管理] ウィンドウでは x86 のチェックボックスをオンにして [完了] ボタンをクリックします。



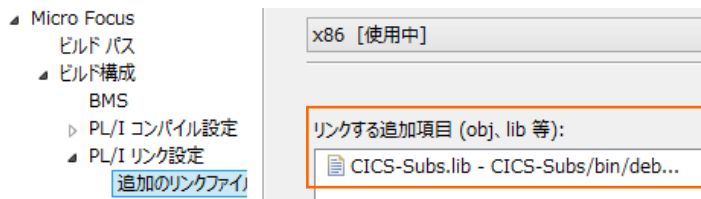
- ③ [Micro Focus] > [ビルド構成] ウィンドウへ戻り [x86] へ変更されたことと、プロジェクト配下の bin\debug フォルダへ実行ファイルが出力されることを確認後 [適用] ボタンをクリックします。



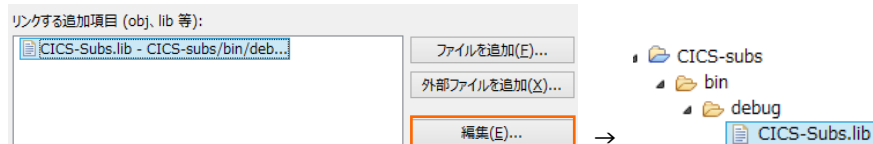
- 6) [Micro Focus] > [ビルド構成] > [PL/I リンク設定] を選択して内容を確認すると、CICS という名前で 32 ビット稼働する実行可能ネイティブライブラリとしてオブジェクトを生成することがわかります。



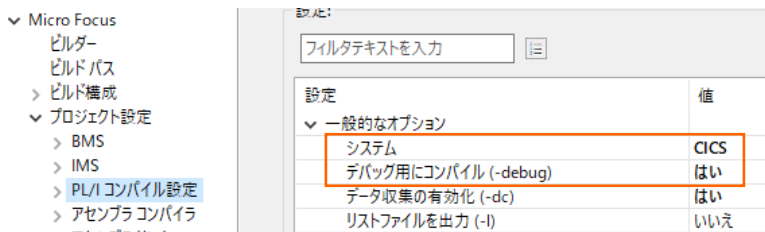
- 7) [Micro Focus] > [ビルド構成] > [PL/I リンク設定] > [追加のリンクファイル] を選択すると、「CICS-Subs.lib」をリンクするように指定していることがわかります。



- 8) CICS-Subs.lib を選択した状態で [編集] ボタンをクリックして、再度 CICS-Subs.lib を選択指定して [OK] ボタンをクリックします。前ウィンドウに戻りましたら [適用] ボタンをクリックします。

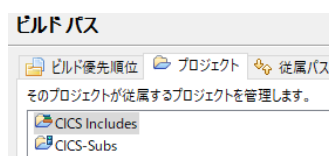


- 9) [Micro Focus] > [プロジェクト設定] > [PL/I コンパイル設定] を選択して内容を確認すると、例題の内容に沿って、「システム」には CICS が設定されており、デバッグ用のファイルも生成することがわかります。

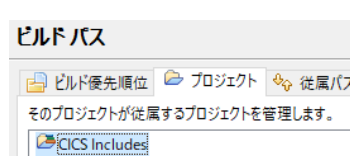


- 10) [Micro Focus] > [ビルドパス] の [プロジェクト] タブで従属プロジェクトを確認します。lib を生成している CICS-Subs プロジェクトと include ファイルが存在する CICS includes プロジェクトに従属していることが確認でき、これを基にコンパイルが実行されます。確認後は [適用して閉じる] ボタンをクリックしてください。

【CICS プロジェクト】



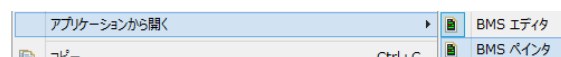
【CICS-sub project】



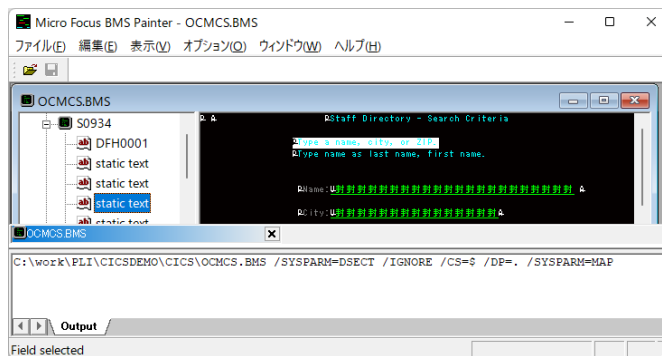
3.5 BMS 画面定義の確認

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。



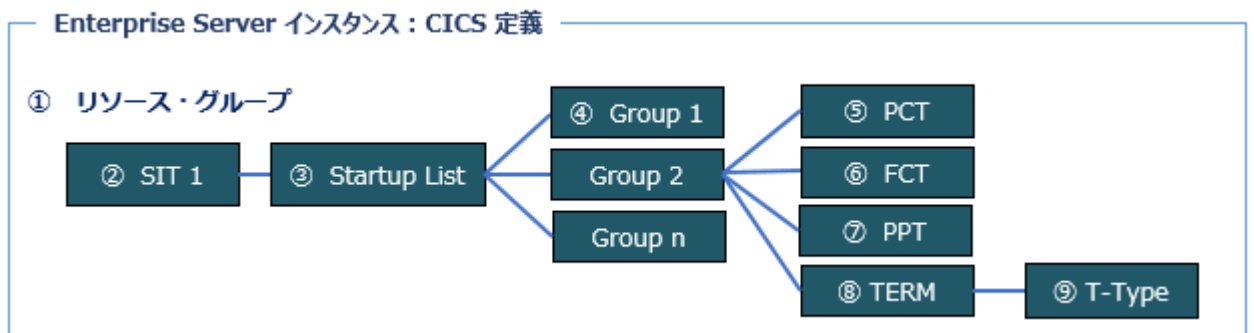
- 2) BMS ペインタウィンドウが表示され、画面定義内容をグラフィカルにメンテナンスできます。左側のツリービューでオブジェクトを選択すると右側のグラフィカルビュー内で対応するオブジェクトがハイライトされます。



- 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、COBOL 環境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
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
7.0.0.81 (C) Copyright 1984-2021 Micro Focus or one of its affiliates.

Windows SDK
Id バージョン 場所
0] 10.0.18362.0 c:\Program Files (x86)\Windows Kits\10
1] 10.0.19041.0 c:\Program Files (x86)\Windows Kits\10
2] 10.0.20348.0 c:\Program Files (x86)\Windows Kits\10

Microsoft Build Tools
Id バージョン 場所
0] 14.29.30133 c:\Program Files (x86)\Microsoft Visual Studio\2019\Professional
```

最新バージョンに設定するコマンド)

cblms -U

```
c:\xa>cblms -U
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
7.0.0.81 (C) Copyright 1984-2021 Micro Focus or one of its affiliates.

Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.20348.0

Microsoft Build Tools
場所 = c:\Program Files (x86)\Microsoft Visual Studio\2019\Professional
バージョン = 14.29.30133
```

特定のバージョンを指定するコマンド例)

cblms -US:1

```
c:\xa>cblms -US:1
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
7.0.0.81 (C) Copyright 1984-2021 Micro Focus or one of its affiliates.

Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.19041.0

c:\xa>cblms -UB:0
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
7.0.0.81 (C) Copyright 1984-2021 Micro Focus or one of its affiliates.

Microsoft Build Tools
場所 = c:\Program Files (x86)\Microsoft Visual Studio\2019\Professional
バージョン = 14.29.30133
```

COBOL 環境が使用するバージョンを表示するコマンド)

cblms -Q

```
c:\xa>cblms -Q
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
7.0.0.81 (C) Copyright 1984-2021 Micro Focus or one of its affiliates.

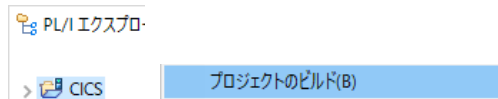
Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.19041.0

Microsoft Build Tools
場所 = c:\Program Files (x86)\Microsoft Visual Studio\2019\Professional
バージョン = 14.29.30133
```

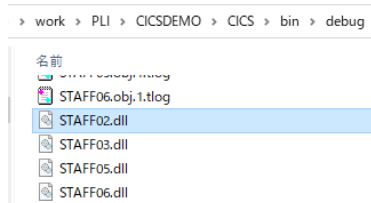
利用可能なオプションを表示するコマンド)

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¥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。
<http://www.microfocus.com/docs/links.asp?vc=cdctables>

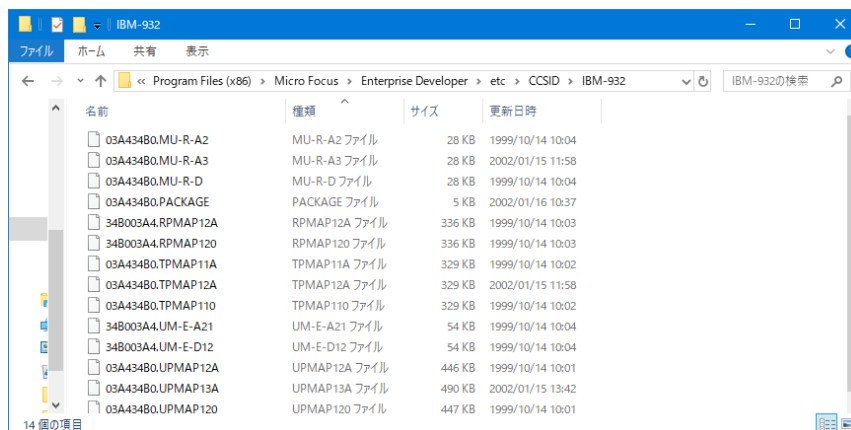
- 2) 製品インストールフォルダ配下の etc フォルダに CCSID フォルダがない場合はこれを作成します。

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

- 3) 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。

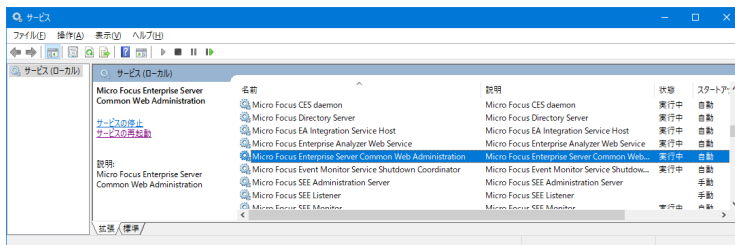
- 4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

- 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。



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。



3.9 Enterprise Server インスタンスの設定

- 1) PL/I を実行するためのエンジンを搭載した 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。



- 2) ブラウザが立ち上がり、Enterprise Server インスタンスを運用、管理する ESCWA が表示されます。



- 3) ESCWA では Micro Focus™ Directory Server サービスのポートへ接続して、登録されている Enterprise Server インスタンスを管理するため、他のマシンへの接続が必要な場合にはこれを作成します。

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。

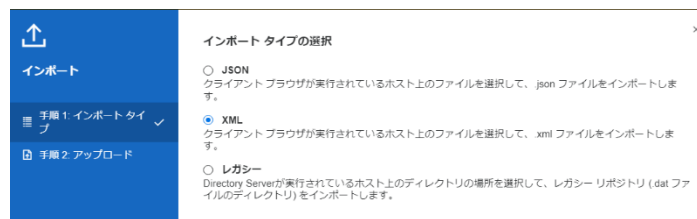


追加ウィンドウでは [名前] は任意ですが、例として default を、
 [ホスト] は localhost またはマシンの IP アドレスを、
 [ポート] は Micro Focus™ Directory Server の
 デフォルトポートである 86 を入力して [保存] ボタンをクリックします。

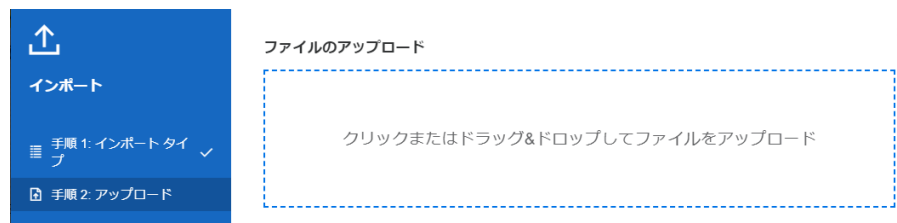
- 4) 例題のフォルダには Enterprise Server インスタンスのサンプルが含まれており、これをインポートします。
 C:\work\PLI\CICSDEMO\CICS_SERVERS.xml がインポート対象のファイルです。
 ESCWA の左側ペインで default を選択し、右側ペインの [インポート] をクリックします。



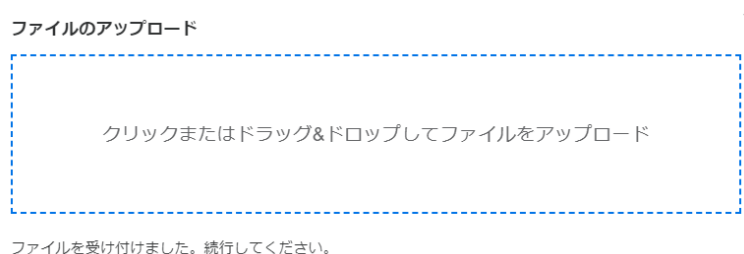
表示されたインポートウィンドウでは、[XML] を選択して [次へ] をクリックします。



[ファイルのアップロード] をクリックし、XML ファイルを指定します。



[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。

- 5) 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。

	名前	タイプ	ステータス	64ビット	MSS有効	セキュリティ
目	STAFF	Region	Stopped		✓	デフォルト
目	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。

- 6) 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。

名前	タイプ	ステータス	64ビット	MSS有効	セキュリティ	PAC
目	STAFF	Region	Stopped	✓	デフォルト	

  
編集

- 7) [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。

名前	システムディレクトリ
STAFF	C:\work\PLI\CICSDEMO\system

- 8) [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 type="checkbox"/> システム起動時に開始する
<input checked="" type="checkbox"/> 64ビット作業モード	<input type="checkbox"/> 以前のログを削除	

- 9) [リージョンの機能] では [MSS 有効], [JES 有効], [PL/I 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

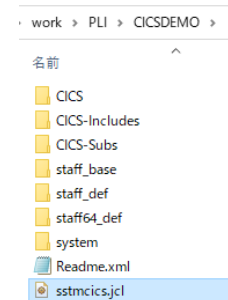
<input checked="" type="checkbox"/> MSS有効	<input checked="" type="checkbox"/> JES有効	<input type="checkbox"/> IMS有効	<input checked="" type="checkbox"/> PL/I有効
<input type="checkbox"/> MQ有効			

- 10) 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) sstmjcls.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)
```



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))
--	--

- 11) [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\Micro Focus\Enterprise
Developer\Samples\PLI-VS or PLI-Eclipse\CICSDEMO
ES_SSTM_CICS="$CICSDEMO\sstmciics.jcl"
#MF_CHARSET=E
```

変更後;

```
[ES-Environment]
CICSDEMO=C:\work\PLI\CICSDEMO
ES_SSTM_CICS="$CICSDEMO\sstmciics.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\sstmciics.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	I	SSTM Startup Phase 1 (CICS) - processing "C:\work\PLI\CICSDEMO\sstmciics.jcl"
JES000007I	I	SSTM Startup Phase 1 (CICS) - starting job J0001000
JCLCM0199I	I	J0001000 SSTMJCL Program NOTUSED is COBOL VSC2 ASCII Big-Endian AMODE31.
JES000040I	I	SSTM environment established successfully. (CICS) Job #: 0001000

12)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\debug	
ファイルパス ⓘ \$CICSDEMO\staff_base	
マップパス ⓘ \$CICSDEMO\staff_base\	

13)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などで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。

14)画面上部の [一般] プルダウンメニューから [PL/I] を選択し、表示される画面の各項目を確認します。

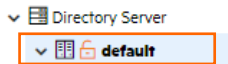
項目名	説明
PL/I 有効	[PL/I] タブ配下の有効設定をオン、オフ指定します。
Codewatch ソース パス	CodeWatch デバッガで使用するソースファイルが存在するパスを指定します。
Codewatch STB パス	CodeWatch デバッガで使用するデバッグファイルが存在するパスを指定します。
PL/I 構成ディレクトリ	プロジェクトのパスを指定します。

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。



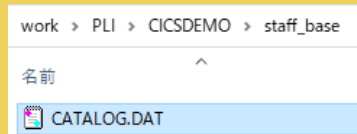
重要

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)

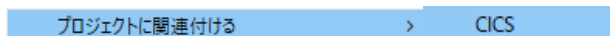
```
set MFSYSCAT= C:¥work¥PLI¥CICSDEMO¥staff_base¥catalog.dat
mfsplcnv -2
```



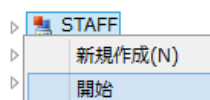
詳しくは製品マニュアルをご参照ください。

3.10 Enterprise Server インスタンスの開始と確認

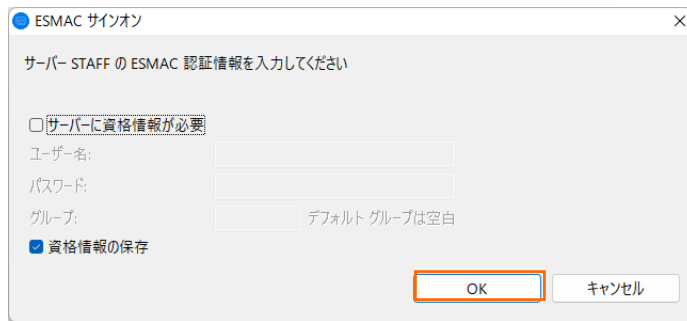
- 1) Eclipse に戻り、サーバーエクスプローラー内に STAFF インスタンスが表示されていることを確認します。表示されていない場合は [default] を右クリックし、[更新] を選択してリフレッシュしてください。
- 2) サーバーエクスプローラー内の STAFF インスタンスを右クリックし、[プロジェクトに関連付ける] > [CICS] を選択します。これにより Eclipse 内の CICS プロジェクトから実行されるプログラムは STAFF インスタンスで処理されることとなります。



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



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



ESMAC サインオン

サーバー STAFF の ESMAC 認証情報を入力してください

サーバーに資格情報が必要

ユーザー名:

パスワード:

グループ: デフォルト グループは空白

資格情報の保存

OK キャンセル

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。

名前	タイプ	ステータス	64ビット	MSS有効	セキュリティ	PAC
STAFF	Region	Started		✓	デフォルト	

編集

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、正常に開始されたか確認します。

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。

コンソール | バックアップ 現行

メッセージID ↓ メッセージ ↓ プロセスID ↓ ログレベル ↓

プロセスID	メッセージID	ログレベル	メッセージ
10748	CASTS5116I	I	ES TSC Log facility file size: 4 blocks (316,236 bytes)
10748	CASTS5117I	I	ES TSC Log facility is running with 4 buffers of 63,020 bytes each
	CASCD1060I	I	JES Initiator created for Server STAFF, process-id = 10936
11268	CASBJ0023I	I	Batch initiator INIT1: class(es) "abcdefghijklmnopqrstuvwxyz0123456789"
	CASCD0127I	I	SEP 00001 created for ES STAFF, process-id = 8296
	CASCD0127I	I	SEP 00002 created for ES STAFF, process-id = 8080
10748	CASTS0002I	I	ES TSC Service Process initialization complete
7188	CASCS5001I	I	Communications interface 01 initialization started
8080	CASSI1500I	I	SEP initialization started
10936	CASBJ0006I	I	Batch initiator initialization started
8296	CASSI1500I	I	SEP initialization started
7188	CASCS5003I	I	Communications interface 01 initialization complete
	CASCD1071I	I	Administration SEP created for Server STAFF, process-id = 11548
7188	CASCS5100I	I	Communications Process instance 01 is ready to accept requests



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。

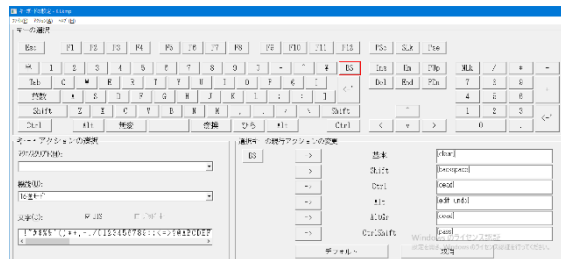
7) 画面左側ペインの [default] をクリックして一覧画面に戻ります。

3.11 CICS の実行

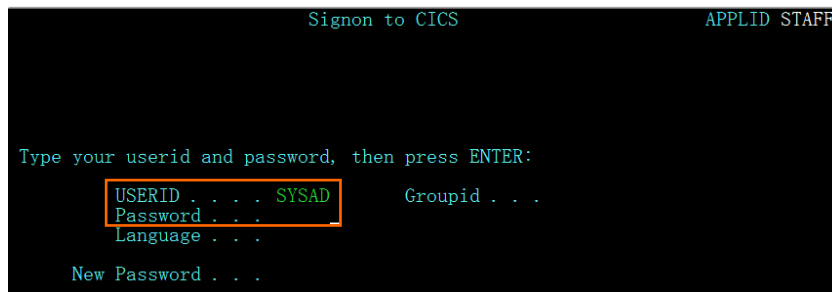
ご使用の TN3270 エミュレーターから、前項で確認した TN3270 リスナーポート(localhost:5150)へ接続します。

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)



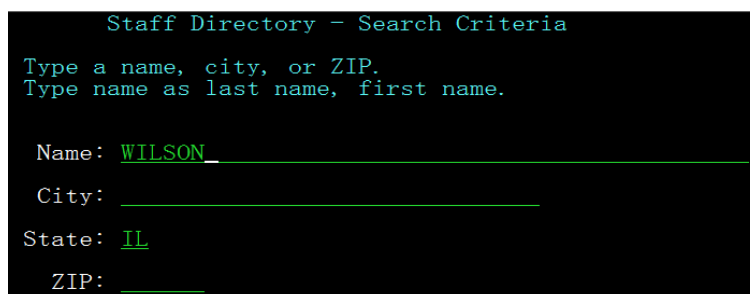
- 1) 接続すると、例題の SIT では初期トランザクションに CESN が指定されているため、接続後 CICS サインイン画面が表示されます。[USERID] と [PASSWORD] へ SYSAD を入力して実行キーを押します。



SIT の設定)



- 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。



PCT の設定)



- 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

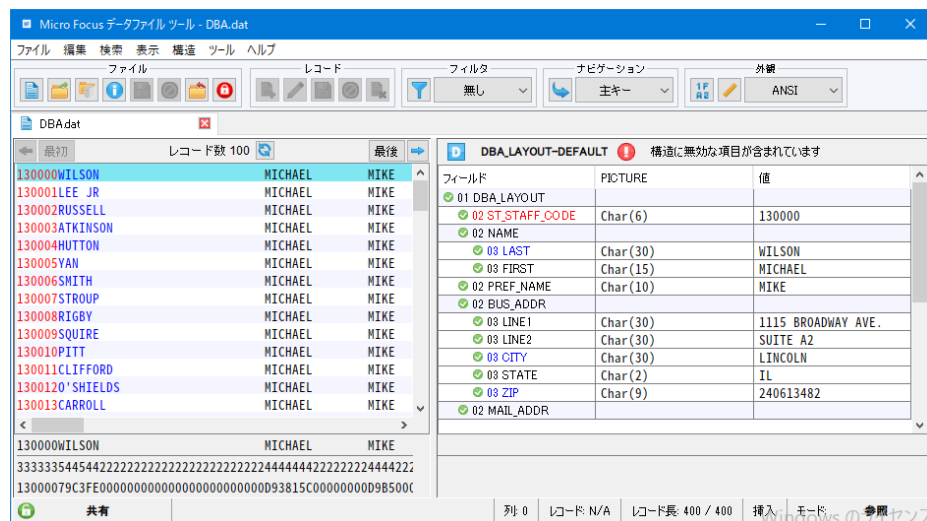
Designations: CLU, FIIC

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)

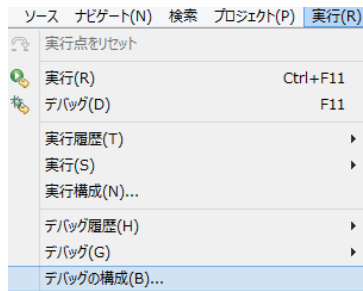


- 4) PCT の実行とファイルデータ読み込みまでを確認できましたので、TN 3270 エミュレーターの接続を切断します。

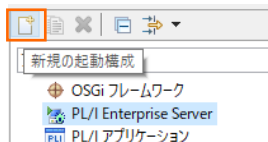
3.12 PL/I ソースのデバッグ

CICS から実行される PL/I プログラムをデバッグします。

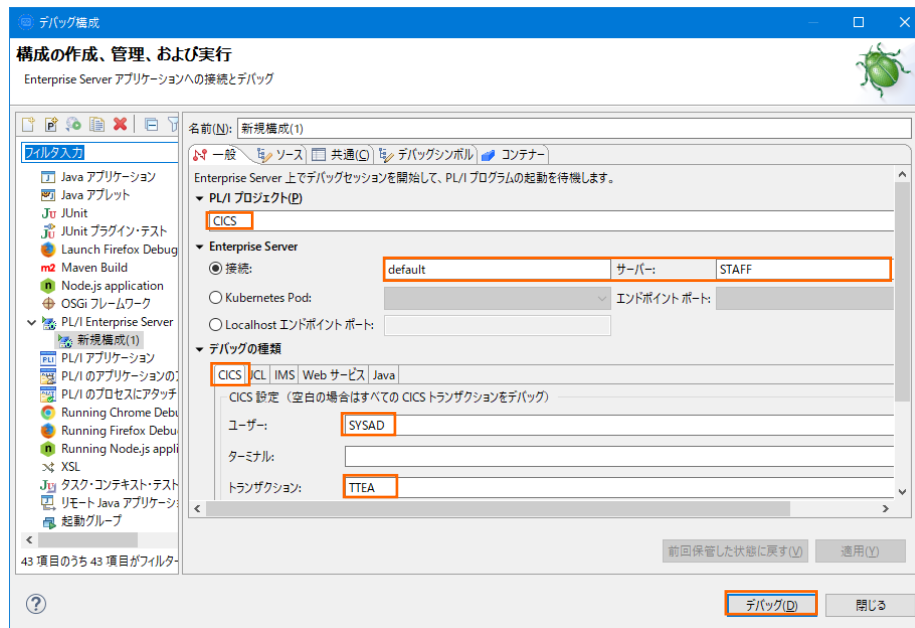
- 1) Eclipse の [実行] プルダウンメニューの [デバッグの構成] を選択します。



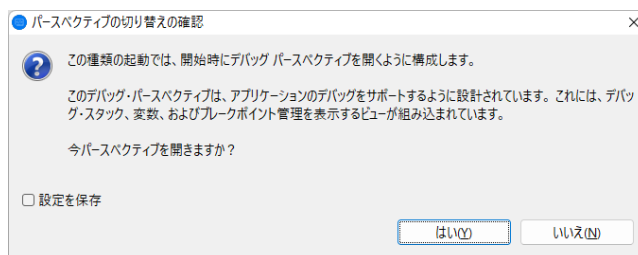
- 2) 左側のツリービューから [PL/I Enterprise Server] を選択して、左上の [新規の起動構成] アイコンをクリックします。



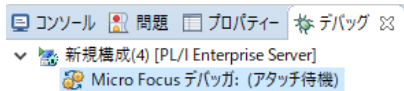
- 3) [PL/I プロジェクト] へ対象となる CICS プロジェクトを入力し、[PL/I Enterprise Server] へ実行させる STAFF インスタンスを指定します。[デバッグの種類] は CICS タブを選択し、[ユーザー] へは SYSAD を、[トランザクション] へは TTEA 入力した状態で、[デバッグ] ボタンをクリックします。



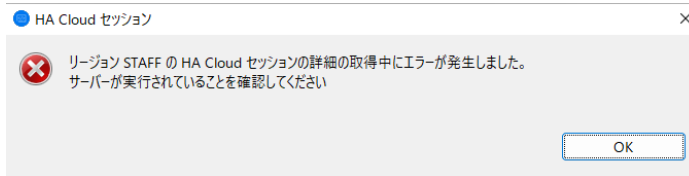
- 4) パースペクティブの切り替え確認ウィンドウでは [はい] をクリックします。



- 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。

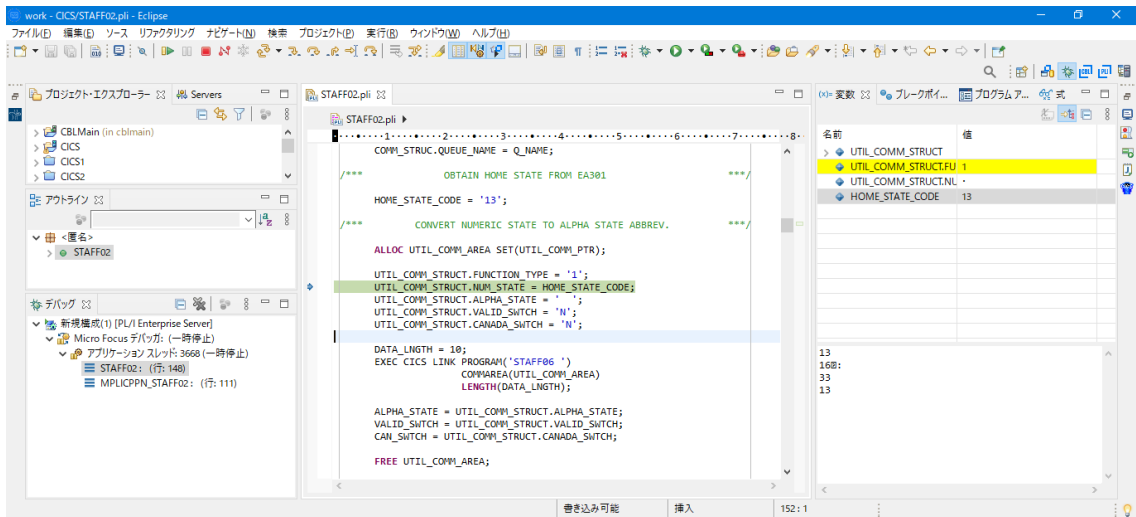


- 6) 製品に含まれる HA Cloud セッションを開始していないため、次のメッセージが表示された場合は、このまま [OK] ボタンをクリックします。

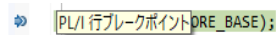


- 7) 前項と同様に TN3270 エミュレーターから接続後、ログインしてトランザクションを実行します。

- 8) 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。



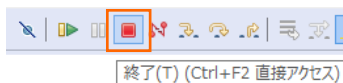
- 9) 希望のステップの左端をダブルクリックすることにより、ブレークポイントを設定することも可能です。



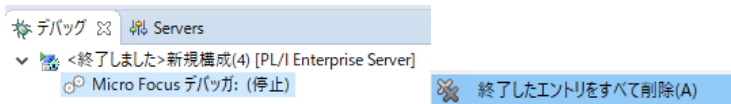
- 10) 先に進める場合は画面上部の再開アイコンをクリックします。



- 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。



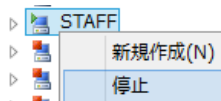
12) デバッガが停止状態になったのを確認後、右クリックして [終了したエントリをすべて削除] を選択し、これを削除します。



13) 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) 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、開始中の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 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。