Micro Focus 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 4.0 for Eclipse がインストール済であること。
- Micro Focus Rumba などの 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. ビルドの実行
6. Enterprise Server インスタンスの設定
7. Enterprise Server インスタンス開始と確認
8. CICS の実行
9. PL/I ソースのデバッグ
10. 終了処理

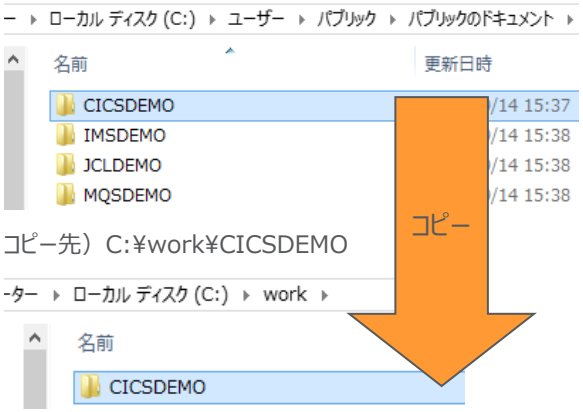
3.1 チュートリアルの準備

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 work フォルダを C ディレクトリ直下に作成します。

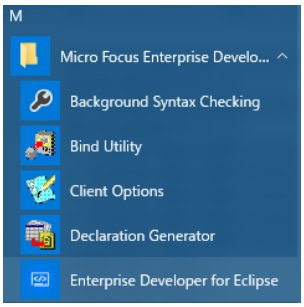
2) 製品をインストールしたフォルダ配下に含まれている下記の例題プログラム（CICSDEMO フォルダ）を、作成した C:¥work へコピーします。

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

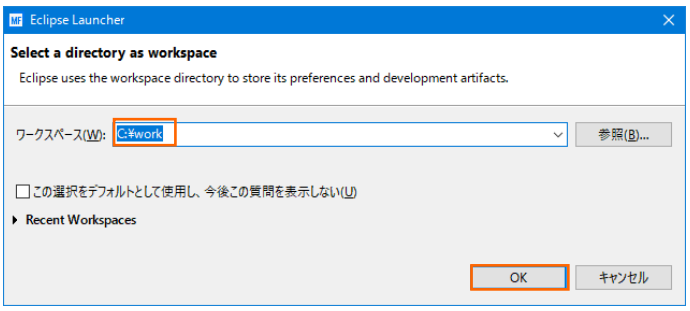


3.2 Eclipse の起動

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

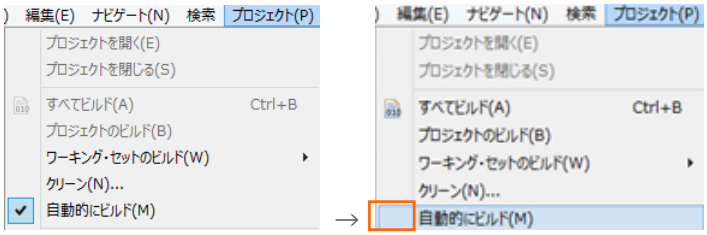


2) 前項で作成した C:¥work をワークスペースへ指定して、[OK] ボタンをクリックします。

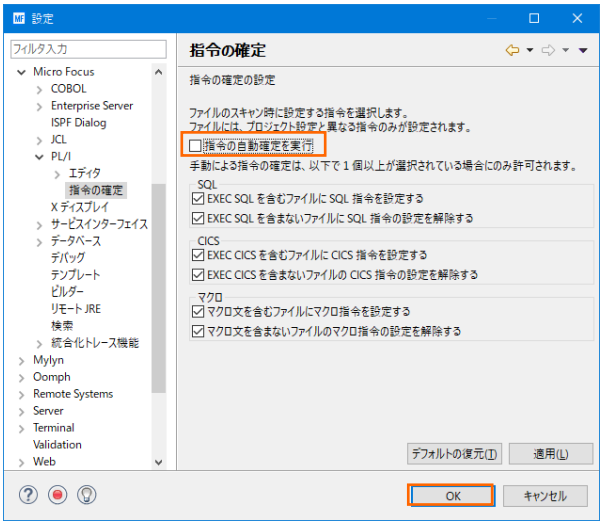


3) [ようこそ] タブが表示されますので、[Open PL/I Perspective] をクリックして、PL/I パースペクティブを開きます。

4) PL/I 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。

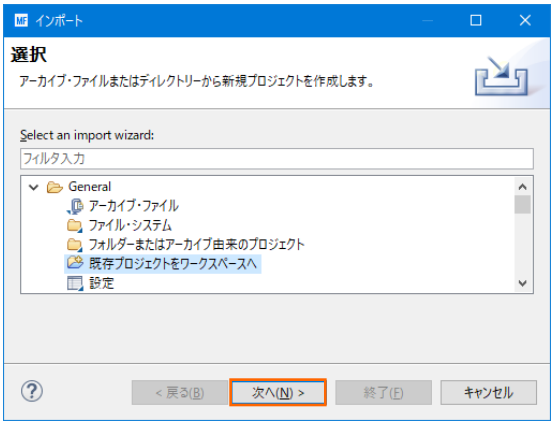


5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。 [ウィンドウ] プロダウンメニューの [設定] > [Micro Focus] > [PL/I] > [指令の確定] > [指令の確定を行う] チェックボックスをオフにして [OK] ボタンをクリックします。

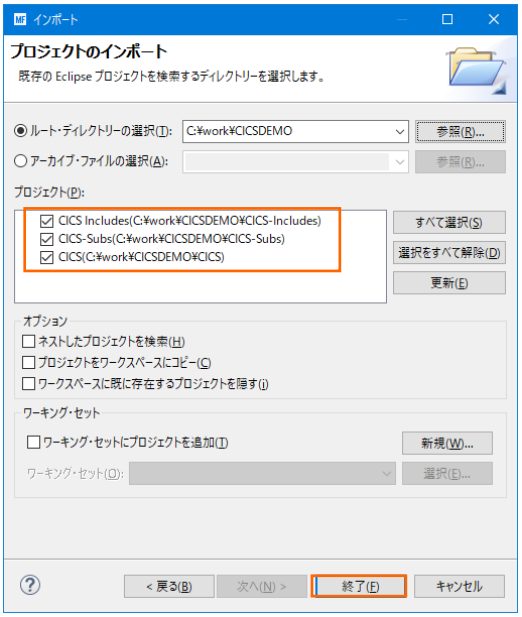


3.3 メインフレーム PL/I プロジェクトのインポート

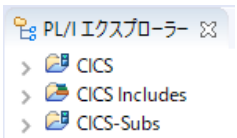
1) 用意した例題プロジェクトをインポートします。 [ファイル] プルダウンメニューから [インポート] を選択し、インポートウィンドウにて [General] > [既存プロジェクトをワークスペースへ] を選択後 [次へ] ボタンをクリックします。



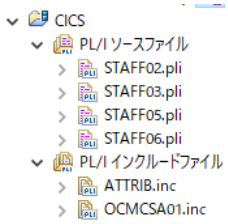
2) [ルート・ディレクトリの選択] へ C:¥work¥CICSDEMO を指定すると、このフォルダに含まれるプロジェクトが表示されます。チェックをオンにした状態で [終了] ボタンをクリックします。



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。



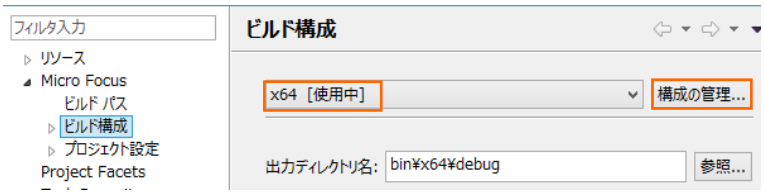
3.4 プロジェクトプロパティの確認

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 を生成する内容になっています。プロジェクトの設定値を確認していきます。

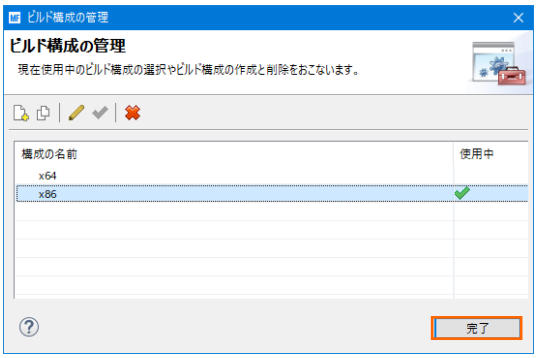
1) CICS-Subs プロジェクトを右クリックし、[プロパティ] を選択すると [プロパティ ウィンドウ] が表示されます。

64 ビット稼働が指定されていますが、ここでは 32 ビット OS で実行することを考慮して 32 ビット稼働へ変更します。

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。

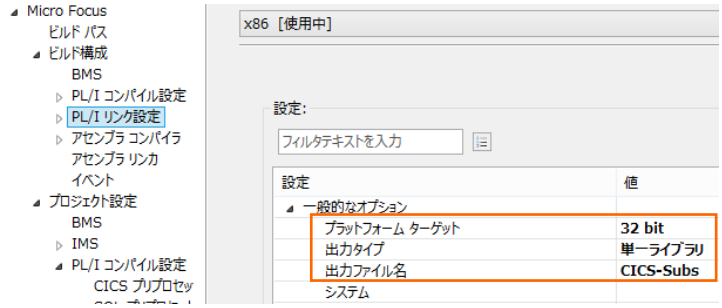


② [ビルドの構成管理] ウィンドウでは [x86] のチェックボックスをオンにして [完了] ボタンをクリックします。

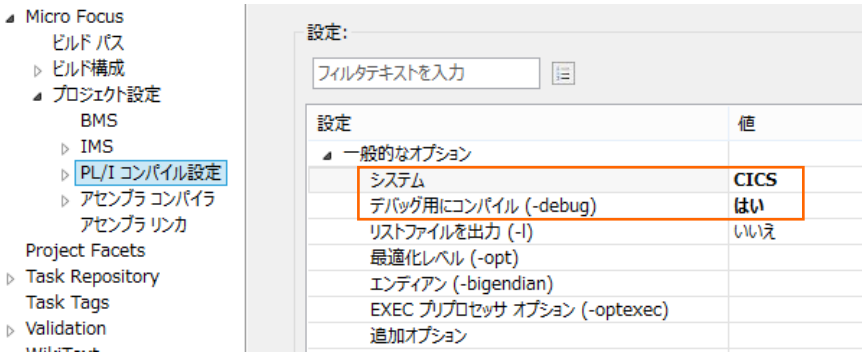


③ [Micro Focus] > [ビルド構成] ウィンドウへ戻り [x86] へ変更されたことと、プロジェクト配下の bin¥debug フォルダへ実行ファイルが出力されることを確認後 [適用] ボタンをクリックします。

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 ビット稼働する単一ライブラリ、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 [Apply and Close] 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] を選択します。

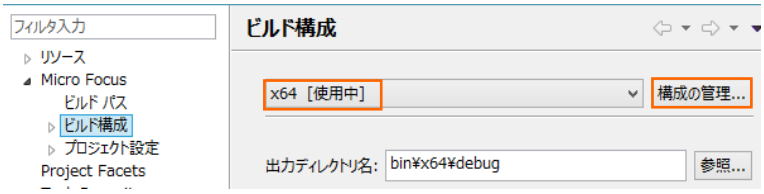
```
combinedbuild.cfg.x86:

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

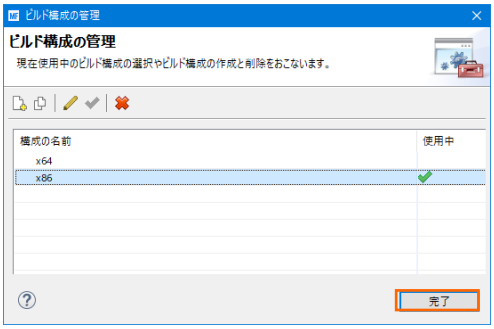
5) 次に CICS プロジェクトも同様に [プロパティ ウィンドウ] を表示します。

64 ビット稼働が指定されていますが CICS-Subs プロジェクトと同じビット数にする必要がありますので、32 ビット稼働へ変更します。

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。

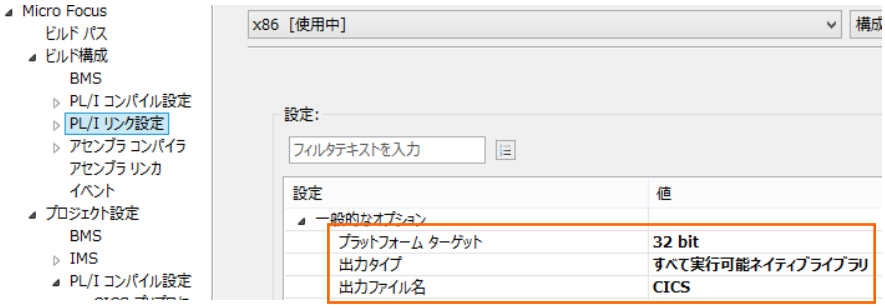


② [ビルドの構成管理] ウィンドウでは x86 のチェックボックスをオンにして [完了] ボタンをクリックします。

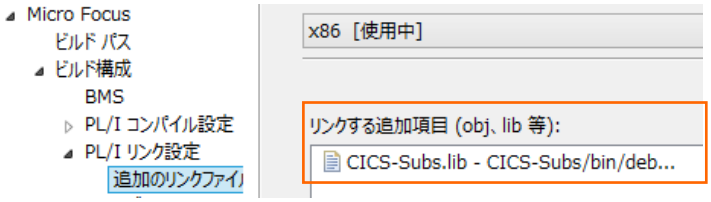


③ [Micro Focus] > [ビルド構成] ウィンドウへ戻り [x86] へ変更されたことと、プロジェクト配下の bin¥debug フォルダへ実行ファイルが出力されることを確認後 [適用] ボタンをクリックします。

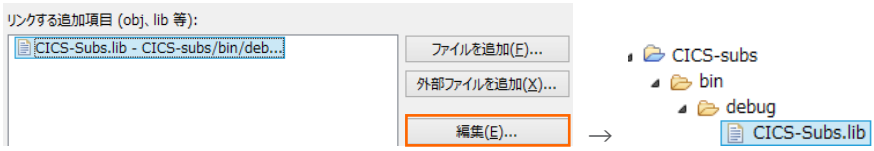
6) [Micro Focus] > [ビルド構成] > [PL/I リンク設定] を選択して内容を確認すると、CICS という名前で 32 ビット稼働する実行可能ネイティブライブラリとしてオブジェクトを生成することがわかります。



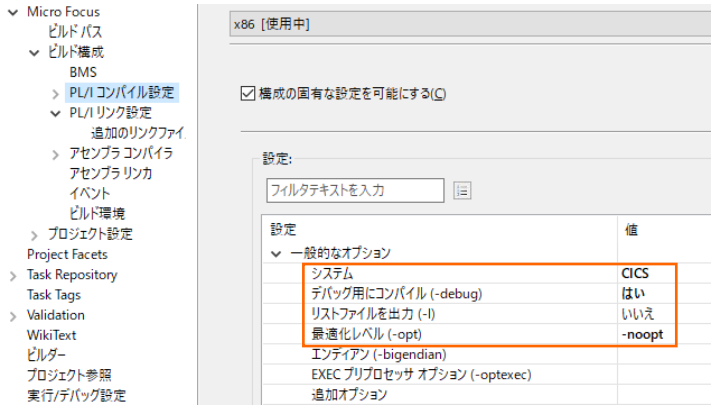
7) [Micro Focus] > [ビルド構成] > [PL/I リンク設定] > [追加のリンクファイル] を選択すると、「CICS-Subs.lib」をリンクするように指定していることがわかります。



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。

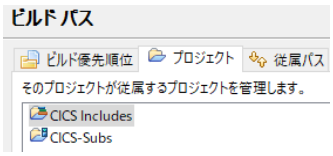


9) [Micro Focus] > [ビルド構成] > [PL/I コンパイル設定] を選択して内容を確認すると、例題の内容に沿って、「システム」には CICS が設定されており、デバッグ用のファイルも生成することがわかります。

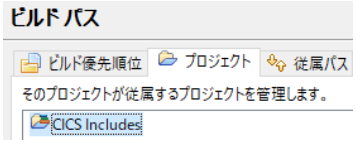


10) [Micro Focus] > [ビルドパス] の [プロジェクト] タブで従属プロジェクトを確認します。 lib を取り込む CICS-Subs プロジェクトと、共通で使用する include ファイルが存在する CICS includes プロジェクトに従属していることが確認でき、これを基にコンパイルが実行されます。確認後は [Apply and Close] ボタンをクリックしてください。

【CICS プロジェクト】

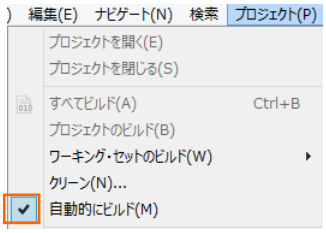


【CICS-subs プロジェクト】



3.5 ビルドの実行

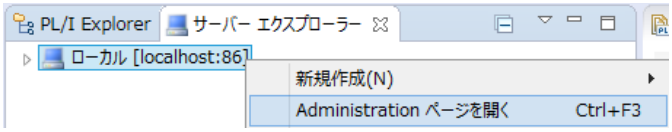
1) [プロジェクト] プルダウンメニューの [自動的にビルド] を選択して、これをオンすると自動的にビルドが実行されます。



2) 各プロジェクトの bin¥debug フォルダ配下に目的の lib と dll が作成されていることを確認してください。

3.6 Enterprise Server インスタンスの設定

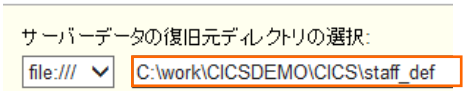
1) PL/I を実行するためのエンジンを搭載した Enterprise Server インスタンスを作成します。 サーバー エクスプローラータブの [ローカル] を右クリックして [Administration ページを開く] を選択します。



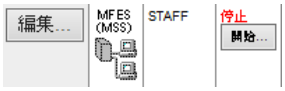
2) C:¥work¥CICSDEMO¥CICS には Enterprise Server インスタンスのサンプルが含まれており、これをインポートします。PL/I アプリケーションは 32 ビット稼働を指定したため、C:¥work¥CICSDEMO¥CICS¥staff_def がインポート対象となります。64 ビットで稼働させる場合は C:¥work¥CICSDEMO¥CICS¥staff64_def をインポートしてください。

Enterprise Server Administration 画面左側メニューの [インポート] をクリックして、表示される下記項目へ前述のパスを入力後、 [次へ] ボタンをクリックします。

サーバー情報のインポート (Page 1 of 4):



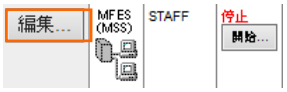
3) 画面の Page 2/4、3/4、ではそのまま [次へ] ボタンを、Page 4/4 では [OK] ボタンをクリックすると、STAFF という名前の 32 ビットアプリケーション稼働用 Enterprise Server インスタンスが追加されます。



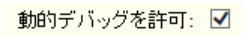
重要

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。

4) 設定を変更するため、[編集] ボタンをクリックします。



5) [サーバー] > [プロパティ] > [一般] タブで表示される画面の [動的デバッグを許可] 欄のチェックをオンにします。これにより、Eclipse からのデバッグが行えます。



6) [サーバー] > [プロパティ] > [一般] タブで表示される画面の [構成情報] 欄を下記のように入力し、[Apply] 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¥CICS¥sstmcics.jcl"
CODEWATCH_NOTIF=Y
```

変更後；

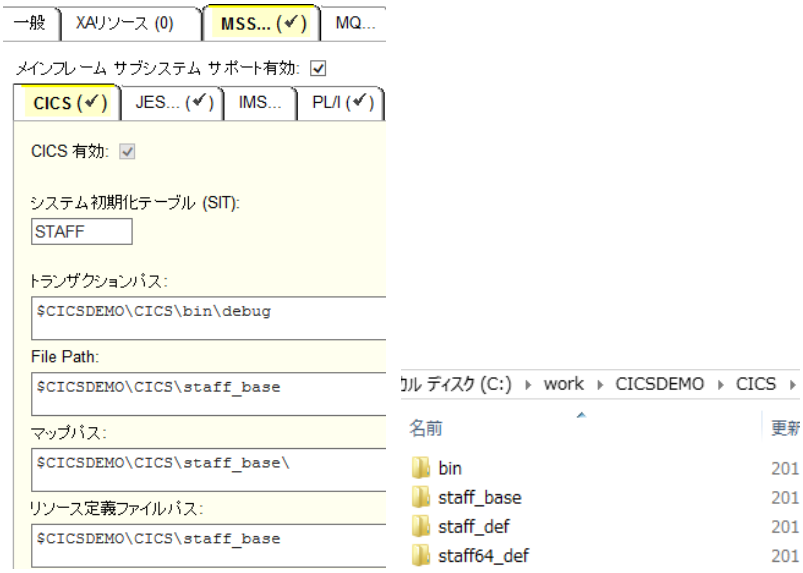
```
[ES-Environment]
CICSDEMO=C:¥work¥CICSDEMO
ES_SSTM_CICS="$CICSDEMO¥CICS¥sstmcics.jcl"
```

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。
CODEWATCH_NOTIF 環境変数：
CodeWatch デバッガを使用する開発用インスタンスに指定しますが、ここでは Eclipse のデバッガを使用しますので削除します。本番インスタンスにはパーフォーマンスの観点から排除することを推奨します。

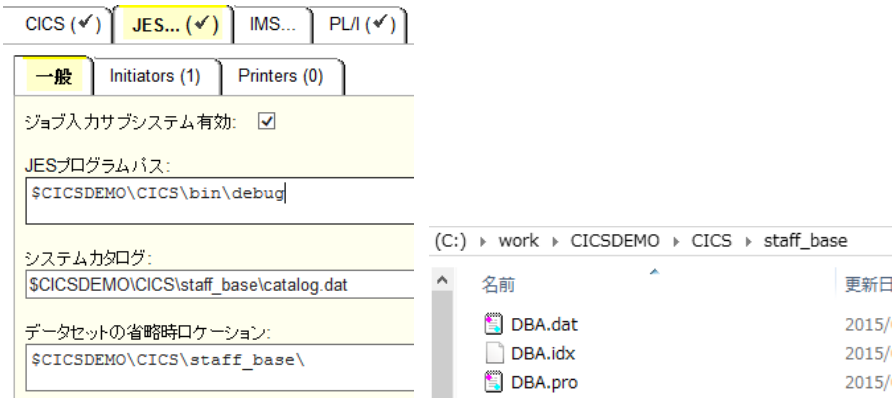
7) [サーバー] > [プロパティ] > [MSS] > [CICS] タブで表示される画面の各項目を確認します。

項目名	説明
メインフレーム サブシステム サポート有効	[MSS] タブ配下の有効設定をオン、オフ指定します。
システム初期化テーブル（SIT）	SIT 名称を指定します。
トランザクションパス	PCT やプログラムから呼ばれる実行ファイルが存在するパスを指定します。
File Path	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	マップ実行ファイルが存在するパスを指定します。
リソース定義ファイルパス	リソース定義ファイルが存在するパスを指定します。



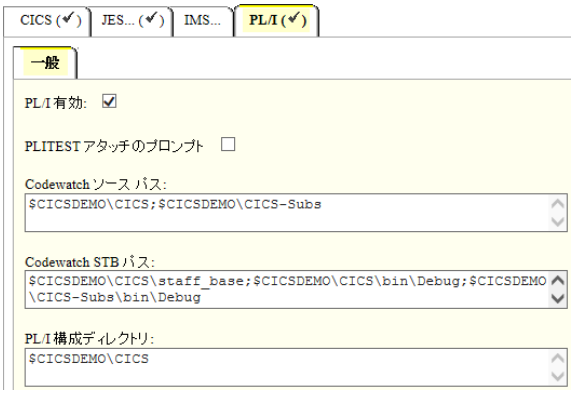
8) [サーバー] > [プロパティ] > [MSS] > [JES] > [一般] タブで表示される画面の各項目を確認します。

項目名	説明
ジョブ入力サブシステム 有効	[JES] タブ配下の有効設定をオン、オフ指定します。
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。

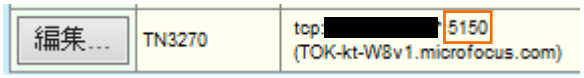


9) [サーバー] > [プロパティ] > [MSS] > [PL/I] > [一般] タブで表示される画面の各項目を確認します。

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。



10) [リスナー] タブで表示される 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。

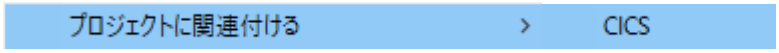


11) 画面左上の [Home] をクリックして Enterprise Server インスタンス一覧画面に戻ります。

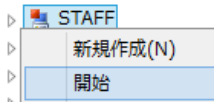
3.7 Enterprise Server インスタンスの開始と確認

1) サーバーエクスプローラ内に 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 [ローカル [localhost:86] 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。

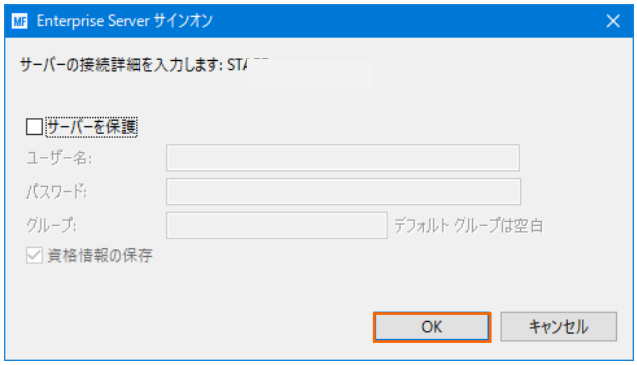
2) サーバーエクスプローラ内の STAFF インスタンスを右クリックし、[プロジェクトに関連付ける] > [CICS] を選択します。これにより Eclipse 内の CICS プロジェクトから実行されるプログラムは STAFF インスタンスで処理されることになります。



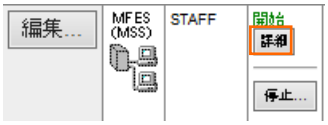
3) STAFF インスタンスを右クリックして [開始] を選択します。



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。

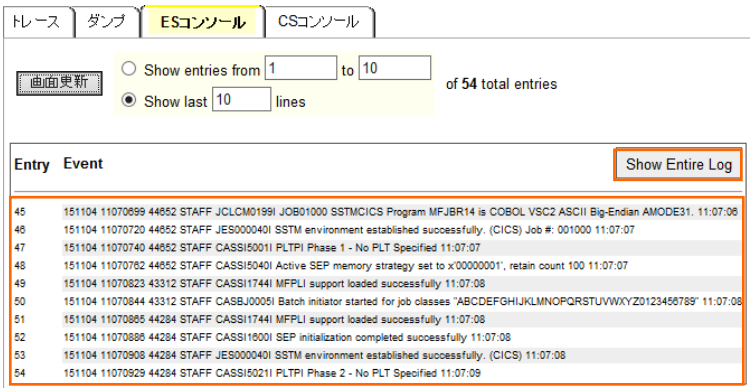


5) Enterprise Server Administration 画面へ移動して開始状態であることを確認後、[詳細] ボタンをクリックします。



6) [サーバー] > [診断] > [ES コンソール] で STAFF インスタンスのコンソールログをリアルタイムにチェックすることができます。また [Show Entire Log] をクリックしてログ全体を表示させることも可能です。

正常に開始されたことを確認します。



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

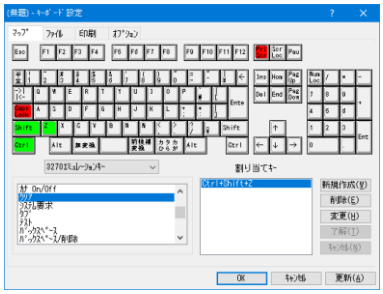
7) 画面左上の [Home] をクリックして一覧画面に戻ります。

3.8 CICS の実行

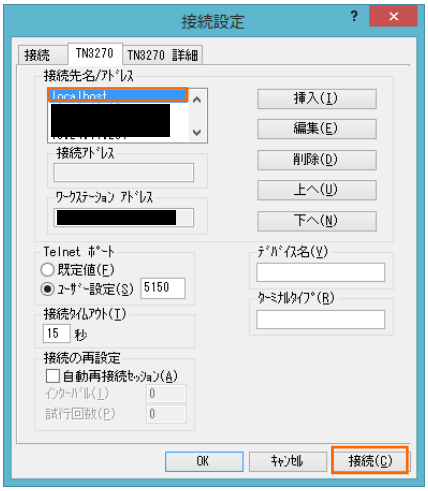
1) TN3270 エミュレーターへ Enterprise Server インスタンスのポートへの接続を設定して、[接続] ボタンをクリックします。

補足）TN3270 エミュレーターで、使用しているキーボード設定をご確認ください。

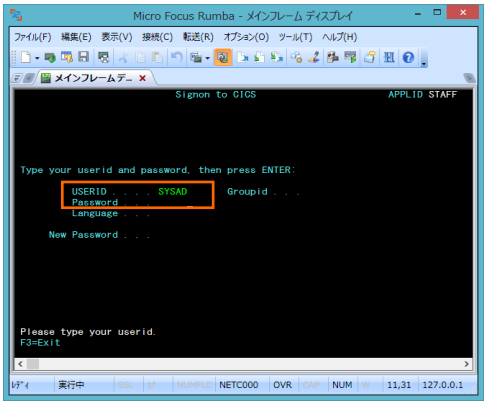
Rumba の例）



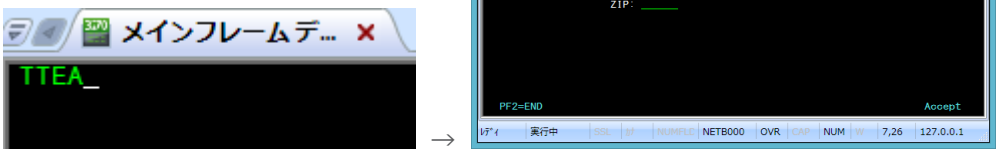
ここでは Micro Focus Rumba を使用します。



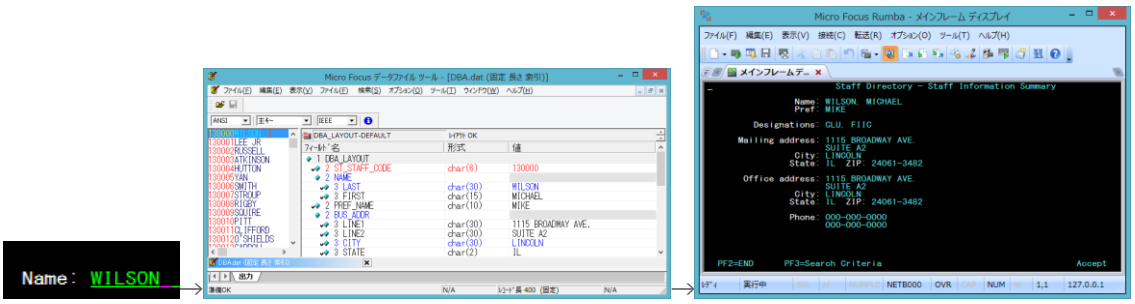
2) CICS ログイン画面が表示されますので USERID と Password へ SYSAD を入力して実行キーを押します。



3) ログイン後クリアキーで画面クリアを行い、PCT 名である TTEA を入力して実行キーを押すと、プログラムが起動されてアプリケーション画面が表示されます。



4) [Name] 欄へ WILSON と入力して実行キーを押すと、ファイルからデータが読み込まれて表示されます。

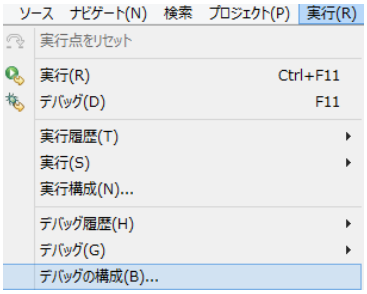


5) 実行ファイルの実行とファイルデータ読み込みまでを確認できましたので、TN 3270 エミュレーターの接続を切断します。

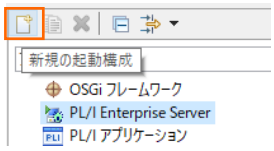
3.9 PL/I ソースのデバッグ

CICS から実行される PL/I プログラムをデバッグします。

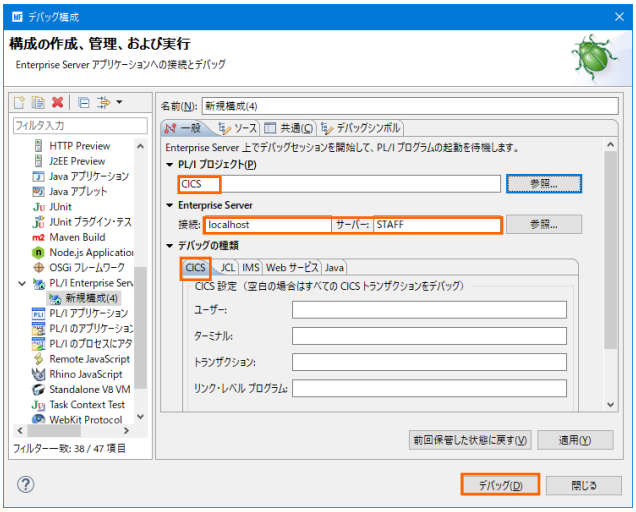
1) [実行] プルダウンメニューの [デバッグの構成] を選択します。



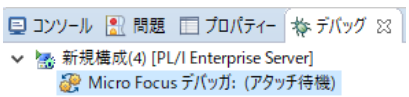
2) 左側のツリービューから [PL/I Enterprise Server] を選択して、左上の [新規の起動構成] アイコンをクリックします。



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 タブを選択した状態で、[デバッグ] ボタンをクリックします。



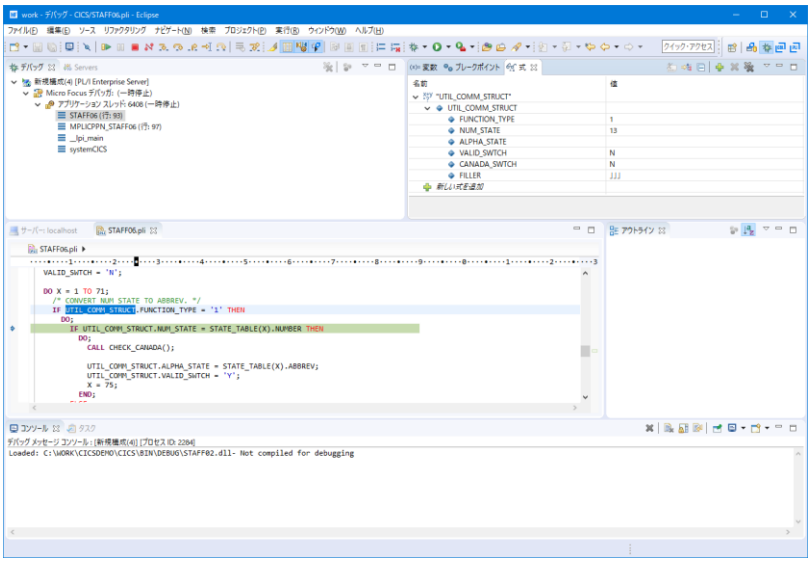
4) デバッグタブで [アタッチ待機] 状態になったことを確認します。



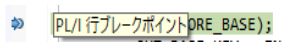
5) 前項と同様に TN3270 エミュレーターから接続後、ログインしてトランザクションを実行します。

6) 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、 [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。

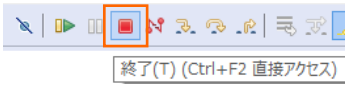


7) 希望のステップの左端をダブルクリックすることにより、ブレークポイントを設定することも可能です。

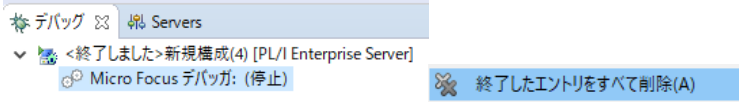


8) 先に進める場合は画面上部の再開アイコンをクリックします。

9) 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。



10) 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。



11) TN 3270 エミュレーターの接続を切断します。

3.10 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。

WHAT'S NEXT

- メインフレーム PL/I 開発：CICS　Eclipse 編
- 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。