
Micro Focus Enterprise Developer チュートリアル

メインフレーム PL/I 開発 : CICS

Visual Studio 2022 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、Visual Studio 2022 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

- 本チュートリアルで使用したマシン OS : Windows 11 Pro
- 使用マシンに Microsoft Visual Studio 2022 がインストールされていること
- Windows 開発環境に Enterprise Developer 8.0 for Visual Studio 2022 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. Visual Studio の起動
3. PL/I ソリューションのインポート
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 Server インスタンスの設定
8. Enterprise Server インスタンス開始と確認
9. CICS の実行
10. PL/I ソースの動的デバッグ
11. 終了処理

3.1 チュートリアルの準備

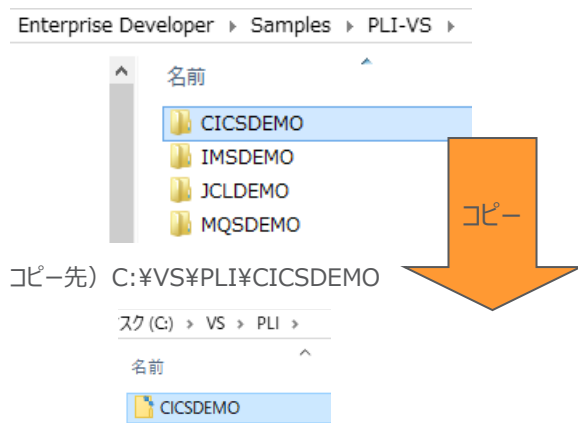
例題プログラムに関連するリソースを用意します。

- 1) Visual Studio のソリューションを保存するフォルダを C:¥ 直下に作成します。

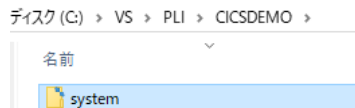
例) C:¥VS¥PLI

- 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例題プログラム CICSDEMO フォルダを作成した C:¥VS¥PLI へコピーします。

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



また、実行時に使用する system フォルダを C:¥VS¥PLI¥CICSDEMO 直下に新規作成してください。



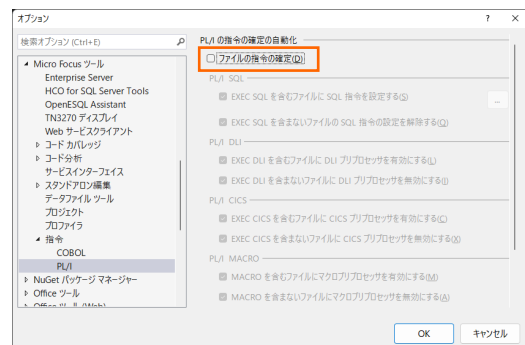
3.2 Visual Studio の起動

- 1) Visual Studio 2022 を起動します。



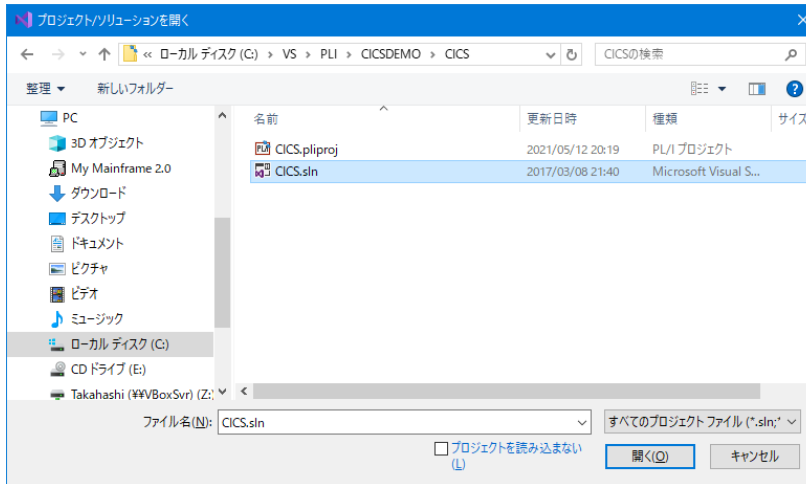
- 2) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ツール] プルダウンメニューの [オプション] を選択してオプションウィンドウを表示します。

左側ツリービューの [Micro Focus ツール] > [指令] > [PL/I] > [ファイルの指令の確定] 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。

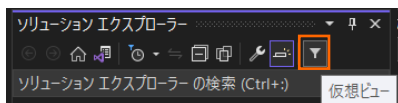


3.3 PL/I ソリューションのインポート

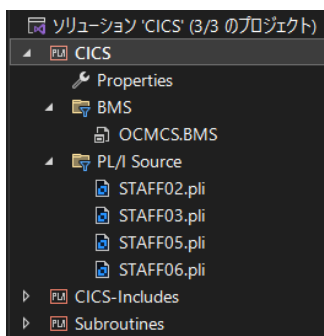
- 1) 用意した例題ソリューションを表示します。[ファイル] プルダウンメニューから [開く] > [プロジェクト/ソリューション] を選択し、[プロジェクトを開く] ウィンドウにて前項でコピーした C:¥VS¥PLI¥CICSDemo¥CICS に存在する CICS.sln を選択後 [開く] ボタンをクリックします。



- 2) 種類別に表示するため、[ソリューション エクスプローラー] 内の [仮想ビュー] アイコンをクリックします。



- 3) [ソリューション エクスプローラー] にインポートしたソリューションと 3 つのプロジェクトが表示されます。

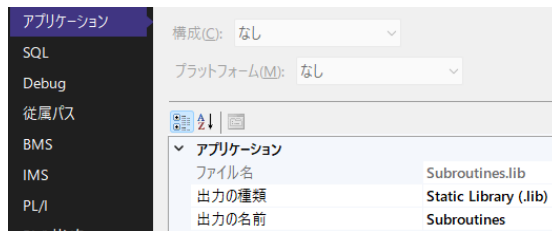


3.4 プロジェクトプロパティの確認

この例題は Subroutines プロジェクトで作成される 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。まずはプロジェクトの設定値を確認します。

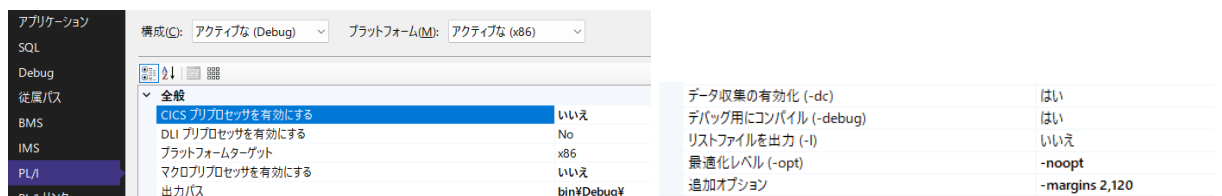
- 1) サブとなるプロジェクトの設定を確認します。[ソリューション エクスプローラー] 内 Subroutines プロジェクトの [Properties] をダブルクリックしてプロパティウィンドウを表示します。

- 2) 左側ツリービュー [アプリケーション] を選択すると、LIB 生成を指定していることが確認できます。



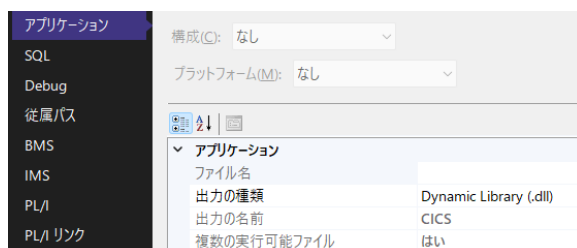
- 3) 左側ツリービューの [PL/I] を選択すると下記項目が確認できます。

項目名	説明
プラットフォーム ターゲット	稼働ビット数を指定します。x86 が選択されており 32-bit 稼働が指定されています。
出力パス	生成されたファイルが出力されるパスを指します。任意に指定可能です。
デバッグ用にコンパイル	デバッグ実行時に使用するファイルを生成するように指定します。

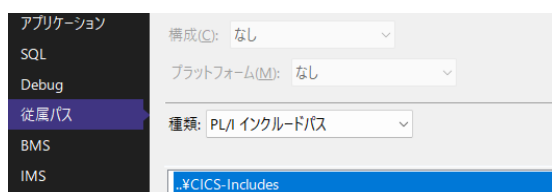


- 4) 次にメインとなるプロジェクトの設定を確認します。[ソリューション エクスプローラー] 内 CICS プロジェクトの [Properties] をダブルクリックしてプロパティウィンドウを表示します。

- 5) 左側ツリービューの [アプリケーション] を選択すると、DLL 生成を指定していることが確認できます。

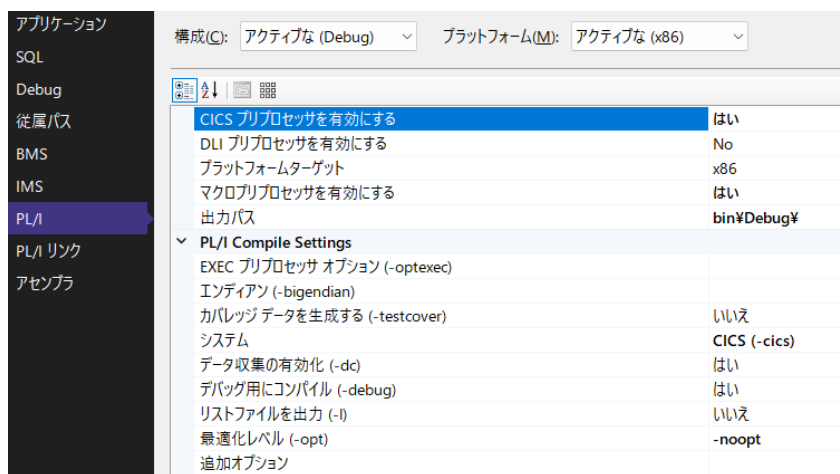


- 6) 左側ツリービューの [従属パス] を選択すると、同じソリューションに含まれている CICS-Includes プロジェクトのパスを PL/I インクルードパスへ指定していることがわかります。このように指定することで共有インクルードファイルへのアクセスが可能になります。



7) 左側ツリービューの [PL/I] を選択すると下記項目が確認できます。

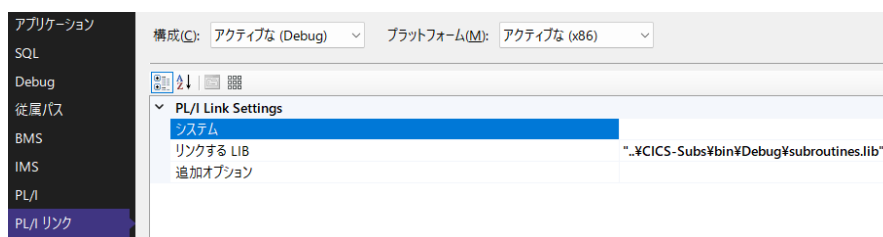
項目名	説明
CICS プリプロセッサを有効にする	プログラムでは EXEC CICS 構文を使用しているため “はい” を選択します。
プラットフォーム ターゲット	稼働ビット数を指定します。x86 が選択されており 32-bit 稼働が指定されています。
出力パス	生成されたファイルが出力されるパスを指します。任意に指定可能です。
システム	CICS として機能させるため CICS を選択します。他に -ims と -mvs が選択可能です。
デバッグ用にコンパイル	デバッグ実行時に使用するファイルを生成するように指定します。



Visual Studio の設定画面で、左側のツリービューに「PL/I」が選択されています。右側の設定パネルには「PL/I Compile Settings」が展開されており、以下の項目が設定されています。

項目名	設定値
CICS プリプロセッサを有効にする	はい
DLI プリプロセッサを有効にする	No
プラットフォームターゲット	x86
マクロプリプロセッサを有効にする	はい
出力パス	bin\Debug*
EXEC プリプロセッサ オプション (-optexec)	
エンディアン (-bigendian)	
カバレッジデータを生成する (-testcover)	いいえ
システム	CICS (-cics)
データ収集の有効化 (-dc)	はい
デバッグ用にコンパイル (-debug)	はい
リストファイルを出力 (-l)	いいえ
最適化レベル (-opt)	-noopt
追加オプション	

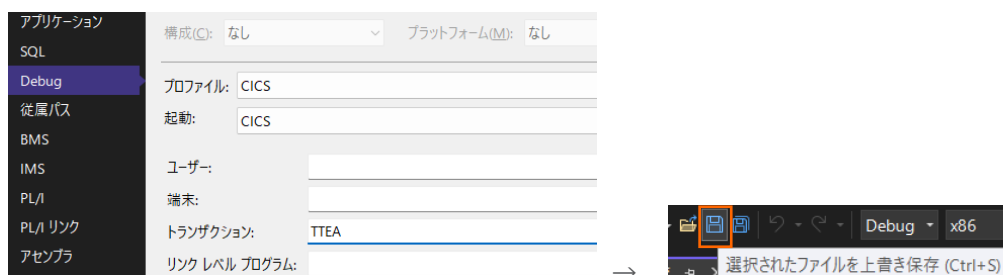
8) 左側ツリービュー [PL/I リンク] を選択すると、サブプロジェクトで生成した LIB ファイルをリンクさせていることが確認できます。



Visual Studio の設定画面で、左側のツリービューに「PL/I リンク」が選択されています。右側の設定パネルには「PL/I Link Settings」が展開されており、以下の項目が設定されています。

項目名	設定値
システム	
リンクする LIB	..\CICS-Subs\bin\Debug\subroutines.lib
追加オプション	

9) 左側メニュー [Debug] を選択し、[プロファイル] には CICS を、[起動] には CICS を、[トランザクション] には TTEA を指定し、設定を保存します。



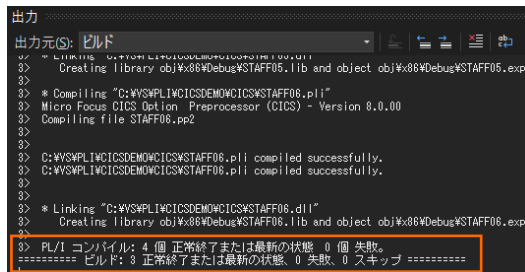
Visual Studio の設定画面で、左側のツリービューに「Debug」が選択されています。右側の設定パネルには以下の項目が設定されています。

項目名	設定値
構成 (C):	なし
プラットフォーム (M):	なし
プロファイル:	CICS
起動:	CICS
ユーザー:	
端末:	
トランザクション:	TTEA
リンクレベル プログラム:	

右下のツールバーには「保存」アイコンが赤い枠で囲われており、その下のステータスバーには「選択されたファイルを上書き保存 (Ctrl+S)」と表示されています。

3.5 ビルドの実行

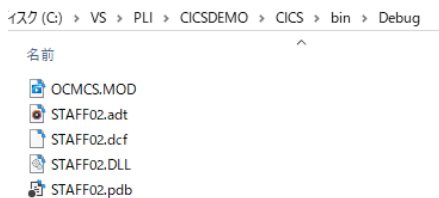
- 1) [ソリューション エクスプローラー] の CICS ソリューションを右クリックして [ソリューションのビルド] を選択すると、コンパイル指定に沿ったビルドが実行されます。
- 2) [出力] ウィンドウで成功を確認します。



```

出力
出力元(S): ビルド
> C:\MSW\PLI\CICSDemo\CICS\STAFF05
> Creating library obj\x86\Debug\STAFF05.lib and object obj\x86\Debug\STAFF05.exp
>
> * Compile "C:\MSW\PLI\CICSDemo\CICS\STAFF06.pli"
> Micro Focus CICS Option Preprocessor (CICS) - Version 8.0.00
> Compiling file STAFF06.pp2
>
>
> C:\MSW\PLI\CICSDemo\CICS\STAFF06.pli compiled successfully.
> C:\MSW\PLI\CICSDemo\CICS\STAFF06.pli compiled successfully.
>
>
> * Linking "C:\MSW\PLI\CICSDemo\CICS\STAFF06.dll"
> Creating library obj\x86\Debug\STAFF06.lib and object obj\x86\Debug\STAFF06.exp
>
> PL/I コンパイル: 4 個 正常終了または最新の状態 0 個 失敗。
===== ビルド: 8 正常終了または最新の状態 0 失敗 0 スキップ =====
  
```

- 3) 前項で確認した出力パスへ実行ファイルに指定した DLL ファイル等が作成されていることを確認します。



```

イスク(C:) > VS > PLI > CICSDemo > CICS > bin > Debug
名前
OCMCS.MOD
STAFF02.adt
STAFF02.dcf
STAFF02.DLL
STAFF02.pdb
  
```

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。

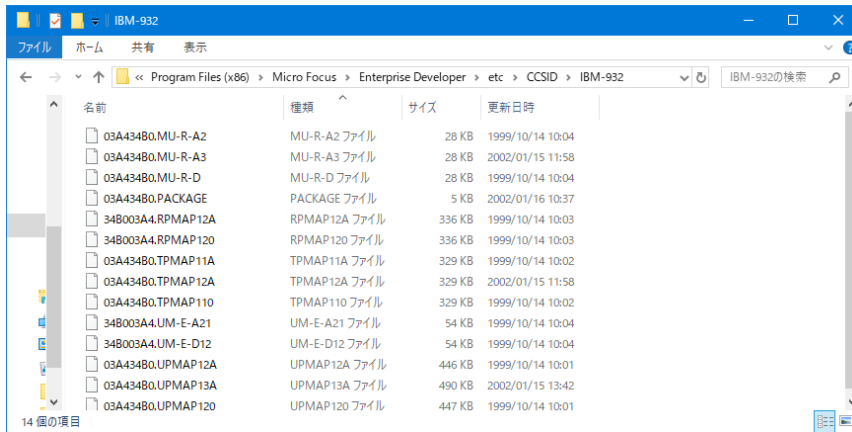
製品マニュアルにリンクされている下記の IBM CCSID 変換テーブルを、Web ブラウザから任意のフォルダへダウンロードします。アドレスは変更される可能性がありますので、製品マニュアルにてご確認ください。

<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。

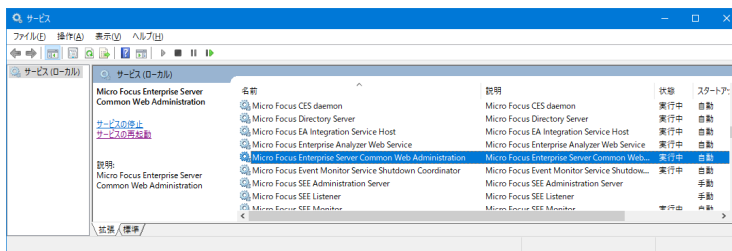
例) C:\Program Files (x86)\Micro Focus\Enterprise Developer\etc\CCSID

- 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。



3.7 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 [Micro Focus Server] を右クリックして [管理] を選択します。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。ESCWA では Micro Focus Directory Server サービスのポートへ接続して登録された Enterprise Server インスタンスを管理するため、まず接続を作成します。既に存在している場合には、インスタンス作成に進んでください。

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



追加ウィンドウでは [名前] は任意ですが、ここでは 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。

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

- 3) 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:\¥VS¥PLI¥CICSDEMO¥CICS_SERVERS.xml がインポート対象のファイルです。
作成した [Default] 接続をクリックし、右側ペインの [インポート] ボタンをクリックします。



- 4) [インポートタイプの選択] では [XML] を選択して [次へ] 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

インポートタイプの選択

JSON
クライアント ブラウザが実行されているホスト上のファイルを選択して、json ファイルをインポートします。

XML
クライアント ブラウザが実行されているホスト上のファイルを選択して、xml ファイルをインポートします。

レガシー
Directory Serverが実行されているホスト上のディレクトリの場所を選択して、レガシー リポジトリ (.dat ファイルのディレクトリ) をインポートします。

- 5) [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。

インポート

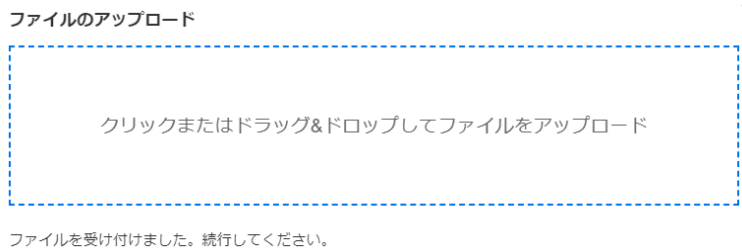
手順 1: インポートタイプ ✓

手順 2: アップロード

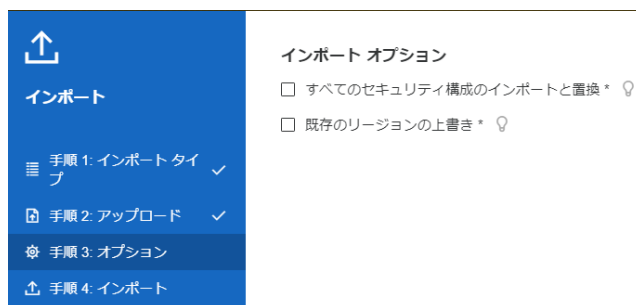
ファイルのアップロード

クリックまたはドラッグ&ドロップしてファイルをアップロード

- 6) [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。



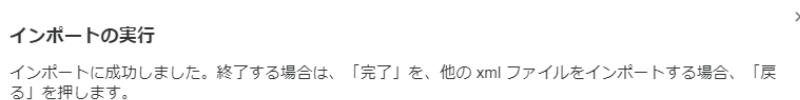
- 7) 手順 3 では [次へ] をクリックします。



- 8) 手順 4 では [インポート] をクリックします。



- 9) 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。



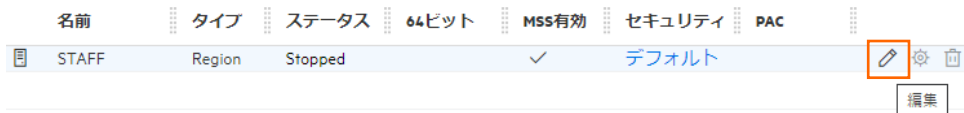
- 10) 32 ビットアプリケーション稼働用の STAFF と、64 ビットアプリケーション稼働用の STAFF64 インスタンスが追加されます。Visual Studio では 32 ビットアプリケーション稼働用にコンパイルしましたので、ビット数が一致する STAFF インスタンスを使用します。

	名前	タイプ	ステータス	64ビット	MSS有効	セキュリティ
📄	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。

11) 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。



12) インスタンスのログなどが出力される [システムディレクトリ] には前項で作成した system フォルダを指定し、[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、Visual Studio からの動的デバッグが可能になります。指定後は [適用] ボタンをクリックします。



13) [リージョンの機能] では、[MSS 有効]、[JES 有効]、[PL/I 有効] のチェックがオンになっていることを確認します。



14) [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:\VS\PLI\CICSDemo
ES_SSTM_CICS="$CICSDemo\sstmciics.jcl"
#MF_CHARSET=E
MFACCCGI_CHARSET=Shift_JIS
```

追加設定

構成情報

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

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。

15) 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。構成情報に指定した環境変数を使用して値が入力されています。

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。例題では STAFF 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。例題で用意されている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。 .gnt ファイルが生成されているパスを指定します。
ファイルパス	データセットのデフォルトパスを指定します。例題で用意されているファイルの置かれているパスを指定します。
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 |

システム初期化テーブル (SIT) <input type="text" value="STAFF"/>	リソース定義ファイルパス <input type="text" value="\$CICSDEMO\staff_base"/>
トランザクションパス <input type="text" value="\$CICSDEMO\CICS\bin\debug"/>	<div style="background-color: #f4a460; padding: 5px; border: 1px solid #ccc;"> <p>重要</p> <p>入力値は全て半角英数字で指定してください。 これらのフィールドでは改行を入れないように注意してください。</p> </div>
ファイルパス <input type="text" value="\$CICSDEMO\staff_base"/>	
マップパス <input "="" type="text" value="\$CICSDEMO\staff_base\"/>	
<input type="checkbox"/> EZASOCKET サポート <input type="text" value=""/>	

16) 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。構成情報に指定した環境変数を使用して値が入力されています。

項目名	説明
JES プログラム パス	COBOL アプリケーションの実行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ロケーション	ジョブ実行時に生成されるスプールデータやカタログされるデータセットのデフォルトパスを指定します。

JESの構成 | 適用

JES プログラムパス	システムカタログ
\$CICSDEMO\CICS\bin\debug	SCICSEMO\staff_base\catalog.dat
データセットの省略時ロケーション	システムプロシージャライブラリ
\$CICSDEMO\staff_base\	
Fileshare 構成ロケーション	

17) [イニシエータ] には A ~ 9 までのクラスに対するイニシエータが設定されていることを確認します。

イニシエータ | *新規作成

名前	クラス	説明
INIT1	abcdefghijkl...	

JESイニシエータ

名前: INIT1

クラス: abcdefghijklmnopqrstuvwxyz0123456789

説明:

*入力必須の項目です

18) セキュリティ観点から、Web リスナーのデフォルトステータスは [Disabled] になっています。安全を確認したうえで、[一般] プルダウンメニューから [リスナー] を選択し、表示された Web リスナーのステータスを [Stopped] へ変更後、[適用] ボタンをクリックします。例題インスタンスが既に [Stopped] の場合はこれを確認します。

TLS設定

ステータス	ステータスの設定	実際のアドレス
Disabled	Stopped	tcp:0.0.0.0:0

19) [TN3270] リスナーを選択すると右側ペインにポート番号が表示されます。TN3270 エミュレータからの接続では、このポートを指定します。

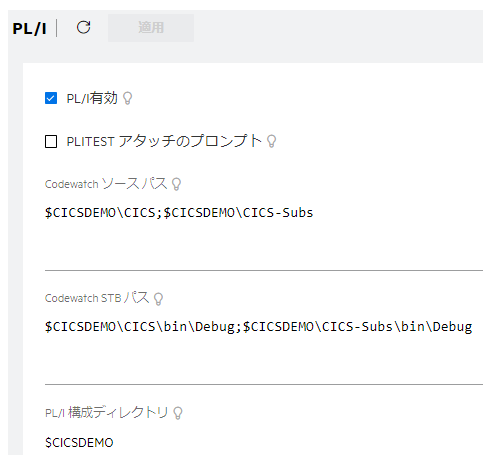
通信プロセス | *通信サーバ

- 通信プロセス 1
 - Web Services and J2EE
 - Web
 - TN3270**

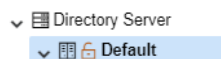
プロトコル	ホスト名またはIP アドレス	ポート
tcp	.	5150

20) 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	オン、オフ指定します。ここではオンを指定します。
Codewatch ソース パス	デバッグで使用するソースファイルのパスを指定します。
Codewatch STB パス	デバッグで使用するデバッグファイルのパスを指定します。例) XXX.stb
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。



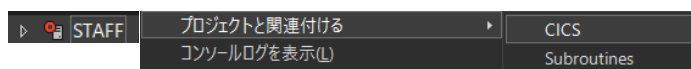
21) 画面左側ペインの [Default] をクリックして一覧画面に戻ります。



3.8 Enterprise Server インスタンスの開始と確認

1) Visual Studio の [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 [Micro Focus Server] を右クリックし、[最新の情報に更新] を選択してリフレッシュしてください。

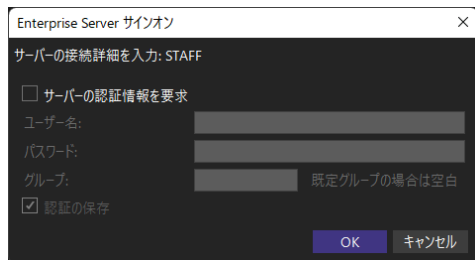
2) [サーバー エクスプローラー] 内の STAFF インスタンスを右クリックし、[プロジェクトと関連付ける] > [CICS] を選択します。これにより CICS プロジェクトから実行される処理は STAFF インスタンスで処理されることとなります。



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。



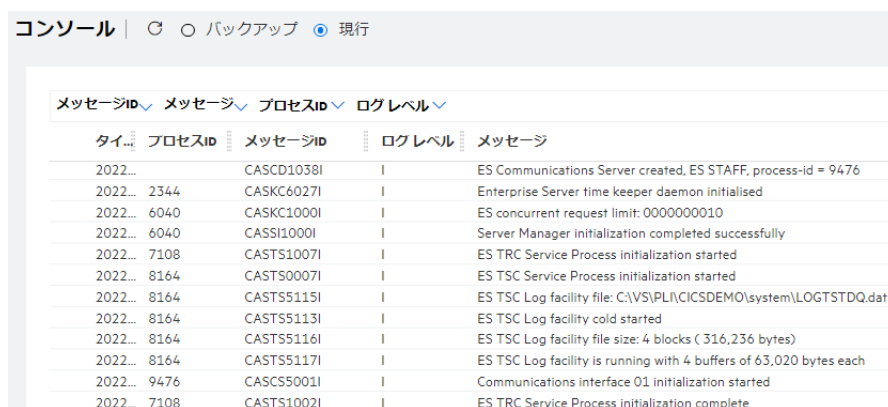
- 4) 下記ウィンドウが表示された場合は、ここではユーザーによる制限を行わないため [OK] ボタンをクリックします。



- 5) 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。



- 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、正常に開始されたことを確認します。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。



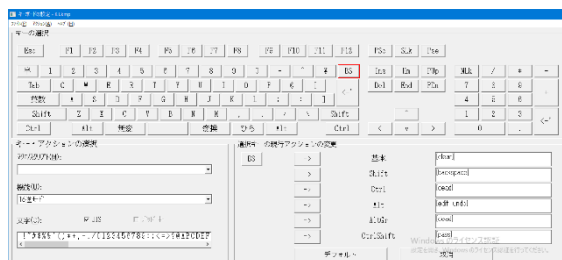
注意
いくつかのサービス開始が失敗してもインスタンスは開始されますので、ログ内容を必ず確認してください。

3.9 CICS の実行

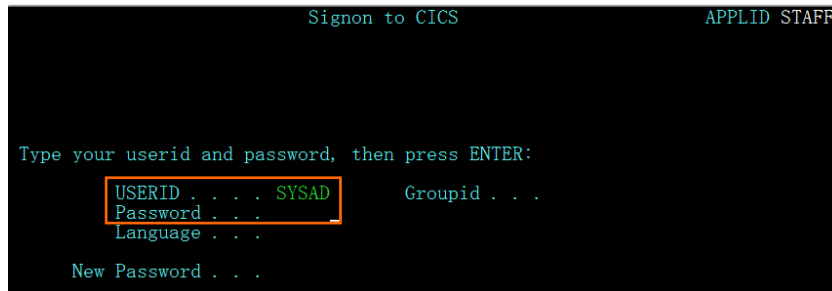
ご使用の TN3270 エミュレータを前項で確認した TN3270 リスナーポート (localhost:5150) へ接続します。

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)



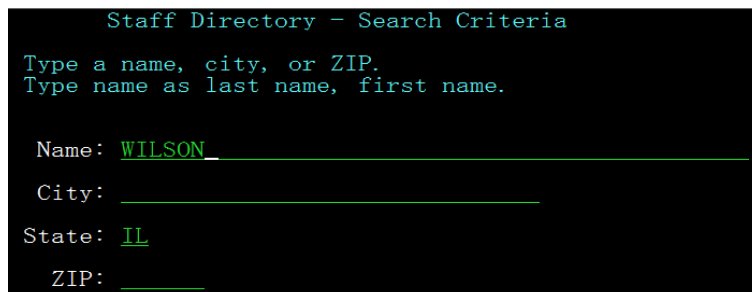
- 1) 接続すると、例題の SIT では初期トランザクションに CESN が指定されているため、接続後 CICS サインイン画面が表示されます。[USERID] と [PASSWORD] へ SYSAD を入力して実行キーを押します。



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:\¥VS¥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

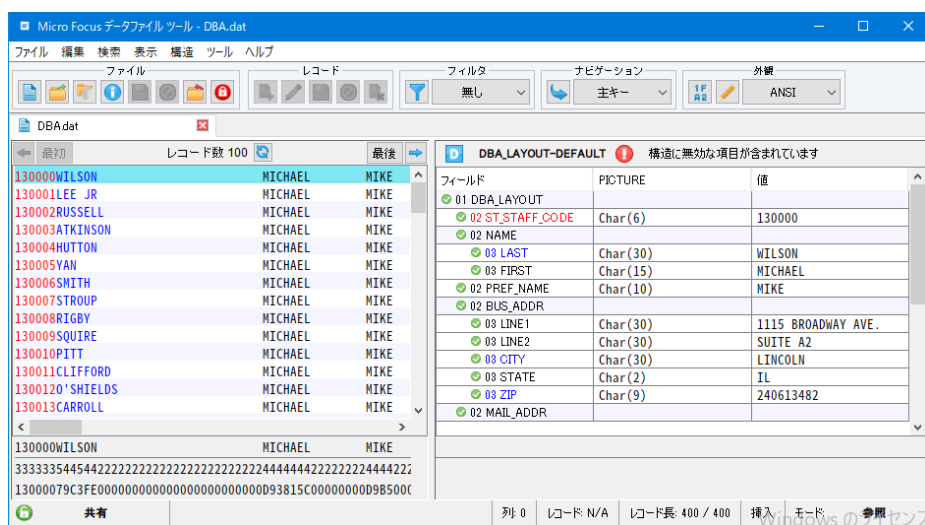
Designations: CLU, FIIC

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
  
```

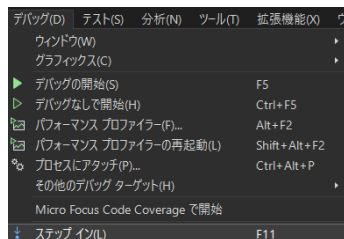
DBA.dat の内容)



3.10 PL/I ソースの動的デバッグ

Visual Studio を使用して、例題プログラムのデバッグを行います。前項で実施しましたが、CICSDEMO インスタンスの [動的デバッグを許可] へのチェックと CICSDEMO インスタンスと Visual Studio プロジェクトの関連付けが必要です。

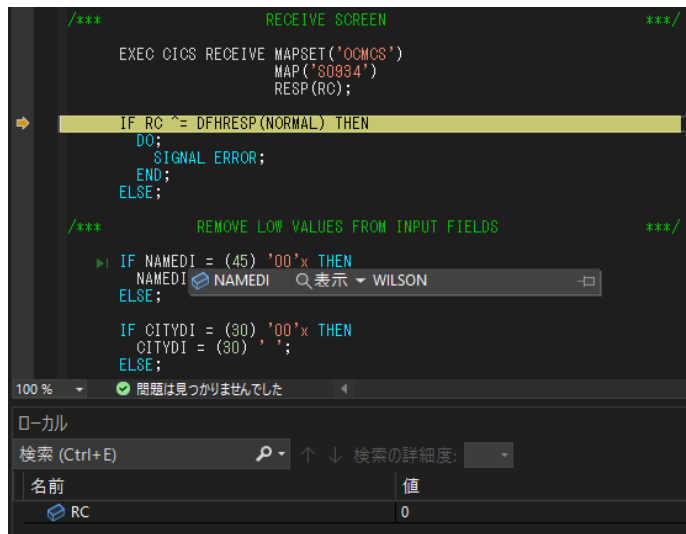
- 1) [デバッグ] プルダウンメニューの [ステップ イン] を選択します。



- 2) 画面左下が [準備完了] となり、アタッチ待機状態になったことを確認します。



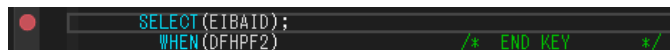
- 3) 前項と同様に TN3270 エミュレータから ACCT トランザクションを実行すると、プログラムのステップ実行が可能になります。F11 キーもしくは [デバッグ] ブルダウンメニューから [ステップ イン] を選択してステップを進めることができます。



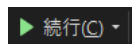
マウスオーバーやウォッチタブを利用して変数の値が確認できます。

なお、この例題プログラムでは TN3270 エミュレータとの画面送受信がありますので、その都度、表示を切り替えてデバッグします。

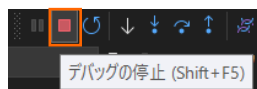
- 4) 希望のステップの左端をクリックすることにより、ブレークポイントを設定することも可能です。



- 5) 先に進める場合は画面上部の [続行] アイコンをクリックします。



- 6) デバッグを終了させるため、画面上部の [デバッグ停止] アイコンをクリックします。



- 7) TN3270 エミュレータを切断します。

- 8) 画面左下が元の色の [準備完了] となり、アタッチ待機状態ではなくなったことを確認します。



3.11 終了処理

- 1) Visual Studio の [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、Visual Studio 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。

WHAT'S NEXT

- メインフレーム PL/I 開発 : JCL Visual Studio 2022 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。