
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 2017 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 2017 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 10 Enterprise
- 使用マシンに Microsoft Visual Studio 2017 がインストールされていること
- Windows 開発環境に Enterprise Developer 4.0 for Visual Studio 2017 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

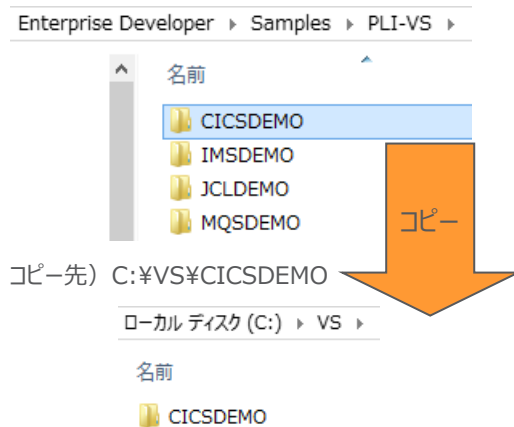
1. チュートリアルの準備
2. Visual Studio の起動
3. PL/I ソリューションのインポート
4. プロジェクトプロパティの確認
5. ビルドの実行
6. Enterprise Server インスタンスの設定
7. Enterprise Server インスタンス開始と確認
8. CICS の実行
9. PL/I ソースのデバッグ
10. 終了処理

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

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



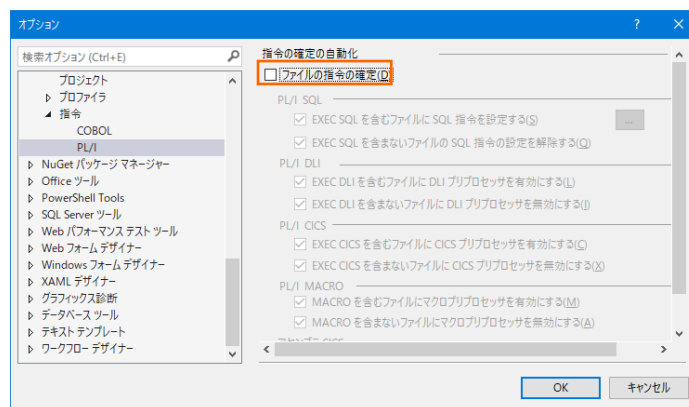
3.2 Visual Studio の起動

- 1) Visual Studio 2017 を起動します。



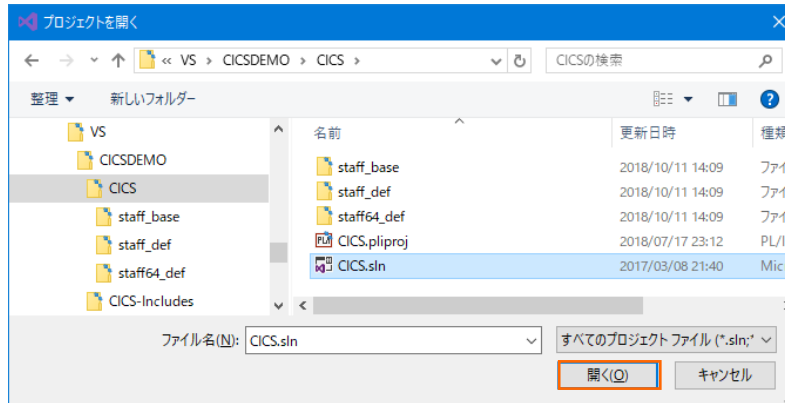
- 2) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ツール] プロダクションメニューの [オプション] を選択してオプションウィンドウを表示します。

左側ツリービューの [Micro Focus] > [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。

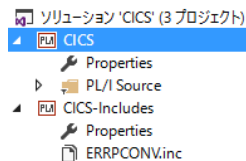


3.3 PL/I ソリューションのインポート

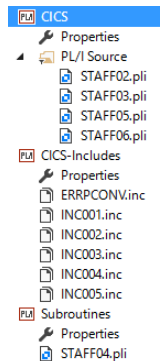
- 1) 用意した例題ソリューションを表示します。[ファイル] プルダウンメニューから [開く] > [プロジェクト/ソリューション] を選択し、[プロジェクトを開く] ウィンドウにて前項でコピーした C:\VS\CICSDemo\CICS に存在する CICS.sln を選択後 [開く] ボタンをクリックします。



- 2) [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) CICS , Subroutines プロジェクトを展開するとインクルードファイルや PL/I ソースが確認できます。

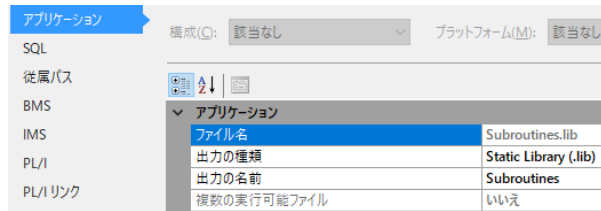


3.4 プロジェクトプロパティの確認

この例題は 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。プロジェクトの設定値を確認していきます。

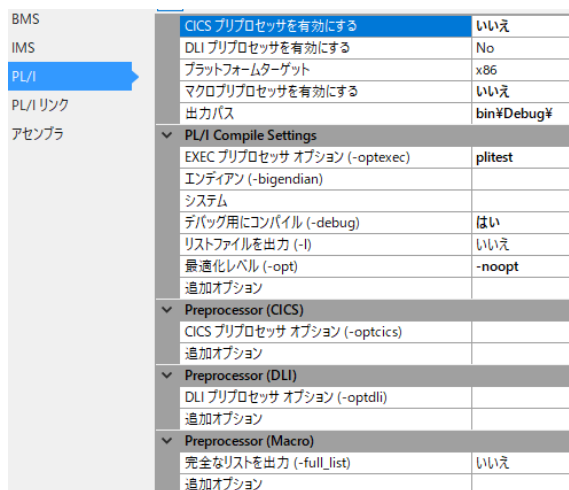
- 1) まずはサブとなるプロジェクトの設定を確認します。[ソリューション エクスプローラー] 内 Subroutines プロジェクトの [Properties] をダブルクリックしてプロパティウィンドウを表示します。

- 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。



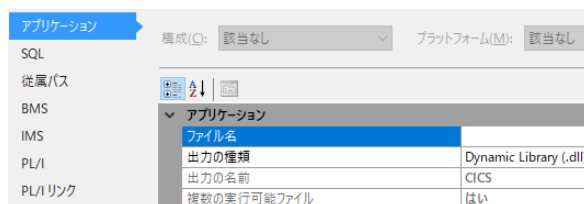
- 3) 左側ツリービューの [PL/I] を選択すると下記項目が確認できます。

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。
PL/I Compile Settings EXEC プロセッサ オプション	デバッグツールを起動するには plitest を指定します。
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。

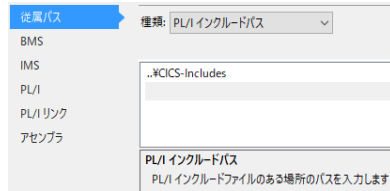


- 4) 次にメインとなるプロジェクトの設定を確認します。[ソリューション エクスプローラー] 内 CICS プロジェクトの [Properties] をダブルクリックしてプロパティウィンドウを表示します。

- 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。

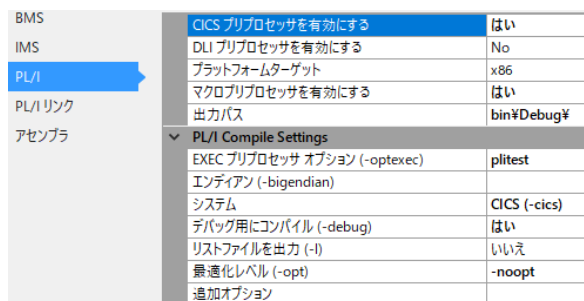


- 6) 左側ツリービューの [従属パス] を選択すると、同じソリューションに含まれている CICS-Includes プロジェクトのパスを PL/I インクルードパスへ指定していることがわかります。このように指定することで共有インクルードファイルへのアクセスが可能になります。



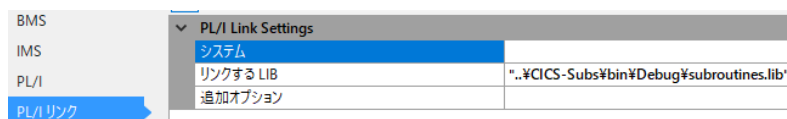
- 7) 左側ツリービューの [PL/I] を選択すると下記項目が確認できます。

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。
PL/I Compile Setting EXEC プロプロセッサ オプション	デバッグツールを起動するには plitest を指定します。使用しない場合は文字を削除します。
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。



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*
▼ PL/I Compile Settings	
EXEC プリプロセッサ オプション (-optexec)	plitest
エンディアン (-bigendian)	
システム	CICS (-cics)
デバッグ用にコンパイル (-debug)	はい
リストファイル出力 (-l)	いいえ
最適化レベル (-opt)	-noopt
追加オプション	

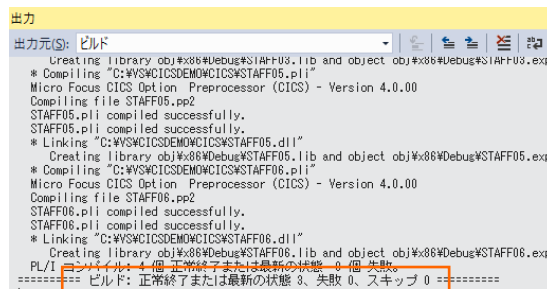
- 8) 左側ツリービュー [PL/I リンク] を選択すると、サブプロジェクトで生成した LIB ファイルをリンクさせていることが確認できます。



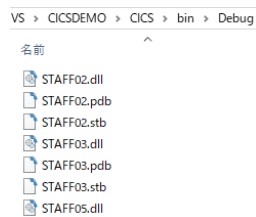
▼ PL/I Link Settings	
システム	リンクする LIB
	"..\CICS-Subs\bin\Debug\subroutines.lib"
追加オプション	

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。

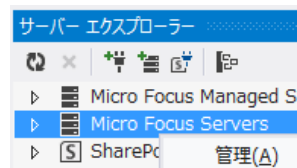


- 3) 前項で確認した出力パスへ実行ファイルに指定した DLL ファイルが作成されていることを確認します。



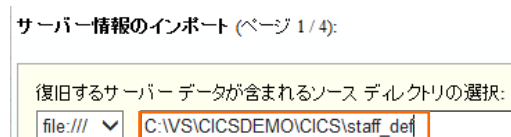
3.6 Enterprise Server インスタンスの設定

- 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 [Micro Focus Server] を右クリックして [管理] を選択します。

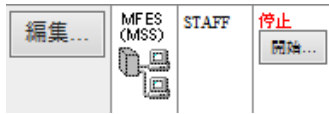


- 2) C:\VS\CICSDEMO\CICS には Enterprise Server インスタンスのサンプルが含まれており、これをインポートします。PL/I アプリケーションは 32 ビット稼働を指定したため、C:\VS\CICSDEMO\CICS\staff_def がインポート対象となります。64 ビットで稼働させる場合は C:\VS\CICSDEMO\CICS\staff64_def をインポートしてください。

Enterprise Server Administration 画面左側の [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。

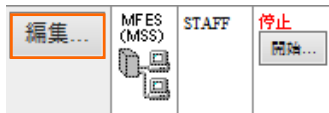


- 3) 画面の Page 2/4、3/4、ではそのまま [次へ] ボタンを、Page 4/4 では [OK] ボタンをクリックすると、STAFF という名前の 32 ビットアプリケーション稼働用 Enterprise Server インスタンスが追加されます。



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 [構成情報] 欄を下記のように入力し、[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\CICS\sstmcics.jcl"
CODEWATCH_NOTIF=Y
```

変更後；

```
[ES-Environment]
CICSDEMO=C:\VS\CICSDEMO
ES_SSTM_CICS="$CICSDEMO\CICS\sstmcics.jcl"
CODEWATCH_NOTIF=Y
```



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：

デバッガを使用する開発用インスタンスに指定します。本番インスタンスにはパフォーマンスの観点から排除することを推奨します。

6) [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] タブ配下の設定をオン、オフ指定します。ここではオンを指定します。
システム初期化テーブル (SIT)	SIT 名称を指定します。ここでは STAFF 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。

一般 XAリソース (0) **MSS... (✓)** MQ...

メインフレーム サブシステム サポート有効: ☒

CICS (✓) JES... (✓) IMS... PL/I (✓)

CICS 有効: ☒

システム初期化テーブル (SIT):
STAFF

トランザクション パス:
\$CICSDemo\CICS\bin\debug

ファイル パス:
\$CICSDemo\CICS\staff_base

マップ パス:
\$CICSDemo\CICS\staff_base\

リソース定義ファイル パス:
\$CICSDemo\CICS\staff_base

名前
bin
obj
Properties
staff_base
staff_def
staff64_def

7) [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] タブ配下の設定をオン、オフ指定します。ここではオンを指定します。
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刻ーション	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。

CICS (✓) **JES... (✓)** IMS... PL/I (✓)

一般 イニシエータ (1) プリンター (0)

ジョブ入力サブシステム有効: ☒

JES プログラム パス:
\$CICSDemo\CICS\bin\debug

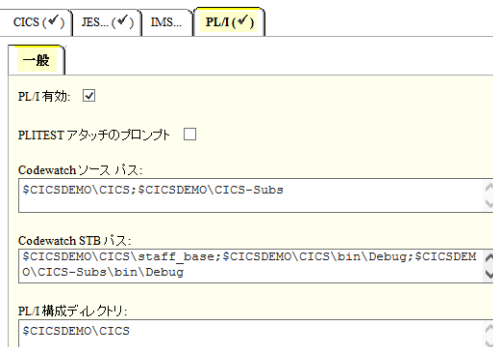
システム カタログ:
\$CICSDemo\CICS\staff_base\catalog.dat

データセットの省略時刻ーション:
\$CICSDemo\CICS\staff_base\

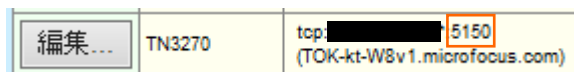
名前
DBA.dat
DBA.idx
DBA.pro

- 8) [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] タブ配下の設定をオン、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が存在するパスを指定します。セミコロンで複数パスも指定可能です。
Codewatch STB パス	デバッグ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。セミコロンで複数パスも指定可能です。
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。



- 9) [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。



- 10) 画面左上の [Home] をクリックして一覧画面に戻ります。

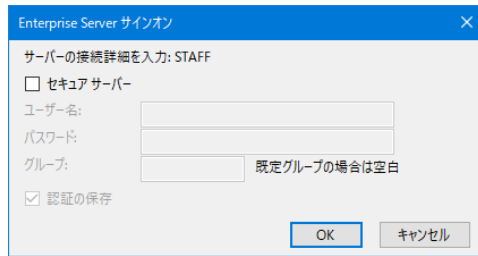


3.7 Enterprise Server インスタンスの開始と確認

- 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て [開始] を選択します。



- 3) 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。



Enterprise Server サインオン

サーバーの接続詳細を入力: STAFF

☐ セキュアサーバー

ユーザー名:

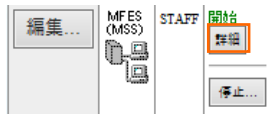
パスワード:

グループ: 既定グループの場合は空白

☒ 認証の保存

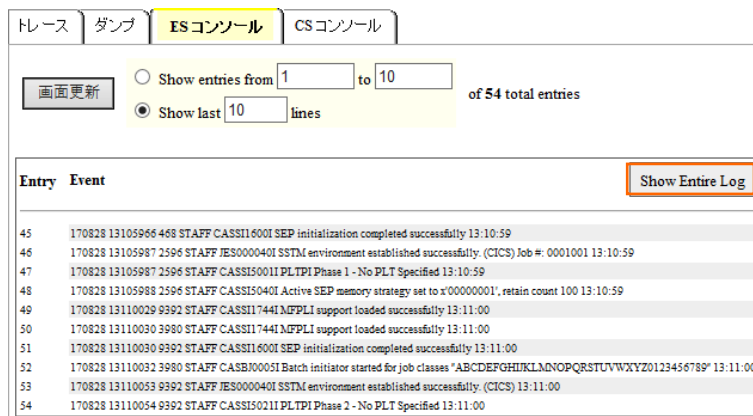
OK キャンセル

- 4) 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。



- 5) [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。



トレース ダンプ **ES コンソール** CS コンソール

画面更新

☐ Show entries from 1 to 10 of 54 total entries

☒ Show last 10 lines

Entry Event Show Entire Log

45	170828 13105966 468 STAFF CASSI1600I SEP initialization completed successfully 13:10:59
46	170828 13105987 2596 STAFF JES000040I SSTM environment established successfully. (CICS) Job #: 0001001 13:10:59
47	170828 13105987 2596 STAFF CASSI5001I PLTPI Phase 1 - No PLT Specified 13:10:59
48	170828 13105988 2596 STAFF CASSI5040I Active SEP memory strategy set to x'00000001', retain count 100 13:10:59
49	170828 13110029 9392 STAFF CASSI1744I MFPLI support loaded successfully 13:11:00
50	170828 13110030 3980 STAFF CASSI1744I MFPLI support loaded successfully 13:11:00
51	170828 13110030 9392 STAFF CASSI1600I SEP initialization completed successfully 13:11:00
52	170828 13110032 3980 STAFF CASSI0005I Batch initiator started for job classes "ABCDEF GHIJKLMNOPQRSTUVWXYZ0123456789" 13:11:00
53	170828 13110053 9392 STAFF JES000040I SSTM environment established successfully. (CICS) 13:11:00
54	170828 13110054 9392 STAFF CASSI5021I PLTPI Phase 2 - No PLT Specified 13:11:00



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。

- 6) 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 の実行と PL/I ソースのデバッグ

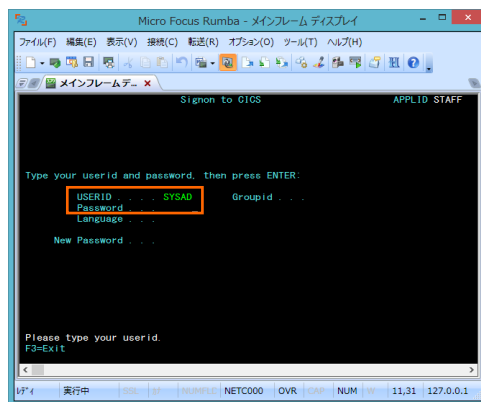
- 1) TN3270 エミュレーターを使用し、Enterprise Server インスタンスのポートへの接続を設定して、[接続] ボタンをクリックします。ここでは Micro Focus Rumba を使用します。

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。

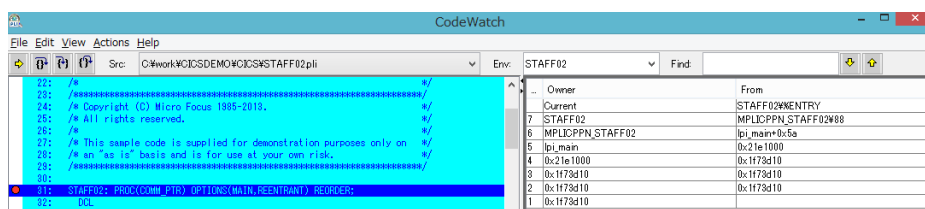
Rumba の例)



- 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。



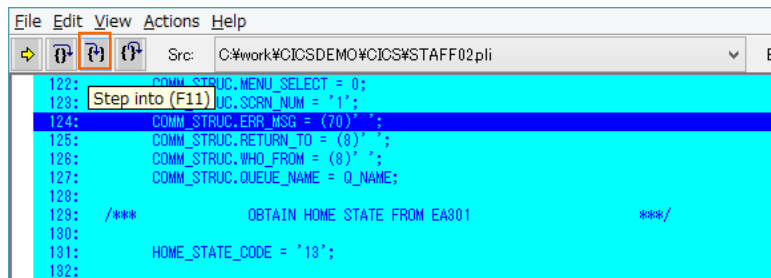
- 3) ログインに成功したら、画面クリア後、PCT 名である TTEA を入力して実行キーを押すと、Codewatch デバッガが自動的に立ち上がってきます。



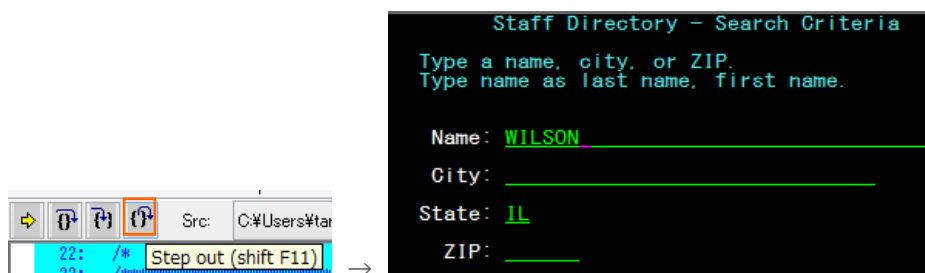
これは前項で確認したプロジェクトの [EXEC プリプロセッサ オプション] の値へ PLITEST を指定したためです。

デバッガの起動を行いたくない場合は、この値をクリアして再ビルドしてください。

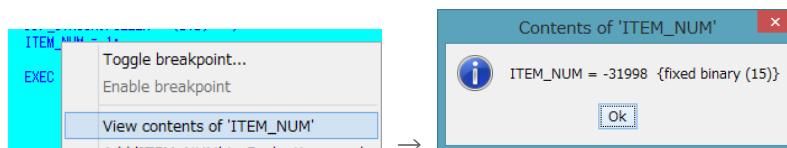
- 4) ステップインを行うことにより、ステップ実行が可能です。



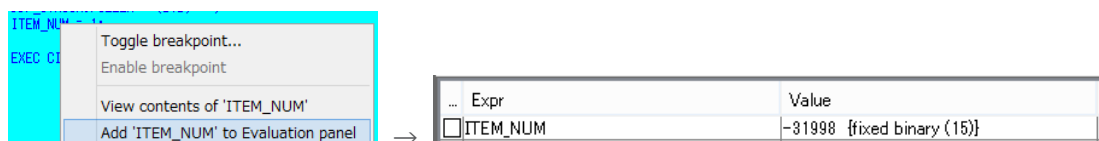
- 5) ステップアウトを行い、アプリケーションを先に進めると前項同様の画面が表示されますので、[Name] 欄へ WILSON と入力して実行キーを押すと、ファイルからデータが読み込まれて表示されます。



- 6) 再度、Codewatch 画面が表示されますので、ステップインします。コード内の変数へマウスオーバーして右クリックし、[View contents of '変数名'] を選択すると、値がポップアップウィンドウで参照できます。



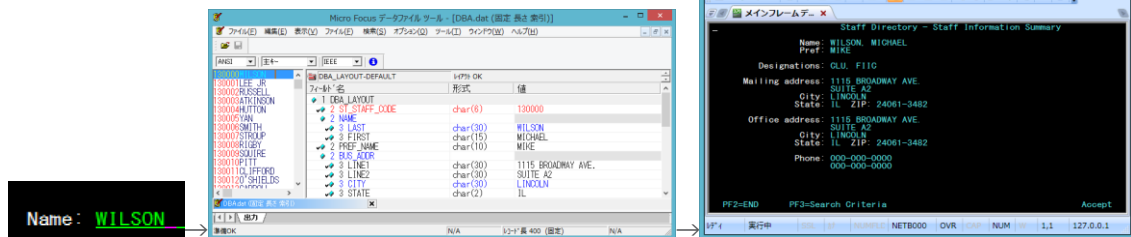
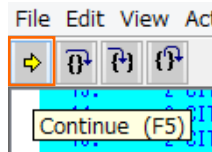
- 7) 変数値を常に監視したい場合には、変数へマウスオーバーして右クリックし、[Add '変数名' to Evaluation panel] を選択すると、右側に表示されているパネルへ常に表示されるようになります。



- 8) ステートメントの行をダブルクリックすることによりブレークポイントの設定が可能です。ブレークポイントには、該当行の左端に赤丸が表示されます。解除も同様にダブルクリックを行います。



- 9) 何度か [Continue] アイコンをクリックすると、情報表示画面が表示されます。



- 10) TN3270 エミュレーターを切断して、デバッグを終了します。

3.9 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、Visual Studio を終了します。

WHAT'S NEXT

- メインフレーム PL/I 開発 : JCL Visual Studio 2017 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。