

# Enterprise Developer チュートリアル

## メインフレーム COBOL 開発 : CICS Visual Studio 2022 編

### 1. 目的

本チュートリアルでは、Visual Studio 2022 を使用したメインフレーム COBOL プロジェクトの作成、コンパイル、CICS を使用したトランザクションの実行、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
- 使用マシンに Micro Focus™ Enterprise Developer 9.0 for Visual Studio 2022 がインストールされていること
- 使用マシンに TN3270 エミュレータがインストールされており、稼働実績があること

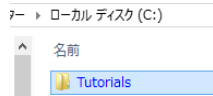
### 3. チュートリアル手順の概要

1. チュートリアルの準備
2. Visual Studio の起動
3. メインフレーム COBOL プロジェクトの作成
4. プロジェクトプロパティの設定
5. ビルドの実行
6. BMS 画面定義の確認
7. CICS リソース定義の概念
8. Enterprise Server インスタンスの設定
9. Enterprise Server インスタンスの開始と確認
10. CICS リソース定義の確認
11. CICS の実行
12. CICS の動的デバッグ
13. Enterprise Server インスタンスの停止

### 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。

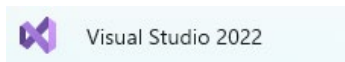
- 1) 使用する例題プログラムは、キットに添付されている Tutorials.zip に圧縮されています。これを C:¥ 直下に解凍します。



- 2) Visual Studio のソリューションを保存する VS という名前のフォルダを C:¥ 直下に作成します。

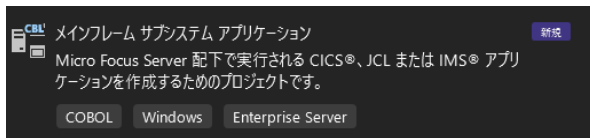
### 3.2 Visual Studio の起動

Visual Studio 2022 を起動します。



### 3.3 メインフレーム サブシステム アプリケーション プロジェクトの作成

- 1) 新しいソリューションとプロジェクトを作成します。[ファイル] プルダウンメニューから [新規作成] > [プロジェクト] を選択して [新しいプロジェクト] ウィンドウを表示し、COBOL が含まれる [メインフレーム サブシステム アプリケーション] を選択後、[次へ] ボタンをクリックします。

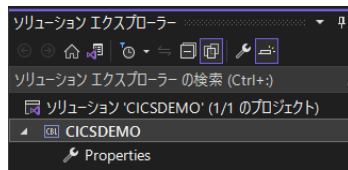


- 2) 下記項目を入力後、[作成] ボタンをクリックします。

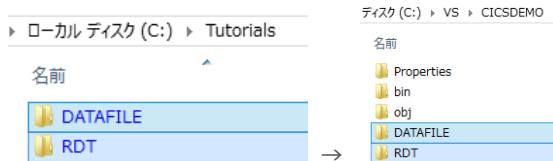
| 項目名                          | 説明                          |
|------------------------------|-----------------------------|
| プロジェクト名                      | 任意ですが、ここでは CICSDEMO を入力します。 |
| 場所                           | 前項で作成した C:¥VS を指定します。       |
| ソリューションとプロジェクトを同じディレクトリに配置する | ここではチェックをオンにします。            |



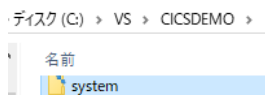
- 3) [ソリューション エクスプローラー] へ作成したプロジェクトが表示されます。



- 4) プロジェクトを作成したことにより C:\VS\CICSDEMO フォルダが作成されていますので、Windows エクスプローラーを使用して C:\Tutorials フォルダ直下の DATAFILE と RDT フォルダを C:\VS\CICSDEMO 配下へコピーします。DATAFILE フォルダには例題プログラムで使用するファイルが、RDT フォルダには例題プログラムで使用する定義済み CICS リソース定義テーブルが含まれています

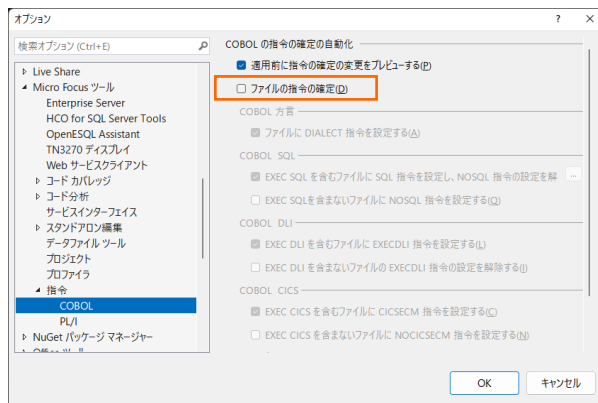


また、実行時に使用する system フォルダを新規作成してください。

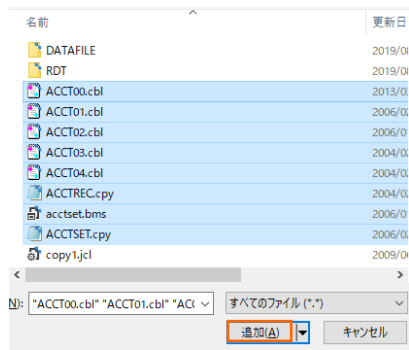


- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。Visual Studio の [ツール] プルダウンメニューの [オプション] を選択してオプションウィンドウを表示します。

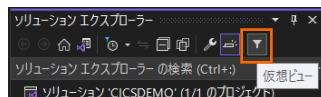
左側ツリービューの [Micro Focus ツール] > [指令] > [COBOL] > [ファイルの指令の確定] チェックボックスをオフにして [OK] ボタンをクリックします。



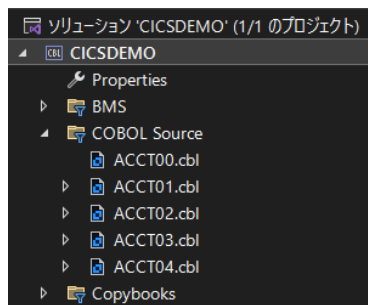
- 6) 用意した例題プログラム類をインポートします。CICSDEMO プロジェクトを右クリックして [追加] > [既存の項目] を選択し、既存項目の追加ウィンドウにて C:\Tutorials を指定すると内容が表示されます。ファイルの種類に [すべてのファイル] を選択後、ファイル名の先頭が ACCT の 8 ファイルを選択して [追加] ボタンをクリックします。この実行により、プロジェクトフォルダへ例題プログラムが配置されます。



- 7) 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。



- 8) [ソリューション エクスプローラー] 内に表示されている CICSDEMO プロジェクトにインポートしたファイルが表示されていることを確認します。



### 3.4 プロジェクトプロパティの設定

この例題は BMS 画面定義、EXEC CICS 文を含むプログラム、COPY メンバーが含まれています。プログラム内容に沿ったプロジェクトのプロパティを設定します。

- 1) [ソリューション エクスプローラー] 内の [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) 左側ツリービューの [アプリケーション] を選択して、生成する実行ファイルを GNT にするため [出力の種類] へ [INT/GNT] を選択します。



- 3) 左側ツリービューの [COBOL] を選択して、下記項目を入力します。

| 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。ここでは [x64] を指定します。                                                                            |
| COBOL 方言          | COBOL 言語方言を指定します。<br>例題プログラムは IBM OS/VS COBOL の方言を使用していますが、COPY 句に G 定数を使用しているためここでは [Micro Focus] を指定します。 |
| 文字セット             | EBCDIC または ASCII を指定します。ここでは [ASCII] を選択します。                                                               |
| 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。                                                                             |
| .GNT にコンパイル       | 実行ファイル形式を GNT に指定するためにチェックをオンにします。                                                                         |
| メインフレーム EXEC CICS | プログラムに EXEC CICS 構文を含むため、チェックをオンにします。                                                                      |
| 出力パス              | 実行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。                                                                             |
| 追加指令              | OSVS を指定します。                                                                                               |

アプリケーション  
SQL  
デバッグ  
従属パス  
BMS  
IMS  
プリプロセッサ  
COBOL  
COBOL リンク  
アセンブラ  
Micro Focus Code Analysis

構成(C): アクティブな (Debug) プラットフォーム(M): アクティブな (x64)

全般

プラットフォームターゲット: x64  
COBOL 方言: Micro Focus 文字セット: ASCII  
ソース フォーマット: 固定

☒ デバッグ用にコンパイル ☒ .GNT にコンパイル  
☐ 複数プロセスによるコンパイル

ビルド イベント...

メインフレーム

☒ EXEC CICS オプション... ☐ EXEC DLI オプション...

エラーおよび警告

警告レベル: 回復可能なエラーを含める(レベル E)  
最大エラー数: 100 ☐ 警告をエラーとして処理

出力

出力パス: %bin%x64%Debug% 参照...

☐ 指令ファイルの生成 ☐ リストファイルの生成  
☐ コード カバレッジを有効にする ☐ プロファイラを有効にする

追加指令

OSVS

- 4) 左側ツリービューの [デバッグ] を選択して、[プロファイル] には CICSDEMO を、[起動] には CICS を指定します。

アプリケーション  
SQL  
デバッグ  
従属パス  
BMS  
IMS  
プリプロセッサ  
COBOL  
COBOL リンク  
アセンブラ

構成(C): なし プラットフォーム

プロファイル: CICSDEMO  
起動: CICS

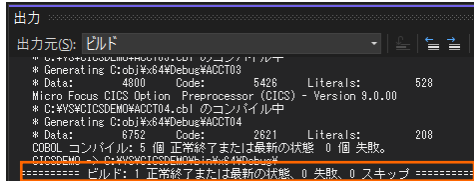
ユーザー:  
端末:  
トランザクション:  
リンク レベル プログラム:

- 5) プロパティファイルを上書き保存します。

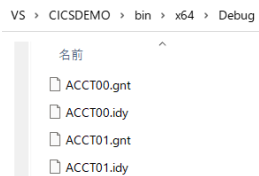


### 3.5 ビルドの実行

- 1) [ソリューション エクスプローラー] の CICSDEMO 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。



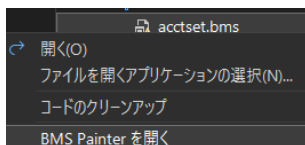
- 3) 前項で確認した出力パスへ、実行ファイル (.gnt) や マップファイル (.mod) が作成されていることを確認します。



### 3.6 BMS 画面定義の確認

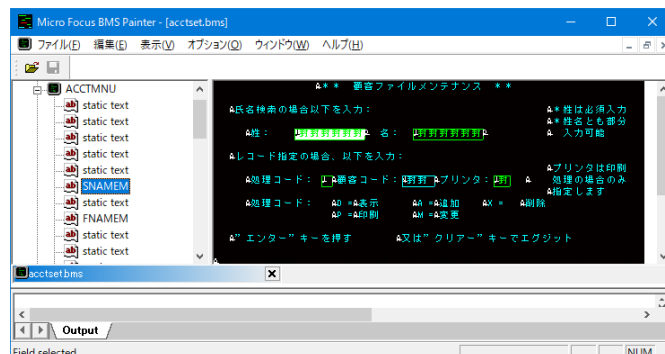
Enterprise Developer には CICS 開発者のために BMS 画面を対話型で編集するユーティリティが装備されています。

- 1) [ソリューション エクスプローラー] のプロジェクト内に存在する acctset.bms ファイルを右クリックして [BMS Painter を開く] 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確認できます。

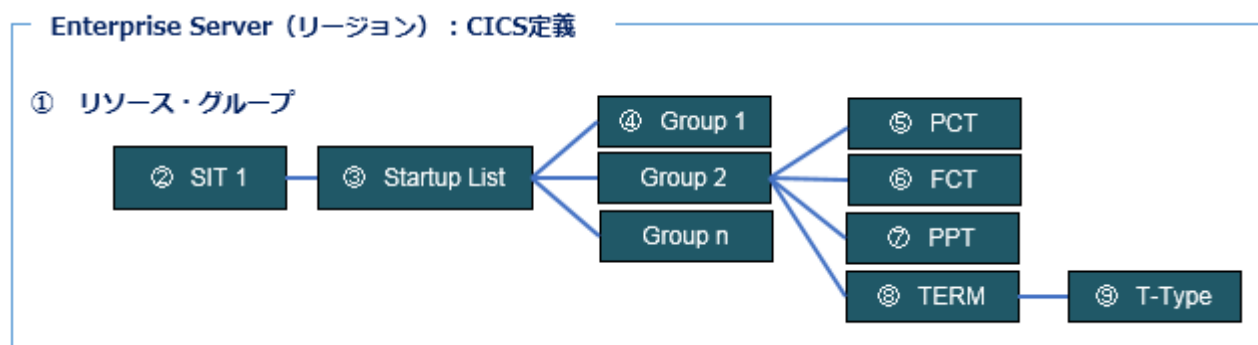
左側のツリービューでオブジェクトを選択すると右側のグラフィカルビュー内で対応するオブジェクトがハイライトされます。



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

### 3.7 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。

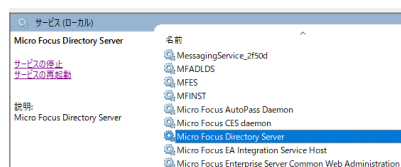
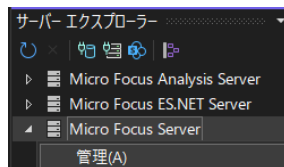


| 項目名            | 説明                                    |
|----------------|---------------------------------------|
| ① リソース・グループ    | CICS リソースのセットを指します。                   |
| ② SIT          | CICS の詳細設定が指定される、システム初期化テーブルを指します。    |
| ③ Startup List | CICS インスタンス起動時、自動的にロードされるグループ一覧を指します。 |
| ④ Group        | 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。                           |

### 3.8 Enterprise Server インスタンスの設定

Enterprise Server インスタンスには CICS をエミュレーションする機能が搭載されており、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) 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 Enterprise Server Common Web Administration (以降 ESCWA) が表示されます。ESCWA 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します。

Default という名前の接続が表示されていない場合は新規に作成します。表示されている場合は作成する必要はありませんので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。



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 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 \*

Default

ホスト \*

localhost

ポート \*

86

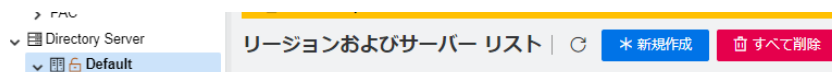
説明

入力必須の項目です

保存

戻る

- 3) Enterprise Server インスタンスを追加します。  
[Default] をクリックし、右側ペインの [新規作成] ボタンをクリックします。



- 4) [リージョンの新規作成] 項目の [名前]、[説明] は任意ですが、ここでは名前に CICSDEMO、説明に CICS チュートリアル用と入力します。Visual Studio では実行可能ファイルは 64 ビットを指定してコンパイルしたため、稼働させる Enterprise Server インスタンスも同様に [64 ビット作業モード] へチェックを入れます。これにより警告が表示されますが無視して先に進んでください。[MSS 有効]、[TN3270 リスナーの作成] にチェックが入っていることを確認し、[TN3270 リスナーポート] へ 9004 を指定して [保存] ボタンをクリックします。

リージョンの新規作成

名前・  
CICSDEMO

説明  
CICS チュートリアル用

▲ Directory Serverとリージョンの作業モードが一致しません。プラットフォームによっては、起動時に不具合が発生する可能性があります。

☒ 64ビット作業モード  
☒ MSS有効  
☒ TN3270リスナーの作成

TN3270リスナーポート 9004

・入力必須の項目です

保存 戻る

**情報**

ポート番号には 1025 より小さい番号を指定しないでください。  
指定したポート番号で TN3270 リスナーが作成されます。

- 5) 64 ビットアプリケーション稼働用の CICSDEMO インスタンスが作成され、一覧に表示されます。CICSDEMO インスタンスにカーソルを合わせ、[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 |                                                                                     |
|----------|--------|---------|-------|-------|--------|-------------------------------------------------------------------------------------|
| CICSDEMO | Region | Stopped | ✓     | ✓     | デフォルト  |  |
|          |        |         |       |       |        | 編集                                                                                  |

- 6) CICSDEMO 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。

開始オプション

名前・? システムディレクトリ ?

CICSDEMO C:\VS\CICSDEMO\system

☐ ローカルコンソールを表示 ☒ 動的デバッグを許可 ☐ システム起動時に開始する

☒ 64ビット作業モード ☐ 以前のログを削除

- 7) 例題では日本語半角カナ表示を有効にするため、[追加設定] の [構成情報] 欄に下記内容を入力し、[適用] ボタンをクリックします。

[ES-Environment]  
MFICODESET=9122

追加設定

構成情報 ?

[ES-Environment]  
MFICODESET=9122

**重要**

入力値は全て半角英数字で指定してください。

- 8) 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設定します。構成情報に指定した環境変数を使用して値を入力後、【適用】ボタンをクリックします。

| 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。ここでは例題に含まれている DBCS 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。ここでは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。ここでは .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。ここでは例題で用意されている VSAM 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** | **適用**

システム初期化テーブル (SIT) リソース定義ファイルパス

DBCS C:\VS\CICSDemo\IRDT

トランザクションパス

C:\VS\CICSDemo\bin\x64\Debug

ファイルパス

C:\VS\CICSDemo\DATAFILE

マップパス

C:\VS\CICSDemo\bin\x64\Debug

☐ EZASOCKET サポート

**重要**

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。

- 9) 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。

**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

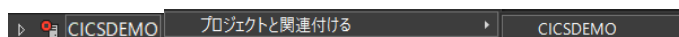
- 10) 画面左側ペインの【Default】をクリックして一覧画面に戻ります。

▼ Directory Server

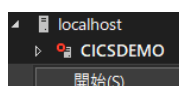
▼ **Default**

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

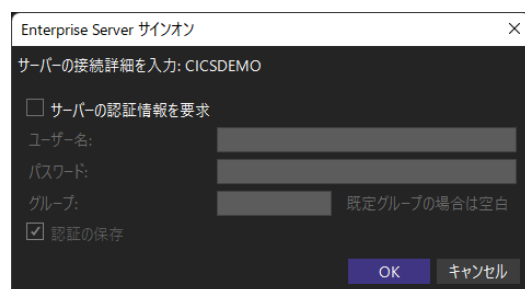
- 1) Visual Studio に戻り、[サーバーエクスプローラー] 内に CICSDEMO インスタンスが表示されていることを確認します。表示されていない場合は [Micro Focus Server] を右クリックし、[最新の情報に更新] を選択してリフレッシュしてください。
- 2) [サーバーエクスプローラー] 内の CICSDEMO インスタンスを右クリックし、[プロジェクトと関連付ける] > [CICSDEMO] を選択します。これにより CICSDEMO プロジェクトから実行されるアプリケーションは CICSDEMO インスタンスで処理されることになります。



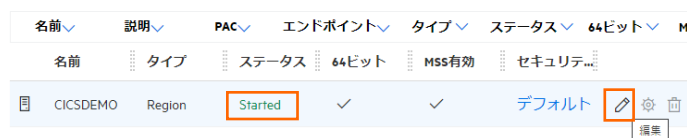
- 3) CICSDEMO インスタンスを右クリックして [開始] を選択します。



- 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。

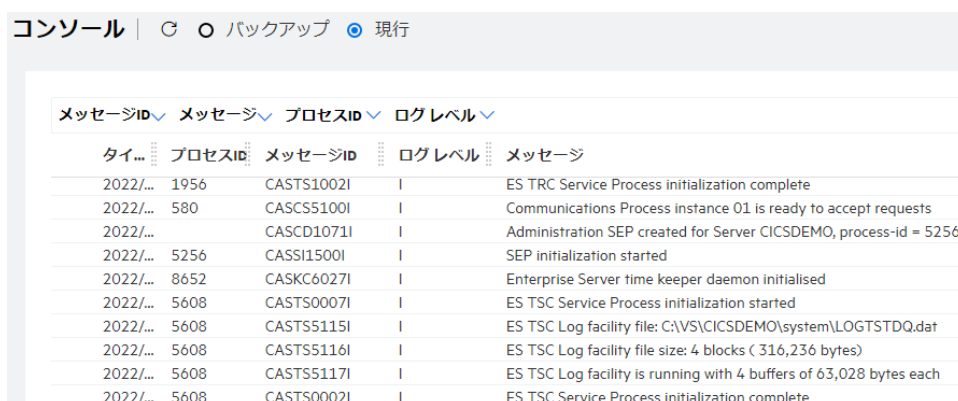


- 5) ESCWA へ移動して開始状態であることを確認後、[編集] アイコンをクリックします。



- 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。





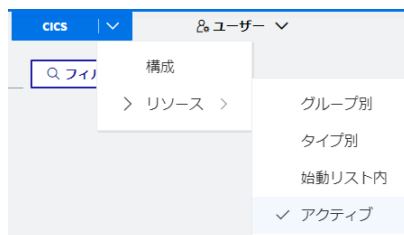
## 注意

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。

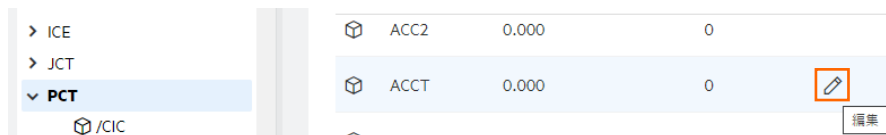
### 3.10 CICS リソース定義の確認

メインフレームの CICS と同様に Enterprise Server インスタンスでも各種リソース定義をオンラインで参照・更新・追加・削除することが可能です。本チュートリアルでは定義済みのリソースファイルを使用していますので、その内容を参照してみます。

- 1) ESCWA の上部にある【CICS】プルダウンメニューから【リソース】>【アクティブ】を選択します。



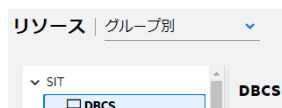
- 2) 左側ペインの【PCT】をクリックすると現在アクティブな PCT 一覧が表示されますので、【ACCT】の編集アイコンをクリックします。



- 3) ACCT トランザクションを呼び出すと ACCT00 プログラムが呼び出されることやステータスが有効であることが確認できます。



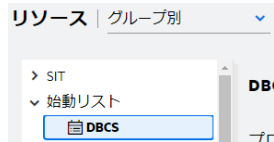
- 4) 次に、表示形式を【グループ別】に変更し、左側ペインから【SIT】を展開後、SIT で指定した【DBCS】をクリックします。



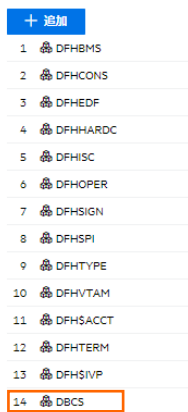
- 5) DBCS SIT には【始動リスト】として同名の定義が、【開始 Tran】には CESN (CICS サインオン・トランザクション)が指定してあることが確認できます。



- 6) 上記で【始動リスト】に指定してある DBCS の内容を確認します。左側ペインの【始動リスト】を展開し、【DBCS】をクリックします。



- 7) DBCS 始動リストにはリソース定義類が属するグループが指定してあり、同じ名称の定義が異なるグループに存在する場合は後述グループが優位になります。この始動リストでは 14 番目に指定されている DBCS グループに属するリソース定義類が最も優位となります。詳しくは【CICS SIT 構成チュートリアル】をご参照ください。



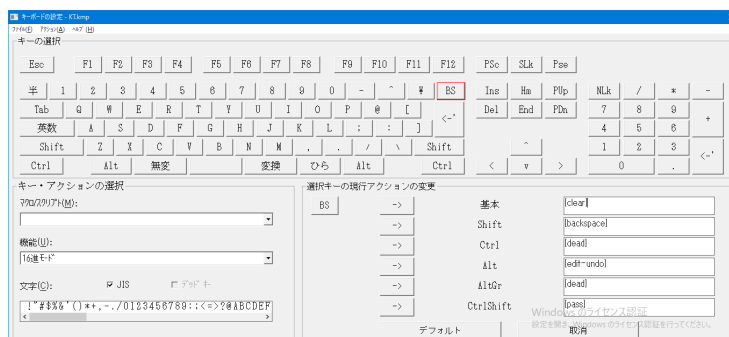
### 3.11 CICS の実行

現在 CICSDEMO インスタンスが稼働していますので、例題プログラムを実行することができます。

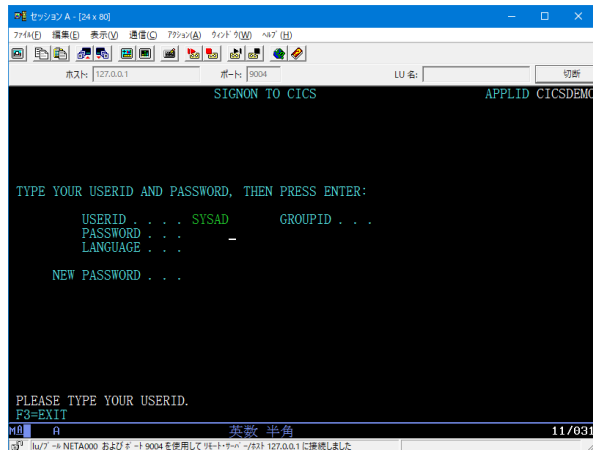
ご使用の TN3270 エミュレータを前項で作成した TN3270 リスナーポート(localhost:9004)へ接続します。

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。



- 2) 正常にサインオンできましたら、クリアキーで画面をクリアします。

CASSE0012I SIGNON COMPLETE AT C000, FOR USER SYSAD. LOCAL SECURITY IS DISABLED.

- 3) PCT に登録されていたトランザクションの ACCT を入力して実行キーを押します。

ACCT\_

- 4) ACCT トランザクションからプログラムが呼ばれて、例題の BMS ファイルに定義されていた下記初期画面が表示されます。

\*\*\* 顧客ファイルメンテナンス \*\*\*

氏名検索の場合以下を入力：

姓：  名：

レコード指定の場合、以下を入力：

処理コード： ☐ 顧客コード：  プリンタ：

処理コード： D = 表示 A = 追加 X = 削除  
P = 印刷 M = 変更

”エンター”キーを押す 又は”クリア”キーでエグジット

\* 姓は必須入力  
\* 姓もとも部分入力可能

プリンタは印刷処理の場合のみ指定します

- 5) Tab キーで入力フィールドを移動して、[処理コード] へ D を、[顧客コード] へ 11111 を入力して実行キーを押します。

\*\*\* 顧客ファイルメンテナンス \*\*\*

氏名検索の場合以下を入力：

姓：  名：

レコード指定の場合、以下を入力：

処理コード： ☒ 顧客コード：  プリンタ：

処理コード： D = 表示 A = 追加 X = 削除  
P = 印刷 M = 変更

”エンター”キーを押す 又は”クリア”キーでエグジット

\* 姓は必須入力  
\* 姓もとも部分入力可能

プリンタは印刷処理の場合のみ指定します

- 6) 例題ファイルから指定顧客コードを持つデータが検索され、表示されます。

顧客ファイル レコード表示

顧客番号: 11111 姓: 柴  
 電話: 0771778888 住所: 式部 4-3、石山寺  
 大津市  
 滋賀県

MI: G 敬称: MRS

その他の請求先:

発行カード枚数: 1 発行日: 06 07 07 理 由: N  
 カードコード: X 承認者: GNG 特別コード:

顧客状況: N 請求限度額: 1000.00

| 履歴: | 残高   | 請求日      | 請求額  | 支払日      | 支払額  |
|-----|------|----------|------|----------|------|
|     | 0.00 | 00/00/00 | 0.00 | 00/00/00 | 0.00 |
|     | 0.00 | 00/00/00 | 0.00 | 00/00/00 | 0.00 |
|     | 0.00 | 00/00/00 | 0.00 | 00/00/00 | 0.00 |

表示終了なら”クリア”か”エンター”を押す

7) 実行キーを押して前画面へ戻り、TN3270 エミュレータを切断します。

### 3.12 CICS 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。

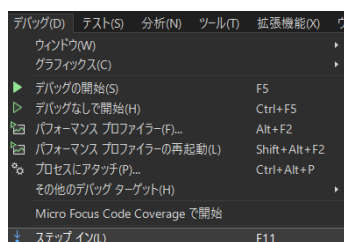
- 1) [ソリューション エクスプローラー] 内に存在する CICSDEMO プロジェクトの ACCT00.cbl をダブルクリックして内容を表示します。EXEC CICS 構文を使用して MAP の SEND と、次トランザクションを呼び出していることがわかります。

```

ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
PROCEDURE DIVISION.
INITIAL-MAP.
    EXEC CICS SEND
        MAP('ACCTMNU')
        MAPSET('ACCTSET') FREEKB
        ERASE MAPONLY
    END-EXEC
    EXEC CICS RETURN TRANSID('AC01') END-EXEC
GOBACK.

```

- 2) これらのプログラムをステップ実行します。[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。



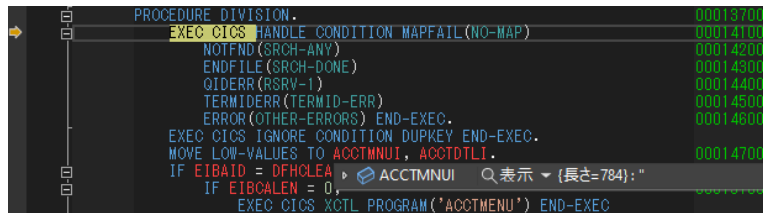
- 3) 画面左下が【準備完了】となり、アタッチ待機状態になったことを確認します。



- 4) 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] 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。

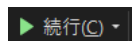
なお、この例題プログラムでは TN3270 エミュレータとの画面送受信がありますので、その都度、表示を切り替えてデバッグします。



- 5) 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。



- 6) 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。



- 7) 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。

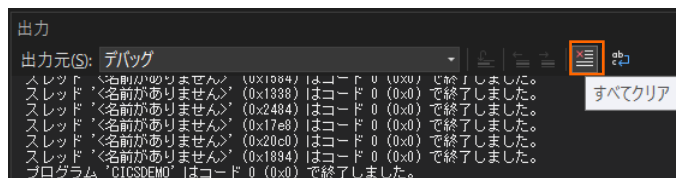


- 8) TN3270 エミュレータを切断します。

- 9) 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。



- 10) 出力タブに表示されているデバッグ履歴を消去するには、[すべてクリア] アイコンをクリックします。



### 3.13 Enterprise Server インスタンスの停止

- 1) CICSDEMO インスタンスを停止します。



- 2) CICSDEMO インスタンスの停止を確認後、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。

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。