

# Enterprise Developer チュートリアル

## メインフレーム COBOL 開発 : CICS Eclipse 編

### 1 目的

本チュートリアルでは、Eclipse 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

1. 本チュートリアルで使用したマシン OS : Windows 11 Pro
2. 使用マシンに Enterprise Developer 11J for Eclipse がインストールされていること
3. 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. Eclipse の起動
3. メインフレーム COBOL プロジェクトの作成
4. テキストファイルのエンコード指定
5. プロジェクトプロパティの設定
6. ビルドの実行
7. BMS 画面定義の確認
8. CICS リソース定義の概念
9. Enterprise Server インスタンスの設定
10. Enterprise Server インスタンスの開始と確認
11. CICS リソース定義の確認
12. CICS の実行
13. CICS の動的デバッグ
14. Enterprise Server インスタンスの停止

### 4 免責事項

### 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) Eclipse のワークスペースで使用する work フォルダを C:¥直下に作成します。

### 3.2 Eclipse の起動

- 1) メニューから 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、[起動] ボタンをクリックします。



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



Enterprise Developer for Eclipse によるこそ



概要 フィーチャーの概要

Web リソース Web 上の詳細情報

Open Team Developer Perspective  
Open the Team Developer perspective, which is the default perspective. This perspective provides tooling to simplify access to your mainframe resources; it also provides tools to develop COBOL and PL/I applications.

ファースト・ステップ ファースト・ステップの開始

移行 新規リリースへの移行

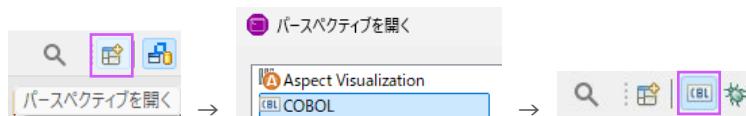
Open COBOL Perspective  
Open the COBOL perspective. This perspective provides tools which enable you to develop COBOL applications.

Open PL/I Perspective  
PL/I パースペクティブを開きます。このパースペクティブでは、PL/I アプリケーションを開発するためのツールを提供します。

- 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。

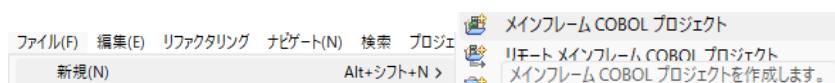


- 5) COBOL パースペクティブが開かない場合は Eclipse 右上の [パースペクティブを開く] アイコンをクリックして表示後、[COBOL] を選択して [開く] ボタンをクリックします。

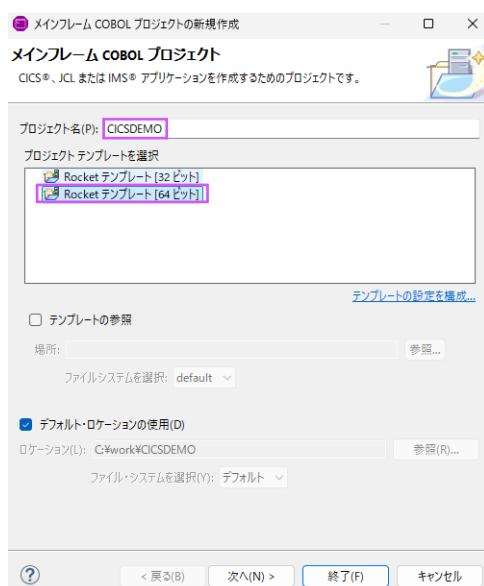


### 3.3 メインフレーム COBOL プロジェクトの作成

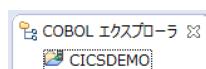
- 1) 用意した例題ソースをインポートするためのプロジェクトを作成します。[ファイル] プルダウンメニューから [新規] > [メインフレーム COBOL プロジェクト] を選択します。



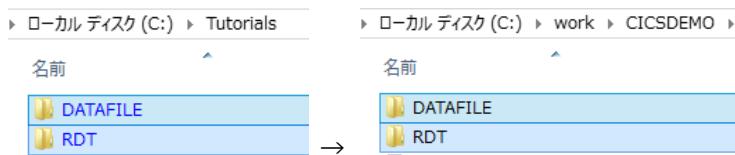
- 2) [プロジェクト名] は任意ですが、ここでは CICSDemo を入力し、テンプレートは 64 ビットを選択後 [終了] ボタンをクリックします。



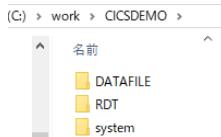
- 3) COBOL エクスプローラーへ作成したプロジェクトが表示されます。



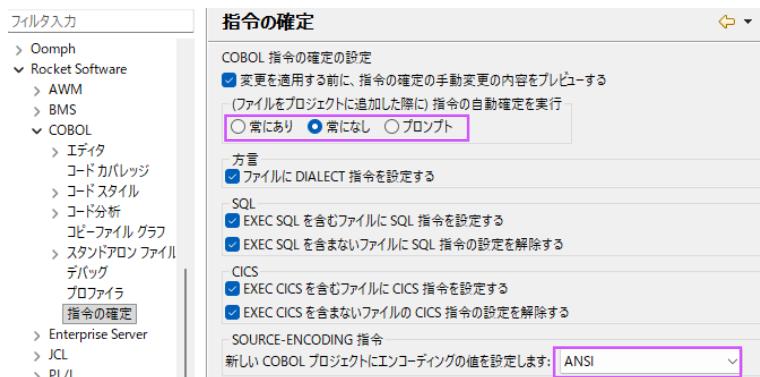
- 4) プロジェクトを作成したことにより C:\work\CICSDEMO フォルダが作成されていますので、C:\Tutorials フォルダにある DATAFILE と RDT フォルダを、C:\work\CICSDEMO へ Windows エクスプローラーを使用してコピーします。DATAFILE には例題プログラムで使用するファイルが、RDT には例題プログラムで使用する定義済みの CICS リソース定義テーブルが含まれています。



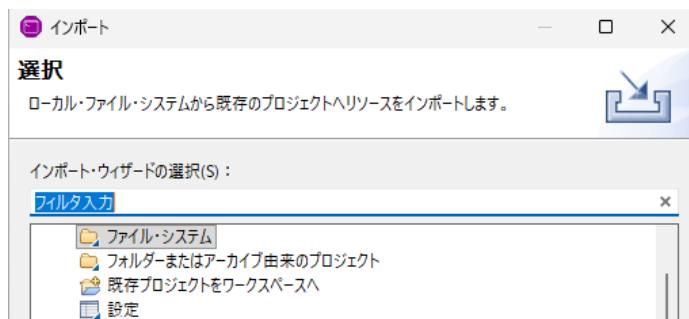
また、実行時に使用する system フォルダを C:\work\CICSDEMO フォルダに新規作成してください。



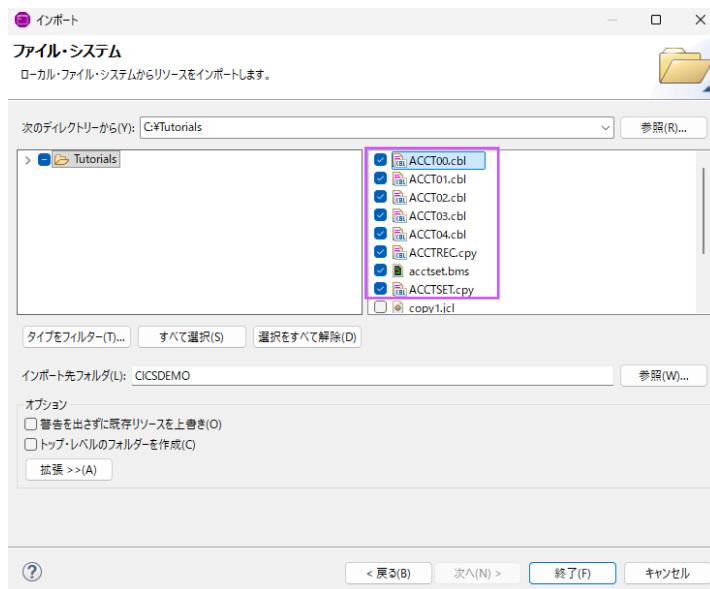
- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ウィンドウ] プルダウンメニューの [設定] > [Rocket Software] > [COBOL] > [指令の確定] > [COBOL 指令の確定の設定] では [常なし] を選択し、[SOURCE-ENCODING 指令] では [ANSI] を選択後、[適用して閉じる] ボタンをクリックします。



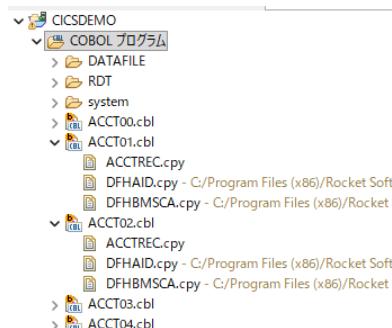
- 6) 用意した例題プログラム類をインポートします。CICSDEMO プロジェクトを右クリックして [インポート] > [インポート] を選択し、インポートウィンドウにて [一般] > [ファイル・システム] を選択後 [次へ] ボタンをクリックします。



- 7) C:\¥Tutorials を [次のディレクトリーから] へ指定すると内容が表示されますので、ファイル名の先頭に ACCT が付く上部 8 ファイルのチェックをオンにして [終了] 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。



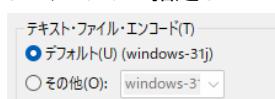
- 8) COBOL エクスプローラー内に表示されている CICSDEO プロジェクトにインポートしたファイルが表示されることを確認します。



### 3.4 テキストファイルのエンコード指定

Eclipse ではワークスペースの設定として、テキストファイルのエンコードを指定できます。ソースファイル類のエンコードに沿って適切なエンコードを指定してください。この例題では Windows-31j を使用します。

- 1) Eclipse の [ウィンドウ] プルダウンメニューから [設定] を選択し、設定ウィンドウを表示します。
- 2) 左側ペインで [一般] > [ワークスペース] を選択し、右側ペインの [テキスト・ファイル・エンコード] に MS932 または Windows-31j が指定されていることを確認します。「デフォルト」がこの値ではない場合は、「その他」を選択して、これらを指定してください。



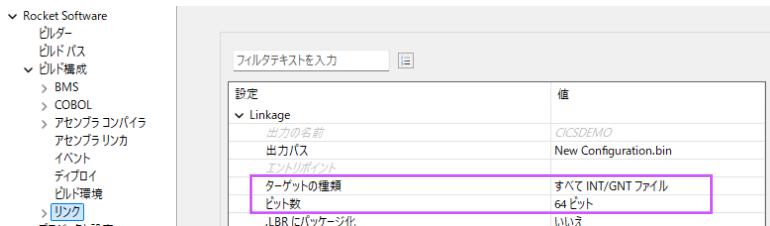
- 3) 指定後は [適用して閉じる] ボタンをクリックします。

### 3.5 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) COBOL エクスプローラー内の [CICSDEMO] プロジェクトを右クリックして [プロパティ] を選択します。
- 2) 左側ツリービューの [Rocket Software] > [ビルド構成] > [リンク] を選択後、下記項目を指定して [適用] ボタンをクリックしてください。

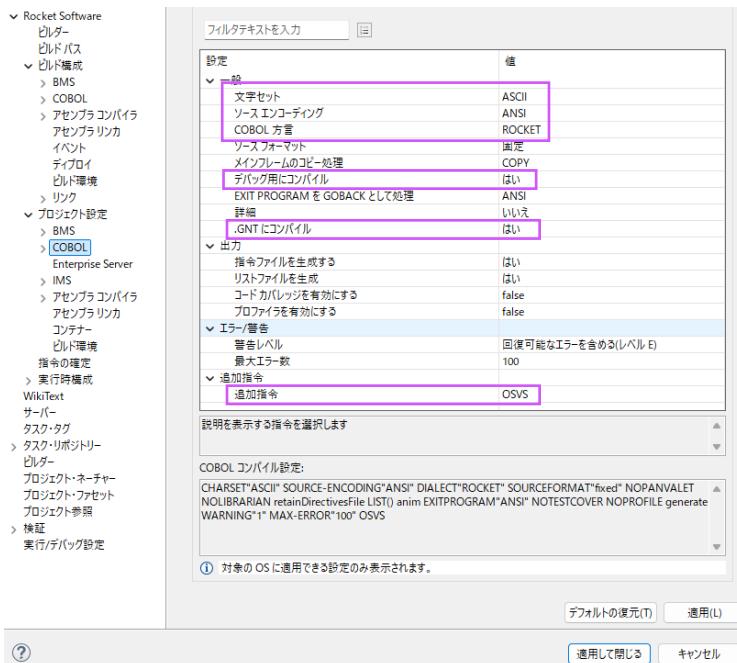
項目名	説明
ターゲットの種類	実行ファイル形式を指定します。 ここでは [すべて INT/GNT ファイル] を選択します。
ビット数	稼働ビット数を指定します。ここでは [64 ビット] を選択します。



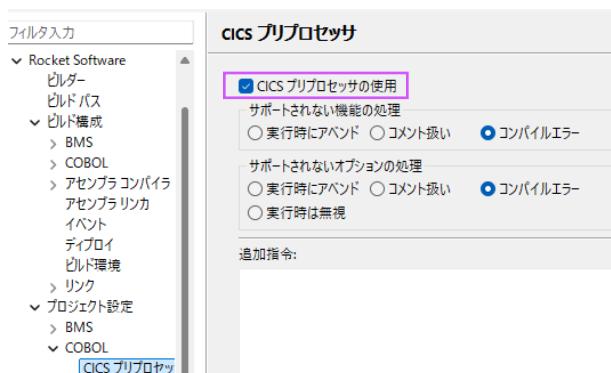
- 3) 左側ツリービューの [Rocket Software] > [プロジェクト設定] > [COBOL] を選択して、下記項目を指定します。指定後は [適用] ボタンをクリックしてください。

項目名	説明
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。
ソースエンコーディング	Eclipse に指定したエンコードと一致するエンコーディングを指定します。 MS932 または Windows-31j と一致するように ANSI を指定します。
言語方言	COBOL 言語方言を指定します。 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 [Rocket] 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 [はい] を選択します。
.GNT にコンパイル	実行ファイル形式を GNT に指定するため [はい] を選択します。
追加指令	ここでは OSVS を入力します。

情報
“.GNT” は 製品独自の動的にロード可能なファイルであり、実行可能ファイルにリンクする必要がありません。

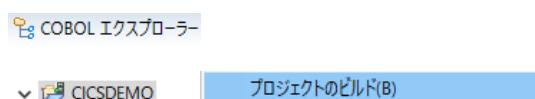


- 4) 左側ツリービューの [Rocket Software] > [プロジェクト設定] > [COBOL] > [CICS プリプロセッサ] を選択して、[CICS プリプロセッサの使用] チェックボックスをオンにします。[適用して閉じる] ボタンをクリックしてウィンドウを閉じます。

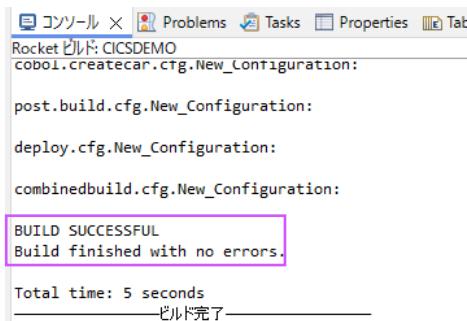


### 3.6 ビルドの実行

- 1) COBOLエクスプローラー内の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。



- 2) コンソールタブで成功を確認します。



```

コンソール × Problems Tasks Properties Tab
Rocket ビルド CICSDEMO
cobo1.createcar.cfg.New_Configuration:
post.build.cfg.New_Configuration:
deploy.cfg.New_Configuration:
combinedbuild.cfg.New_Configuration:
BUILD SUCCESSFUL
Build finished with no errors.

Total time: 5 seconds
-----ビルト完了-----

```

- 3) COBOL エクスプローラーのプロジェクト内に存在する New\_Configuration.bin フォルダ配下に実行ファイル(.gnt ファイル)や BMS マップ(.mod ファイル)が作成されていることを確認してください。



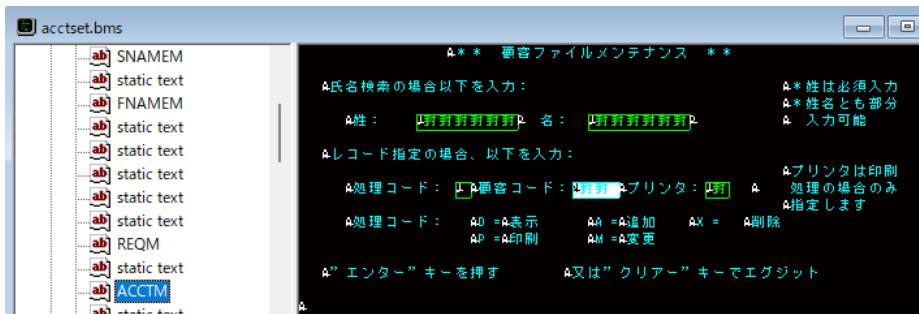
### 3.7 BMS 画面定義の確認

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ーが装備されています。

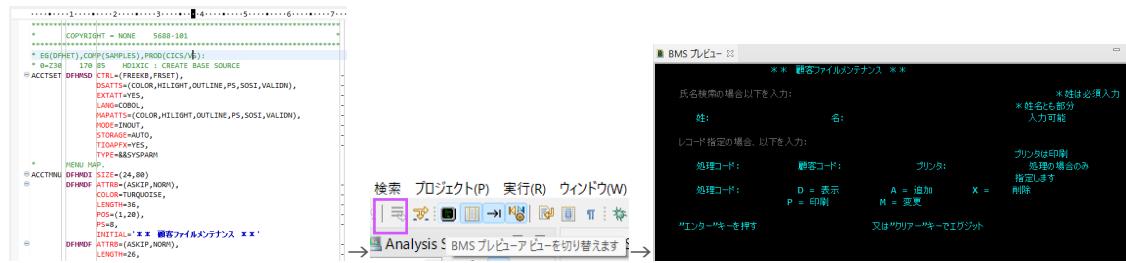
- 1) COBOL エクスプローラーのプロジェクト内に存在する BMS ソースファイルフォルダの acctset.bms ファイルを右クリックして【アプリケーションから開く】>【BMS ペインタ】を選択します。



- 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。

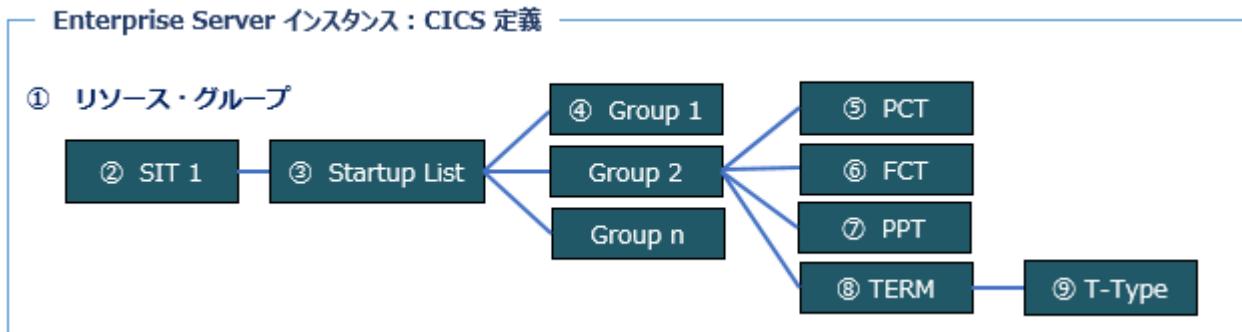


- 3) 【ファイル】 プルダウンメニューの【終了】 を選択して BMS ペインタウィンドウを終了します。
- 4) BMS ソースを BMS エディタで開き【BMS プレビュー】 アイコンをクリックすると、画面のプレビューを表示させることもできます。表示させたいマップにカーソルを合わせると表示画面が切り替わります。



### 3.8 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。

### 3.9 Enterprise Server インスタンスの設定

Enterprise Developer は CICS のエミュレーション機能を搭載している開発用 Enterprise Server インスタンスを内包しており、各開発者がこのインスタンスを占有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うことができます。本番環境にはコンパイラーなどを含まないランタイムのみの実行環境製品の 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。

- Enterprise Server インスタンスを運用、管理する Enterprise Server Common Web Administration (以降 ESCWA) は、製品が提供する VSAM 外部セキュリティマネージャー (ESM) モジュールによるセキュリティがデフォルトで有効になっており、すべての処理において実行ユーザーの認証が行われます。

まずはデフォルトユーザーと初期パスワードを取得するために、スタートメニューから [Enterprise Developer] を選択し、[Enterprise Developer コマンドプロンプト] を起動します。



Enterprise Developer コマンドプロンプトから次のコマンドを実行して、デフォルトユーザーと初期パスワードを取得します。

```
mfsecretsadmin read microfocus/temp/admin
```

```
C:\$Users\$tarot\$Documents>mfsecretsadmin read microfocus/temp/admin
[{"mfUser": "SYSAD", "mfPassword": "CKfa+xWt"}]
```

上記例の場合、SYSAD がデフォルトユーザー、CKfa+xWt が初期パスワードです。

この情報は ESCWA のログオン時に使用しますので、記憶しておいてください。

デフォルトセキュリティを無効にすることもできますが、安全を確認後に実施してください。

詳しくは製品マニュアルの [ここからはじめよう] > [Getting Started] にある [デフォルトセキュリティの構成] チュートリアルをご参照ください。

- 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 [localhost] を右クリックして [管理ページを開く] を選択します。表示されていない場合は、Windows サービスの [Micro Focus Directory Server] が開始されているかを確認し、停止している場合は開始してください。



- Web ブラウザが立ち上がり、ユーザー認証を求められます。

パスワードを変更してログオンします。

[パスワード変更] をクリックし、前項で取得したデフォルトユーザーと初期パスワード、新しいパスワードを入力して [サブミット] をクリックします。

TN3270 エミュレータに設定されたコードページによって、自動的に半角英小文字が半角英大文字に変換されることがあります。この場合は、パスワードに含まれる半角英小文字と CICS サインオン時のパスワードが不一致となり、CICS にサインオンできません。

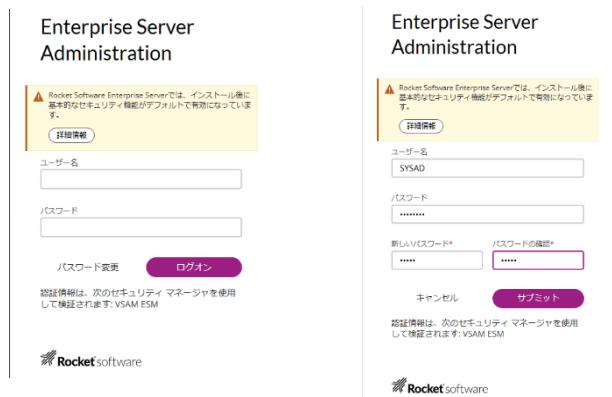
これを回避するために、新パスワードの英字はすべて大文字で入力してください。

良い新パスワードの例) SYSAD123

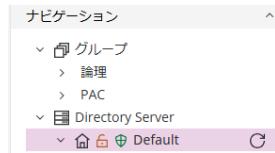
悪い新パスワードの例) sysad123

変更後のパスワードはご自身の責任で管理してください。

パスワード変更後、ESCPWA がタイムアウトした場合は新しいパスワードを使用してログオンしてください。



- 4) メニューで [オペレーション] を選択後、[ナビゲーション] に [Default] という名前の接続が表示されます。表示されない場合は新規に作成します。表示されている場合はこの項目はスキップしてください。

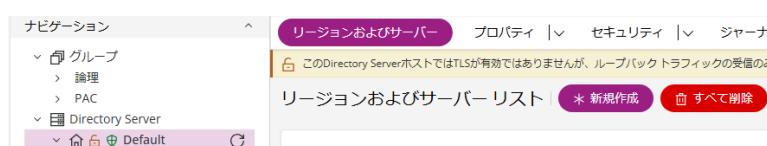


左側ペインで [Directory Server] をクリックし、右側ペインの [追加] ボタンをクリックします。



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 ホスト名またはマシンの 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。

- 5) 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。



- 6) [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。Eclipse の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

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。

名前*	CICSDEMO
説明	CICSチュートリアル用
<input checked="" type="checkbox"/> 64ビット作業モード <input checked="" type="checkbox"/> MSS 有効 <input checked="" type="checkbox"/> 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

- 7) 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。

ア...	名前	タ...	ステータス	64ビット	MSS有効	セキュ...	アクション		
■	CICSDEMO	Region	Stopped	✓	✓	デフォルト			

- 8) 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。



- 9) 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 フォルダを指定します。

開始オプション      \* 入力必須の項目です

名前*	CICSDEMO	システム ディレクトリ	C:\work\CICSDEMO\system
-----	----------	-------------	-------------------------

- 10) 表示画面の下にある【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 からの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 checked="" type="checkbox"/> 64ビット作業モード	<input type="checkbox"/> 以前のログを削除
<input type="checkbox"/> システム起動時に開始する	

- 11) 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ます。また、プロジェクトのパスを指定する環境変数も設定し、最後に [適用] ボタンをクリックします。

入力値)

[ES-Environment]

MFCODESET=9122

proj=C:\work\CICSDEMO

追加設定

 **重要**

入力値はすべて半角英数字で指定してください。

構成情報 

[ES-Environment]  
MFCODESET=9122  
proj=C:\work\CICSDEMO

- 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。

 **重要**

入力値はすべ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。

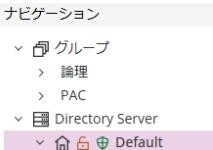
CICSの構成 

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	\$proj\RD
トランザクション パス 	
\$proj\New_Configuration.bin	
ファイル パス 	
\$proj\DATAFILE	
マップ パス 	
\$proj\New_Configuration.bin	
<input type="checkbox"/> EZASOCKET サポート 	

- 13) セキュリティの観点から、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。

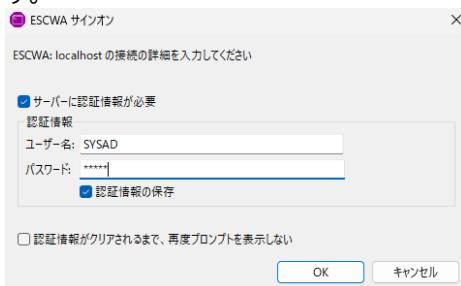


- 14) 画面左側ペインの [Default] をクリックして一覧画面に戻ります。



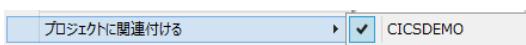
### 3.1 Enterprise Server インスタンスの開始と確認

- 1) Eclipse に戻り、サーバーエクスプローラー内の [localhost] をクリックすると、ESCW Aへの認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。

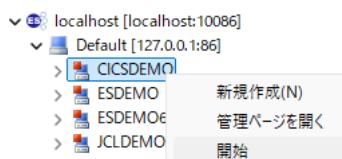


- 2) [Default] を展開して作成した CICSDEMO インスタンスが表示されていることを確認します。表示されていない場合は [Default] を右クリックし、[更新] を選択してリフレッシュしてください。

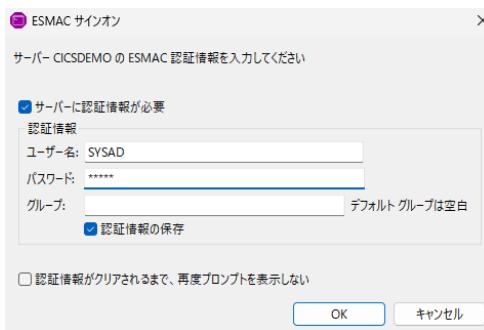
- 3) サーバーエクスプローラー内の CICSDEMO インスタンスを右クリックし、[プロジェクトに関連付ける] > [CICSDEMO] を選択します。これにより CICSDEMO プロジェクトから実行されるアプリケーションは CICSDEMO インスタンスで処理されることになります。



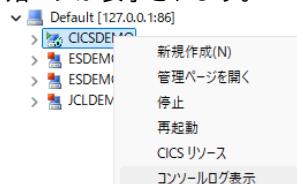
- 4) CICSDEMO インスタンスを右クリックして [開始] を選択します。



- 5) 開始ユーザーの認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。



- 6) 開始後、CICSDEMO インスタンスを右クリックして [コンソールログ表示] を選択すると、[コンソール] タブに開始ログが表示されます。



メッセージコードの最終桁のログレベルが I はインフォメーション、W は警告、S や E の場合はエラーです。

```

Enterprise Server コンソールログ: localhost/Default/CICSDEMO
251204 15423109 12844 CICSDEMO CASS1001I ES support for IBM(R) CICS(R) WEBSERVICES is enabled 15:42:31
251204 15423118 12844 CICSDEMO CASS1745I Processing resources in startup list DBCS 15:42:31
251204 15423120 12844 CICSDEMO CASS1000I Region will use SYSID $IVP, local CCSID 00037 and ascii CCSID 00437 15:42:31
251204 15423123 12844 CASC0111I Enterprise Server time keeper daemon created for Server CICSDEMO, process-id = 10656 15:42:31
251204 15423123 12844 CASC01093I ES TR Service Process created for Server CICSDEMO, process-id = 11588 15:42:31
251204 15423123 12844 CASC01073I ES TSC Service Process created for Server CICSDEMO, process-id = 8392 15:42:31
251204 15423138 12844 CASC01033I ES Communications Server created, ES CICSDEMO, process-id = 14524 15:42:31
251204 15423139 12844 CASC01033I ES Communications Server created, ES CICSDEMO, process-id = 14524 15:42:31
251204 15423140 10656 CICSDEMO CASKC6027I Enterprise Server time keeper daemon initialized 15:42:31
251204 15423142 12844 CICSDEMO CASKC1000I ES concurrent request limit: 000000010 15:42:31
251204 15423143 12844 CICSDEMO CASS1000I Server Manager initialization completed successfully 15:42:31
251204 15423143 12844 CICSDEMO CASS1400I Single signon is not enforced for IBM(R) CICS(R) clients and is not enforced for IMS TM clients for this startup 15:42:31
251204 15423145 12844 CASC0112I SEP 00001 created for ES CICSDEMO, process-id = 12308 15:42:31
251204 15423146 12844 CASC0112I SEP 00002 created for ES CICSDEMO, process-id = 12284 15:42:31
251204 15423155 12284 CICSDEMO CASS1500I SEP initialization started 15:42:31

```

### 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。

## 3.2 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・削除することができます。本チュートリアルでは定義済みのリソース定義テーブ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 へ移動し、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] を指定します。



- 2) 左下ペインの [リソースナビゲーション]>[PCT] 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。



The screenshot shows the 'リソースナビゲーション' (Resource Navigation) panel on the left with 'PCT' selected. The main area displays the 'アクティブ プログラム管理テーブル リソース' (Active Program Table Resources) with a list of entries. The 'ACCT' entry is selected, and its edit icon is highlighted.

- 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。



The screenshot shows the 'Program Management Table Entry - ACCT' screen. The '通用' (General) tab is selected. It displays fields such as TranClass (DFHTCLO0), Program Name (ACCT00), and PCT Type (ローカル). The '有効' (Effective) checkbox is checked, indicating the transaction is active.

- 4) 次に、左下ペインの [リソースナビゲーション]>[グループ別]>[SIT] をクリックすると、SIT 一覧が表示されますので、CICS 構成の SIT で指定した [DBCS] をクリックします。



The screenshot shows the 'リソースナビゲーション' (Resource Navigation) panel on the left with 'SIT' selected. Under 'SIT', the 'DBCS' entry is selected.

- 5) DBCS SIT には [始動リスト] として DBCS が、[SysID] として\$IVP 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。

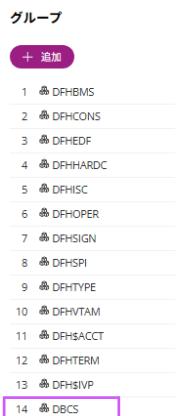


The screenshot shows the 'SIT' configuration screen for 'DBCS'. The '一般' (General) tab is selected. It shows the '説明' field containing 'DBCS Installation verification SIT'. The '始動リスト' (Start List) is set to 'DBCS', and the '開発用SIT' (Development SIT) checkbox is checked. The 'SysID' field is set to '\$IVP', and the '開始Tran ID' (Start Transaction ID) is set to 'CESN'. Other fields like 'WORK AREA (CWA)' and '最小COMMAREA' are also visible.

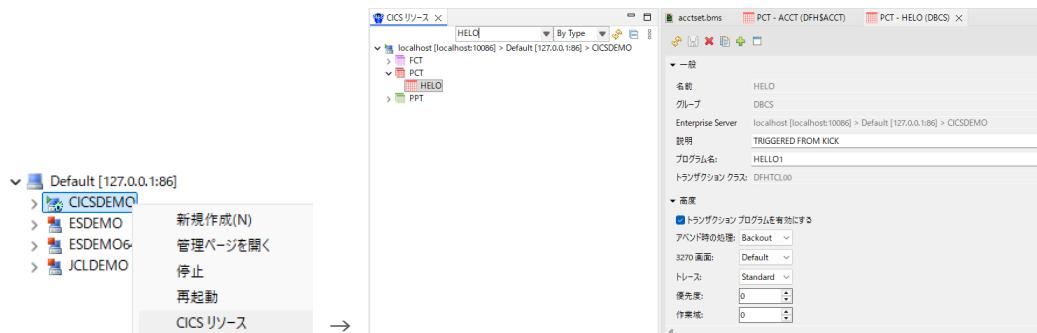
- 6) 上記で【始動リスト】に指定してある DBCS の内容を確認します。左下ペインの【リソースナビゲーション】>【グループ別】>【始動リスト】をクリックし、一覧から【DBCS】をクリックします。



- 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おり、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。



- 8) Enterprise Server インスタンスが開始状態であれば、Eclipse からも一部の CICS リソースをメンテナンスすることができます。



### 3.3 CICS の実行

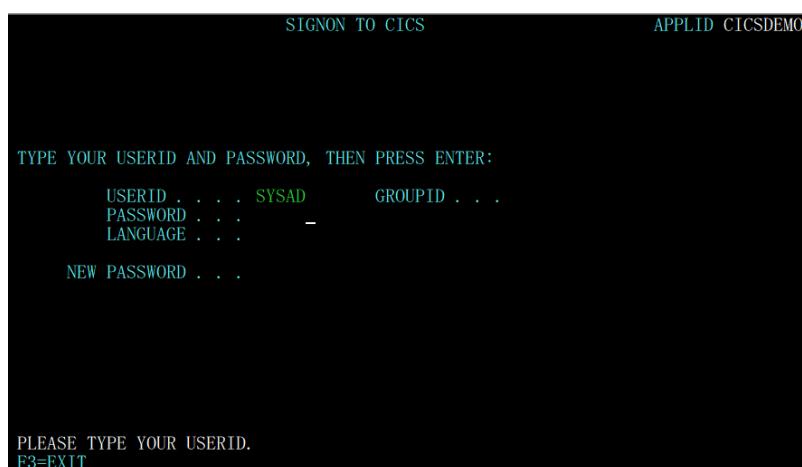
現在 CICSDEMO インスタンスが稼働しているので、例題プログラムを実行することができます。

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。



- 1) 下記は IBM パーソナル・コミュニケーションズを使用した画面です。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 サインオン画面が表示されます。  
[USERID]へ SYSAD を、[PASSWORD] へは前項で指定したパスワードを入力して実行キーを押します。



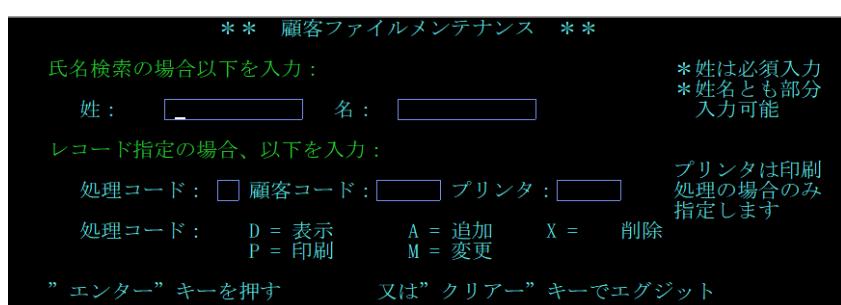
- 2) 正常にサインオン後、クリアキーで画面をクリアします。

```
CASSE0012I SIGNON COMPLETE AT B000, FOR USER SYSAD. LOCAL SECURITY IS DISABLED.
15:47:47
```

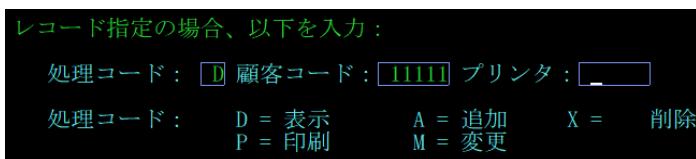
- 3) PCT に登録されていたトランザクションの ACCT を入力して実行キーを押します。

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。



- 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。



- 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。



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

### 3.4 CICS の動的デバッグ

Eclipse 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] ヘチェックと CICSDEMO インスタンスと Eclipse プロジェクトの関連付けが必要です。

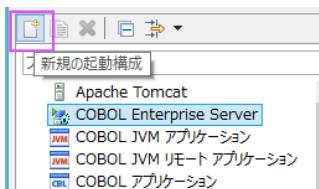
- 1) COBOL エクスプローラー内に存在する CICSDEMO プロジェクトの ACCT00.cbl をダブルクリックして内容を表示します。EXEC CICS 構文を使用して MAP の SEND と、次トランザクションを呼び出していることがわかります。

```
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。[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。



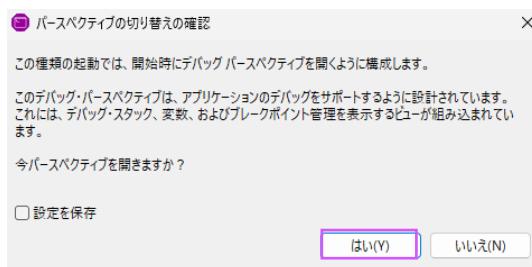
- 3) 左側のツリービューから [COBOL Enterprise Server] を選択して、左上の【新規の起動構成】アイコンをクリックします。



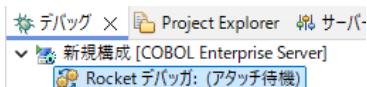
- 4) [COBOL プロジェクト] の [参照] ボタンから CICSDEMO を選択すると、関連する CICSDEMO インスタンスが表示されます。  
[デバッグの種類] は「CICS」を選択した状態で、[デバッグ] ボタンをクリックします。



- 5) パースペクティブの切り替え確認ウィンドウが表示されますので【はい】ボタンをクリックし、デバッグ用のパースペクティブを開きます。



- 6) 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。

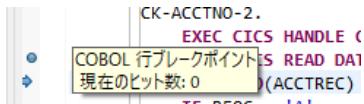


- 7) プログラムのステップ実行が可能になります。[F5] キーもしくは [実行] プルダウンメニューから [ステップイン] を選択してステップを進めることができ、変数タブでは使用している変数の値が確認できます。  
この例題プログラムでは TN3270 エミュレータとの画面送受信がありますので、その都度、表示を切り替えてデバッグします。

```
ACCT01.cbl x 变数 x ブレークポイント x プログラムアトライト x 式
名前 値 式部 GMRS 07717778
  A1CTC 111111 111111
  ACCTREC 111111
  ACCTDO 111111
  SNAMEDO 東
  FNAMEDO 武部
  MIDO G
  TILDO MRS
  TELDO 0771778888
  ADDR1DO 4 - 3. 石山
  ADDR2DO 大阪市
  AUTH1DO 遊堺
  AUTH2DO
  AUTH3DO
  AUTH4DO
  AUTH5DO
  CARDSDO 1
  IMDOD 06
  IDAYDO 07
  IYRDO 07
  RSNDO N
  CCODEDO X
  APPRDO GNG
  SCODE1DO
  SCODE2DO
  SCODE3DO
  STATDO N
  LIMITDO 1000.00
  PAY-HIST 0.00000000 0.00000000 0.00 0.00000000 0.00XX

名前 値
  SRCH-ANY
  IF LINE-CNT = 0, MOVE ? TO MSG-NO. 000221790
  MOVE ! TO SNAME1, GO TO MENU-RESEND. 000221800
  MOVE DFHBRYUP TO SURN1MA (1), SURN1MA (2), SURN1MA (3), 000221809
  SURN1MA (4), SURN1MA (5), SURN1MA (6). 000222008
  MOVE DFHBRYUP TO MSGNO, MOVE DFHBRYUP TO SURN1MA. 000222009
  EXEC CICS READ (PAP-FORM) MAPSET('ACCTSET') 000222009
  FREEK DATASET ERASEAND END-EXEC. 000222009
  IF LINE-CNT NOT > MAX-LINES. 000222009
  EXEC CICS RETURN TRANSID('ACB1') END-EXEC 000222708
  ELSE
    EXEC CICS RETURN TRANSID('ACB1') COMMAREA(SRCH-CTRL) 000223080
    LENGTH(44) END-EXEC. 000223090
  CK-ANY.
  IF IN-REQ = SPACES, MOVE ? TO SNAME1,
  MOVE 8 TO MSG-NO, GO TO MENU-RESEND. 000223480
  CK-ACCTNO.
  IF ACCTNO = SPACES, MOVE STARS TO ACCTDO,
  MOVE 5 TO MSG-NO, GO TO ACCT-ERR. 000223580
  IF (ACCTC > '10000' OR ACCTC > '99999' OR ACCTC NOT NUMERIC), 000223980
  MOVE 6 TO MSG-NO, GO TO ACCT-ERR. 000240000
  CK-ACCTC.
  EXEC CICS HANDLE CONDITION NOTIND( NO-ACCT-RECORD) END-EXEC 000240100
  EXEC CICS READ DATASET('ACCTFILE1') RIDFLD(ACCTC)
  INTO(ACCTREC) LENGTH(ACCTC-LNG) END-EXEC. 000243490
  IF REQC = 'A', 000244800
  MOVE 9 TO MSG-NO, GO TO ACCT-ERR, 000244800
  ELSE GO TO CK-REQ. 000244800
  NO-ACCT-RECORD.
  IF REQC = 'A', GO TO CK-REQ. 000249800
  MOVE 10 TO MSG-NO. 000250000
  ACCT-ERR.
  MOVE -1 TO ACCTHL, MOVE DFHBRYBC TO ACCTHA. 000251100
  CK-REQ.
  IF REQC = 'D' OR 'P' OR 'A' OR 'M' OR 'X',
  IF MSG-NO = 0 GO TO CK-USE, ELSE GO TO MENU-RESEND. 000251500
  IF REQC = SPACE, MOVE STARS TO REQNO. 000251500
```

- 8) 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。



- 9) 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。



- 10) 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。



- 11) TN3270 エミュレータを切断します。

- 12) デバッグ履歴を消去するには、[デバッグ] パースペクティブの [デバッグ] タブ内で右クリックし、[終了したエントリをすべて削除] を選択します。



- 13) 有上の COBOL パースペクティブアイコンをクリックして、COBOL パースペクティブに戻ります。



### 3.5 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、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すべてを対象として、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。