

Enterprise Developer チュートリアル

リモート メインフレーム COBOL 開発 : CICS システム間通信 Eclipse 編

1. 目的

本チュートリアルは Linux 上に COBOL プロジェクトと CICS 用 Enterprise Server インスタンスを複数作成し、SysC を介してインスタンス間通信を行う手順の習得を目的としています。また、ローカル Windows マシンから、リモート Linux マシンへ、Eclipse のリモート メインフレーム COBOL プロジェクトを使用して実施します。

2. 前提

- 本チュートリアルで使用したリモートマシン OS : Red Hat Enterprise Linux 9.4
- 本チュートリアルで使用したローカルマシン OS : Windows 11 Pro
- リモートマシンに Enterprise Developer 11J PU2 for Linux and Unix をインストール
- ローカルマシンに Enterprise Developer 11J PU2 for Eclipse をインストール
- 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
- メインフレーム COBOL 開発:CICS チュートリアルを終了していること
- リモートメインフレーム COBOL 開発:JCL チュートリアルを終了していること

3. 実施するシナリオ

Linux マシン上に CICS 機能を利用する COBOL プロジェクトを 2 つ作成し、それぞれ実行させる Enterprise Server インスタンスを作成します。1つ目を CICS1 インスタンス、2 つ目を CICS2 インスタンスと称し、CICS1 インスタンスから CICS2 インスタンスへ通信を介したルーティングを行う下記 4 つのシナリオを実施します。

- A) CICS1 インスタンスへログインして、全てのトランザクションを CICS2 インスタンスで実行させる。
- B) CICS1 インスタンスへログインして、特定トランザクションだけを CICS2 インスタンスで実行させる。
- C) CICS1 インスタンスへログインして、トランザクションから出力するデータを CICS2 インスタンスに存在するファイルへ出力する。
- D) CICS1 インスタンスへログインして、トランザクションのプログラムから CICS2 インスタンスに存在するプログラムへリンクさせる。

4. チュートリアル手順の概要

1. リモートマシンの準備
2. チュートリアルの準備
3. Eclipse の起動
4. リモート メインフレーム COBOL プロジェクトの作成
5. プロジェクトプロパティの設定
6. ビルドの実行
7. Enterprise Server インスタンスの設定
8. Enterprise Server インスタンスの開始と確認
9. シナリオ A に対するインスタンスの準備
10. シナリオ A に対するルーティングの実施
11. シナリオ B に対するインスタンスの準備
12. シナリオ B に対するルーティングの実施
13. シナリオ C に対するインスタンスの準備
14. シナリオ C に対するルーティングの実施
15. シナリオ D に対するコードと実施方法の紹介
16. Enterprise Server インスタンスの停止
17. リモートマシンの切断

4.1 リモートマシンの準備

リモートマシンの準備を行うために、リモートマシンヘルートユーザーでログインします。

- 1) 環境変数 LANG に SJIS ロケールを設定します。

コマンド例) export LANG=ja_JP.sjis

```
#export LANG=ja_JP.sjis
```

- 2) COBOL を実行する環境を設定します。製品フォルダ配下の bin フォルダ内に存在する cobsetenv を実行すると、環境変数の COBDIR が COBOL 環境として設定された旨のメッセージが表示されます。

コマンド例) ./opt/mf/ED11PU2/bin/cobsetenv

```
#. /opt/mf/ED11PU2/bin/cobsetenv  
COBDIR set to /opt/mf/ED11PU2
```

- 3) COBOL 作業モードを設定します。

COBOL の作業モード(32-bit または 64-bit)を指定します。cobmode コマンドまたは環境変数 COBMODE を使用して設定します。

64-bit 設定コマンド例) export COBMODE=64

```
#export COBMODE=64
```

- 4) 後述の startrdodaemon コマンドを使用する場合は JAVA_HOME 環境変数を設定します。

- 5) Windows、Linux 両マシンで稼働する共通管理画面の Enterprise Server Common Web Administration(以降 ESCWA)では、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します。

Windows マシン)

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。



Windows マシンで 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 Windows マシンで稼働する 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。

Linux マシン)

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
```

```
{"mfUser": "SYSAD", "mfPassword": "hi_wvpvJ"}
```

上記例の場合、デフォルトユーザーは SYSAD、初期パスワードは hi_wvvPJ です。

この情報は Linux マシンで稼働する ESCWA 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。

- 6) Linux マシンの ESCWA を、ループバックモードをオフにして起動します。

コマンド)

```
cd $COBDIR/bin
```

```
nohup escwa --BasicConfig.MfRequestedEndpoint="tcp:/:10086" --write=true < /dev/null > escwa.out 2>&1 &
```

```
#nohup escwa --BasicConfig.MfRequestedEndpoint="tcp:/:10086" --write=true < /dev/null > escwa.out 2>&1 &
[3] 17254
```

- 7) Enterprise Server インスタンスを運用、管理する Micro Focus Directory Server(以降 MFDS と称す)を起動します。

Linux マシンに作成された Enterprise Server インスタンスを制御する MFDS を、mfds コマンドを使用して起動します。32-bit 環境用には mfd32 コマンド、64-bit 環境用には mfd64 コマンドを明示的に実行することも可能です。

コマンド例) mfd &

上記 “&” を付加すると、設定済の COBOL 環境変数を基に別プロセスで mfd が起動されます。

```
#mfds &
[1] 15850
```

- 8) Windows マシンからのアクセス方法を RSE に指定する場合は接続ポートの解放を行います。本チュートリアルでは SSH 接続を使用しますが、この場合はポートの解放は必要ありません。

COBOL 環境の配下に存在する startrdodamn を実行します。

コマンド例) \$COBDIR/remotedev/startrdodamn 5000

上記 5000 をポート番号へ指定しない場合には、デフォルトの 4075 がポート番号として指定されます。

```
$COBDIR/remotedev/startrdodamn 5000
Starting RSE daemon...

Reading "/opt/mf/ED11PU2/remotedev/rdo.cfg" ...

Server port range loaded from "/opt/mf/ED11PU2/remotedev/rdo.cfg": 10000-10003
#Daemon running on: RHELKTJP, port: 5000
```



注意

ディレクトリ配下の書き込み権限がない場合はビルド時にエラーとなります。

ls -l コマンドなどで権限を確認後、chmod コマンドなどで適切な権限設定を前もって実行してください。

4.2 ローカルマシンの準備

例題プログラムに関するリソースを Windows マシンに用意します。

- 1) 使用する例題プログラムは、CICS チュートリアル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。

4.3 Eclipse の起動

- 1) Windows マシンで 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 COBOL Perspective] をクリックして、COBOL パースペクティブを開きます。



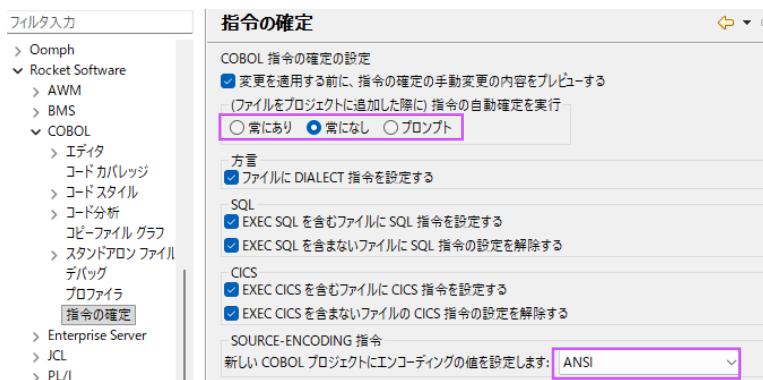
Enterprise Developer for Eclipse によるこそ



- 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。

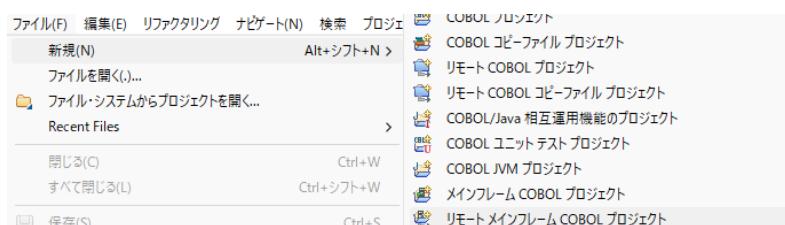


- 5) 既存ファイルのインポート時、自動的にコンパイル指令が指定される機能が用意されていますが、本チュートリアルではこれを解除します。[ウィンドウ] プルダウンメニューの [設定] > [Rocket Software] > [COBOL] > [指令の確定] > [COBOL 指令の確定の設定] では [常なし] を選択し、[SOURCE-ENCODING 指令] では [ANSI] を選択後、[適用して閉じる] ボタンをクリックします。



4.4 リモート メインフレーム COBOL プロジェクトの作成

- 1) 新しい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] > [リモート メインフレーム COBOL プロジェクト] を選択します。

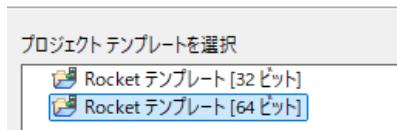


- 2) プロジェクト作成ウィンドウには以下のように入力します。

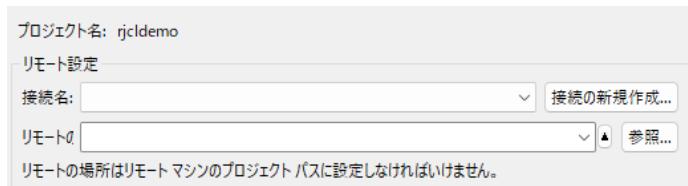
| 項目名 | 説明 |
|--------------|---|
| プロジェクト名 | 任意です。ここでは CICS1 を指定します。 |
| ファイル システムを選択 | リモートマシンと接続するファイル システムを指定します。ここでは「セキュアシェル(SSH)ファイル システム」を選択します。RSE 接続の場合は「リモートファイルシステム (RSE)」を選択します。 |



- 3) テンプレート指定ウィンドウでは、[Rocket テンプレート 64 ビット] を選択して [次へ] ボタンをクリックします。



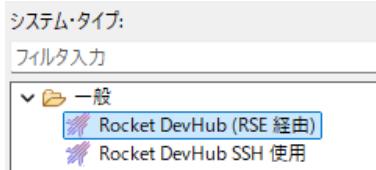
- 4) 新しい接続を作成するため、[接続の新規作成] ボタンをクリックします。



- 5) 接続タイプでは2種類から選択可能です。

5-1) RSE 指定の場合

- ① [RSE 経由] を選択して [次へ] ボタンをクリックします。



- ② [ホスト名] へ Linux マシンのホスト名または IP アドレスを指定して [次へ] ボタンをクリックします。

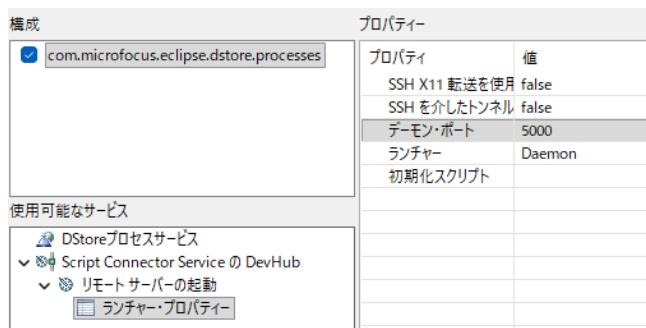
[接続名] は任意に変更可能です。



- ③ 下記画面の [使用可能なサービス] > [Script Connector Service の DevHub] > [リモート サーバーの起動] > [ランチャー・プロパティー] > [デーモン・ポート] 項目値を、前項で指定した Linux マシンのポート 5000 へ変更後、[終了] ボタンをクリックします。デフォルトポート番号(4075)を解放した場合は前画面で [終了] ボタンをクリックして構いません。

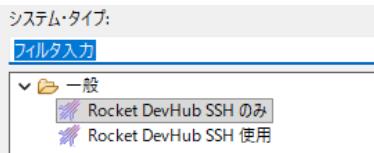
デフォルト値)4075

変更値)5000



5-2) SSH 指定の場合

- ① [SSH のみ] を選択して [次へ] ボタンをクリックします。



- ② [ホスト名] へ Linux マシンのホスト名または IP アドレスを指定して [次へ] ボタンをクリックします。

[接続名] は任意に変更可能です。



- ③ 下記画面の [使用可能なサービス] > [DStore Connector Service] > [リモート サーバーの起動] > [ランチャー・プロパティー] > [サーバー起動コマンド] 項目値を製品のインストールパスへ修正後、[終了] ボタンをクリックします。

変更前の値)

/opt/rocketsoftware/EnterpriseDeveloper

変更後の値例)

/opt/mf/ED11PU2

DevHub SSH アクセス

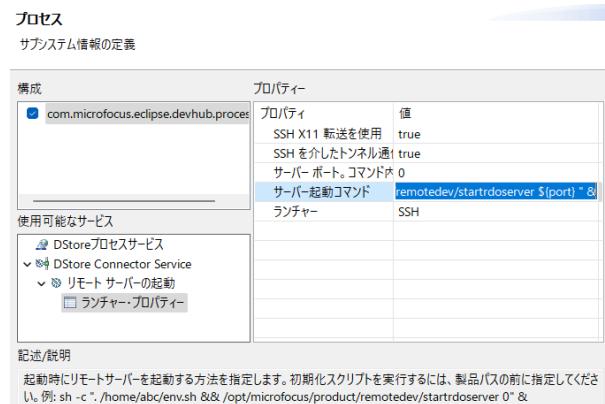
Rocket DevHub のインストール場所の定義

Rocket DevHub インストール ディレクトリの場所はフルパスである必要があります。
これは 'DevHub Ssh Access' サブシステムのプロパティを使用して後で変更できます。

DevHub の場所: /opt/mf/ED11PU2

変更後の値例)

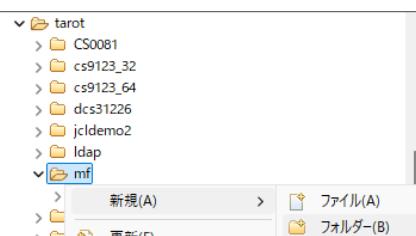
```
sh -c "/opt/mf/ED10GA/remotedev/startrdoserver ${port} " &
```



- 6) Linux マシンにプロジェクトを作成するロケーションを指定するため、[参照] ボタンをクリックします。Linux マシンへのログオンウィンドウが表示ますので、権限を持つユーザーID とパスワードを指定してアクセスしてください。



- 7) Linux マシンのブラウザウィンドウが表示されますので、配置したいパスへプロジェクト用のディレクトリを作成します。作成可能なロケーションを右クリックして [新規] > [フォルダー] を選択します。



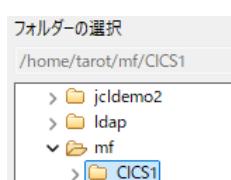
- 8) 新しいディレクトリ名は任意ですが、ここではプロジェクト名と同様の CICS1 を指定して [終了] ボタンをクリックします。

リモート・フォルダー

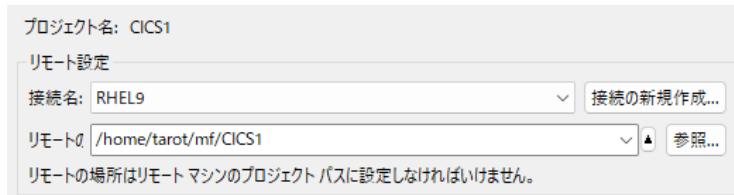
新規フォルダーの作成

| | |
|--------------|----------------|
| 接続名(A) : | RHEL9 |
| 親フォルダー(C) : | /home/tarot/mf |
| 新規フォルダ名(D) : | <u>CICS1</u> |

- 9) プロジェクトディレクトリへ CICS1 を選択して、[OK] ボタンをクリックします。



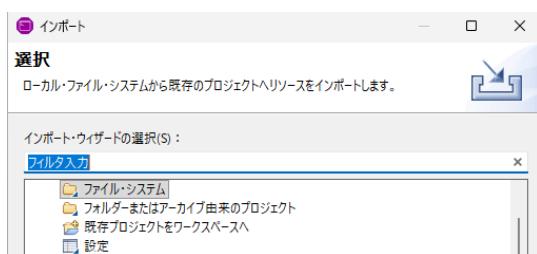
- 10) 指定項目を確認後、[終了] ボタンをクリックします。



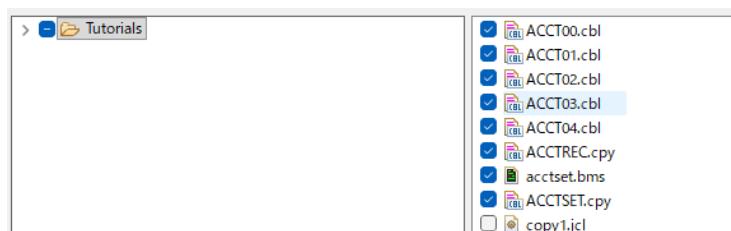
- 11) 同様の操作で、CICS2 プロジェクトも作成してください。COBOL エクスプローラーに 2 つのリモートプロジェクトが表示されます。



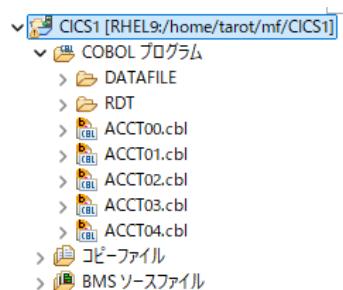
- 12) 用意した例題プログラム類をインポートします。CICS1 プロジェクトを右クリックして [インポート] > [インポート] を選択し、インポートウィンドウにて [一般] > [ファイル・システム] を選択後 [次へ] ボタンをクリックします。



- 13) 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、左側の [Tutorials] ハエックし、不要な下から 4 ファイルのチェックをはずします。ファイル名先頭に [ACCT] が付く上部 8 ファイルが残りますので [終了] ボタンをクリックします。この実行により、リモートマシンのプロジェクトディレクトリへ例題プログラムが配置されます。



- 14) COBOL エクスプローラー内に表示されている CICS1 プロジェクトにインポートしたファイルやフォルダが表示されていることを確認します。



- 15) 同じ手順で CICS2 プロジェクトへ例題プログラム類をインポートします。

4.5 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 命令を含むプログラム、コピー文が含まれています。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 CICS1、CICS2 各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。

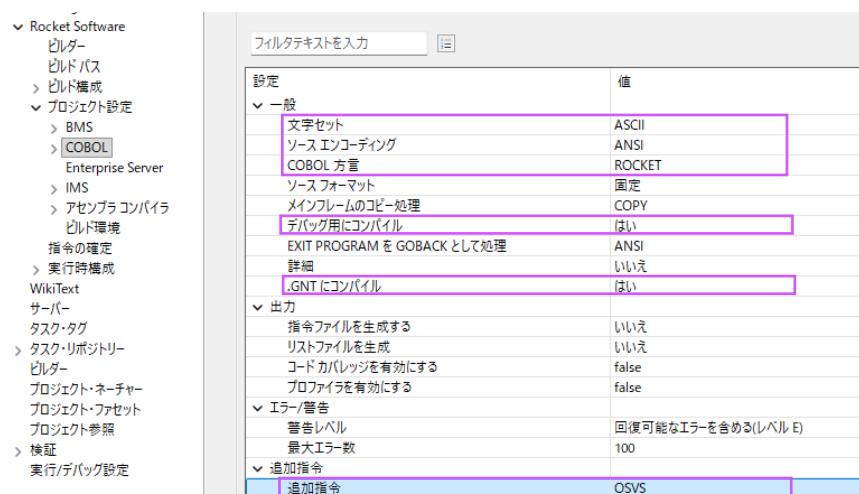
2) 左側ツリービューの [Rocket Software] > [ビルト構成] > [リンク] を選択して、下記項目を指定します。指定後は [適用] ボタンをクリックしてください。

| 項目名 | 説明 |
|----------|--|
| ターゲットの種類 | 実行ファイル形式を指定。ここでは [全て 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] を指定します。 |

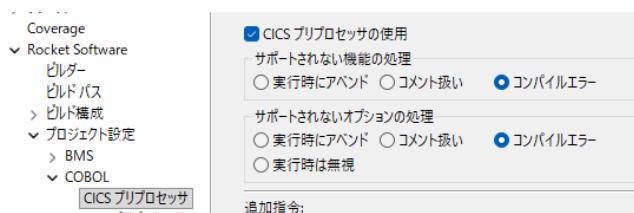


3) 左側ツリービューの [Rocket Software] > [プロジェクト設定] > [COBOL] を選択して、下記項目を指定します。指定後は [適用] ボタンをクリックしてください。

| 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 | サンプルソースは SJIS を使用しているため、ANSI を指定します。 |
| 言語の方言 | 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 に指定します。 |
| 追加指令 | ここでは OSVS を入力します。 |



- 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て [適用して閉じる] ボタンをクリックするとプロパティウィンドウが閉じます。

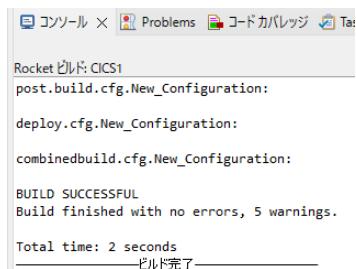


4.6 ビルドの実行

- 1) COBOLエクスプローラー内のCICS1 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。CICS2 プロジェクトも同様にビルドします。



- 2) コンソールタブで成功を確認します。BMS ファイルもリモートマシン上でビルドされ、MOD ファイルが出力されます。

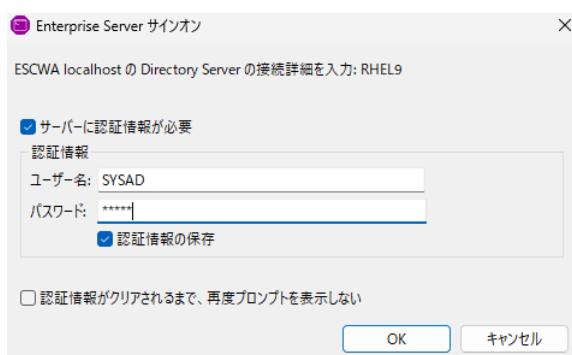


- 3) COBOL エクスプローラーのプロジェクト内に存在する New_Configuration.bin フォルダ配下に実行ファイル (.gnt ファイル) が作成されていることを確認してください。

4.7 Enterprise Server インスタンスの設定

- 1) Linux マシンで稼働する Enterprise Server インスタンスを作成します。

Windows マシンの Eclipse で [サーバー エクスプローラー] タブを開き、前項で作成した Linux マシンへの接続である [RHEL9] をクリックすると認証ウィンドウが表示されますので、取得したデフォルトユーザーと初期パスワードを指定してサインオンします。



- 2) [RHEL9] を右クリックして [管理ページを開く] を選択します。
- 3) 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 Linux マシンの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。

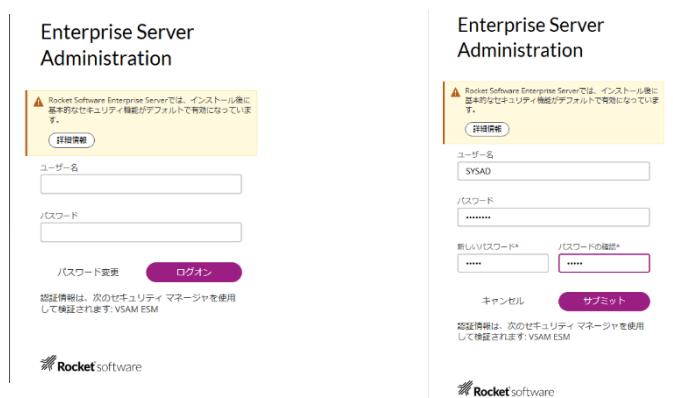
これを回避するために、新パスワードの英字はすべて大文字で入力してください。

良い新パスワードの例) SYSAD123

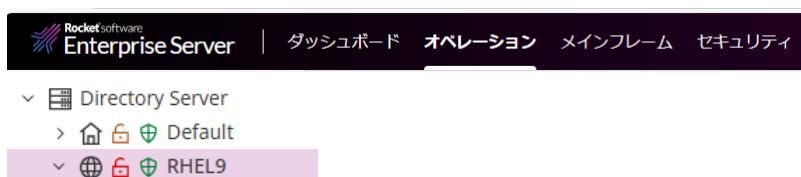
悪い新パスワードの例) sysad123

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。

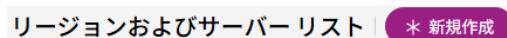
パスワード変更後、ESCPWA がタイムアウトした場合は新しいパスワードを使用してログオンしてください。



- 4) メニューで [オペレーション] を選択後、[ナビゲーション] で [RHEL9] をクリックします。



- 5) 右側ペインの画面中央にある [新規作成] ボタンをクリックして Linux マシンで稼働する Enterprise Server インスタンスを作成します。



- 6) 表示された画面の [名前] に CICS1 を入力します。64 ビットの実行可能ファイルを生成したため [64 ビット作業モード] にチェックし、[TN3270 リスナーの作成] がチェックされていることを確認後、ポート番号に 9004 を入力します。最後に [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* | <input type="text" value="CICS1"/> |
| 説明 | <input type="text"/> |
| <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 | <input type="text" value="9004"/> ^ ↓ |
| <small>* 入力必須の項目です</small> | |
| <small>保存 戻る</small> | |

- 7) 作成した CICS1 インスタンスが一覧に表示されます。CICS1 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。

| ア... | 名前 | タイプ | ステータス | 64ビット | MSS有効 | セキュ... | アクション |
|------|-------|--------|---------|-------|-------|--------|---|
| | CICS1 | Region | Stopped | ✓ | ✓ | デフォルト |    |


重要

実行ファイル生成に指定した稼働ビット数 = Enterprise Server インスタンス稼働ビット数である必要があります。

- 8) [リージョンの機能] の [MSS 有効] がチェックされていることを確認します。

リージョンの機能

| | |
|---|--------------------------------|
| <input checked="" type="checkbox"/> MSS有効 | <input type="checkbox"/> JES有効 |
| <input type="checkbox"/> IMS有効 | <input type="checkbox"/> MQ有効 |

- 9) 表示画面の下にある [動的デバッグを許可] にチェックします。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 | |

- 10) [追加設定] の [構成情報] 欄へ、日本語半角カナを有効にするため下記の環境変数を設定し、最後に [適用] ボタンをクリックします。

入力値)

[ES-Environment]
MFCODESET=9122

追加設定

構成情報

[ES-Environment]
MFCODESET=9122


重要

入力値は全て半角英数字で指定してください。

- 11) 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。 |

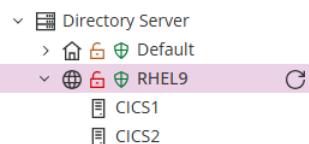
 **重要**

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。

- 12) 同様の手順で CICS2 インスタンスを作成します。CICS1 インスタンスと異なる TN3270 ポート番号を指定してください。
- 13) 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。

- 14) 各リスナーの [tcp] にはリモートマシンのホスト名または IP アドレスを入力して [適用] ボタンをクリックします。

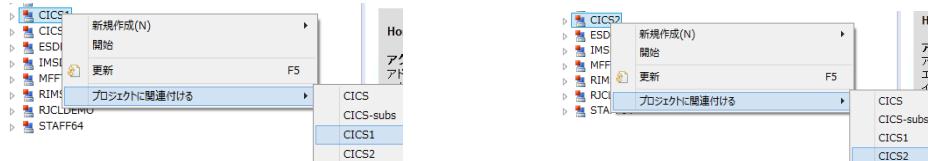
- 15) 2つのインスタンス作成が終了したら、画面左側ペインの [RHEL9] をクリックして一覧画面に戻ります。



4.8 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リモート環境に CICS1 と CICS2 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リモート環境を右クリックし、[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ー内のリモート環境に存在する CICS1 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] > [CICS1] を選択します。これにより CICS1 プロジェクトから実行されるアプリケーションは CICS1 インスタンスで処理されることになります。

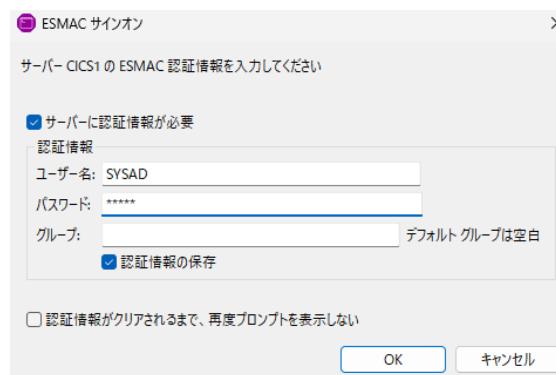
CICS2 インスタンスも同様に CICS2 プロジェクトへ関連付けます。



- 3) CICS1 インスタンスと CICS2 インスタンスをそれぞれ右クリックして [開始] を選択します。



- 4) ESMAC サインオンウィンドウが表示されます。開始権限を持つ SYSAD ユーザーと変更したパスワードを指定します。



- 5) CICS2 インスタンスの SysID をユニークな値に変更します。ESCWA に移動し、CICS2 インスタンスの開始を確認後、[CICS] > [グループ別] を選択します。



- 6) [リソースナビゲーション] で [SIT] > [DBCS] を選択します。



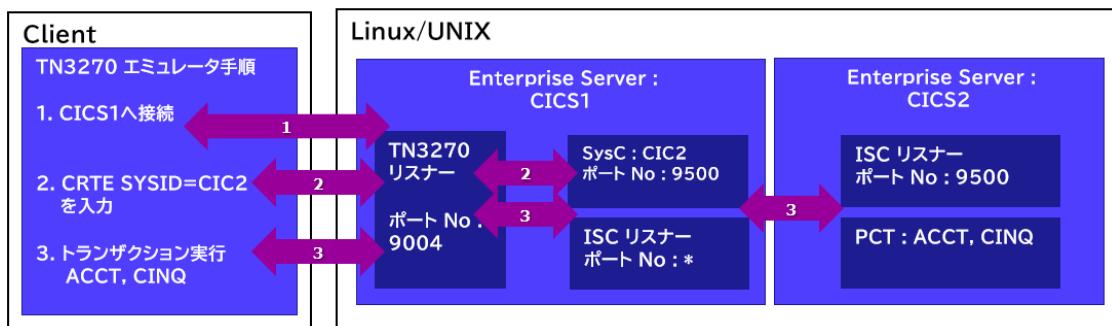
- 7) 右側ペインの [SysID] を\$IVP から CIC2 に変更して [適用] ボタンをクリックします。



4.9 シナリオ A に対するインスタンスの準備

シナリオ) CICS1 インスタンスへログインして、全てのトランザクションを CICS2 インスタンスで実行させる。

【概要図】



- 1) CICS1 インスタンスへ CICS 間通信に必要な ISC リスナー定義を作成します。ESCWA から CICS1 インスタンスの編集画面を開き、[一般] プルダウンメニューから [リスナー] を選択し、[リスナーの新規作成] ボタンをクリックします。



- 2) 下記内容を入力後、[保存] ボタンをクリックすると、自動的にリスナーが開始されます。

| 項目名 | 説明 |
|---------|--|
| 名前 | 任意です。ここでは ISC1 を指定しますが、他インスタンスと重ならないように指定してください。 |
| IP アドレス | ホスト名または IP アドレスもしくは*を指定します。 |
| ポート | 使用していないポート番号もしくは*を指定します。 |
| 会話タイプ | [MSS システム間通信] を選択します。 |

リスナー プロパティ | **保存**

* 入力必須の項目です

名前* レガシー Micro Focus アプリケーション形式

説明

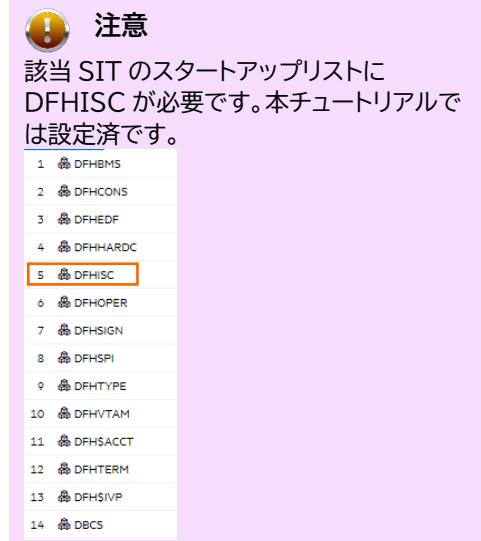
□ このエンドポイントはネットワーク経由でアクセス可能になり、TLSが無効になります。

プロトコル ホスト名またはIP アドレス* ポート *

TLS設定

⚠ 合法タイプを変更したため、高度なプロパティの値がリセットされました。この変更を適用すると、変更は確定されます。

合法タイプ WebサービスとJ2EE
 SOAPとJ2EE (レガシー)
 Web
 Fileshare
 TN3270
 DCAS
 IMS Connect
 RFA
 MSSシステム間通信
 MSSシステム間通信 (インバウンドのみ)



- 3) 作成後、再度 ISC リスナーを開き、ステータスを [Stopped] から [Started] に変更して [適用] ボタンをクリックすると自動的に開始されます。

ステータスの設定

- 4) CICS1 インスタンスに CICS 間通信に必要な SysC 定義を作成します。[CICS] プルダウンメニューから [CICS] > [リソース] > [タイプ別] を選択し、左側ペインで [SysC] をクリック後、[新規作成] ボタンをクリックします。

CCI システムの接続 リソース | *** 新規作成**

- 5) 下記項目を入力し [保存] ボタンをクリックします。

| 項目名 | 説明 |
|---------|--|
| 名前 | 4文字で指定します。任意ですが通信相手先が認識可能な名前を指定します。ここでは CIC2 を指定します。この名前がリモートトランザクションの SYSID となります (下記画面参照)。 |
| グループ | CICS1 インスタンスで指定している SIT を指定します。ここでは DBCS を選択します。 |
| 説明 | 説明を任意で入力します。 |
| ノード | 通信相手インスタンスが存在するアドレスを指定します。この場合は CICS2 インスタンスが存在する Linux/UNIX マシンの IP アドレスです。(例:11.22.33.44) |
| ポート | CICS 間通信に使用するポート番号を指定します。他で使用していない 4 行の数値を指定してください。ここでは 9500 を指定します。 |
| ネット名 | 通信相手インスタンス名を指定します。ここでは CICS2 です。 |
| セッション最大 | 1 以上の数値を設定します。ここでは 4 を指定します。 |

CCI システムの接続 リソースの作成 | **保存**

一般

| | | |
|------|------|------|
| 名前* | 名前 | グループ |
| CIC2 | DBCS | |
| 説明 | | |

接続パラメーター

| | | |
|--------|-------|---------|
| プロトコル | ネット名 | セッション最大 |
| TCP/IP | CICS2 | 4 |
| ノード* | ポート | |
| RHEL9 | 9500 | |

- 6) CICS2 インスタンスへ ISC リスナー定義を作成します。CICS1 インスタンスと同様の手順で、CICS1 インスタンスの SysC 定義で指定した [ポート] 番号を指定します。リスナー追加画面で下記項目を入力後、[保存] ボタンをクリックします。

| 項目名 | 説明 |
|-------------|--|
| 名前 | ここでは ISC2 を指定します。 |
| IP アドレス、ポート | CICS1 インスタンスが存在する IP アドレスと CICS1 インスタンスの SysC 定義で指定した [ポート] 番号を指定します。同じ IP アドレス内であれば “*:9500” が指定可能です。(例:11.22.33.44:9500) |
| 会話タイプ | [MSS システム間通信] を選択します。 |

リスナー プロパティ | **通用** | **削除**

* 入力必須の項目です

| | | |
|--|-----------------|-------------------------------------|
| 名前* | 説明 | レガシー Micro Focus アプリケーション形式 |
| ISC2 | | <input checked="" type="checkbox"/> |
| このエンドポイントはネットワーク経由でアクセス可能で、TLSが有効ではありません。 | | |
| プロトコル | ホスト名またはIP アドレス* | ポート |
| ccitcp | RHEL9 | 9500 |
| <input type="button" value="+"/> TLS設定 | | |
| ステータス | ステータスの設定 | 実際のアドレス |
| Stopped | Started | ccitcp:0.0.0.0:0 |
| 会話タイプ | | |
| <input type="radio"/> WebサービスとJ2EE <input type="radio"/> SOAPとJ2EE (レガシー) <input type="radio"/> Web <input type="radio"/> Fileshare <input type="radio"/> TN3270 <input type="radio"/> DCAS <input type="radio"/> IMS Connect <input type="radio"/> RFA <input checked="" type="radio"/> MSS システム間通信 | | |

- 7) CICS1 と CICS2 インスタンスを再起動します。

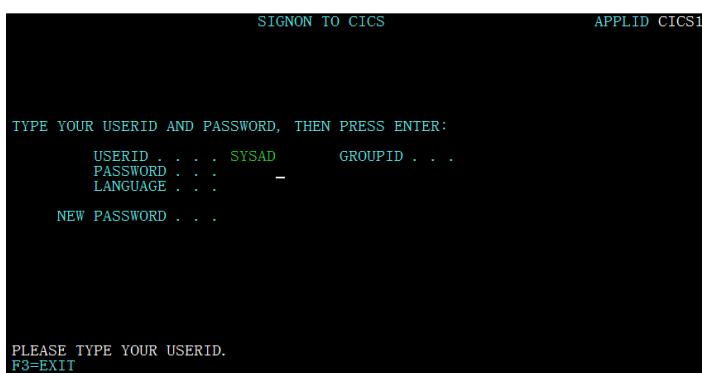
CICS1 インスタンスのコンソールログで SysC 接続を確認します。下記内容が表示されれば認識されています。

| | |
|------------|--|
| CASCS1122I | Attempting to establish ISC connection CIC2 to system CICS2 |
| CASCS1109I | Connection CICS2 (sysid CIC2) is activated contention winner |

4.10 シナリオ A に対するルーティングの実施

シナリオ)CICS1 インスタンスへログインして、全てのトランザクションを CICS2 インスタンスで実行させる。

- 1) TN3270 エミュレータを CICS1 インスタンスの TN3270 リスナーポート(9004)へ接続します。

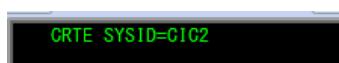


- 2) [USERID] に SYSAD を、[PASSWORD] に設定したパスワードを入力してログインします。

- 3) 画面クリア後に下記コマンドを実行します。

補足)TN3270 エミュレータで、使用しているキーボード設定のクリア設定をご確認ください。

CRTE SYSID=CIC2



- 4) ルーティングが開始されました。

DFH4409 THE ROUTING SESSION TO SYSTEM CIC2 HAS BEEN STARTED.

- 5) 確認のため、CINQトランを実行します。

セキュリティは Linux マシンの ESCWA で管理されています。トランの実行が拒否される場合は、起動済の Linux マシンの ESCWA(IP アドレス:10086)へ接続し、ESCUWA のセキュリティから [リソース] > [TCICSTRN] > [CINQ] でSYSADユーザーが実行できる権限を与えてください。

VSAM ESMセキュリティ)

デフォルト値)ALLOW:OPERATOR GROUP:read;ALLOW:SYSADM GROUP:read;DENY:execute

変更例)ALLOW:*:read

トラン)CINQ

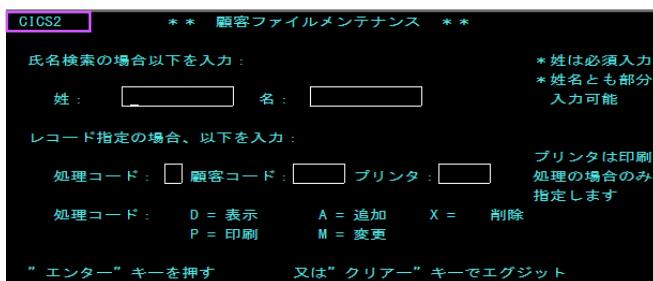
CICS2 インスタンスへルーティングされていることが確認できます。

```
INVALID-REQUEST (IVRQ ...) ABCODE(____) ABDUMP(____) ABPROGRAM(____)
ALTSCRNHT(000) ALTSRNRWD(000) APLKYBD(____) APLTEXT(____) APPLID(CICS2____)
ASRAINTRT(LOWVALUE) ASRAPSW(LOWVALUE) ASRAREGS(LOWVALUE) BTRANS(____) CMDSEC(____)
COLOR(Y) CWALENG(00512) DEFSCRNHT(024) DEFSRNRWD(080) DELIMITER(X'00')
DESTCOUNT(IVRQ...) DESTID(IVRQ...) DESTIDLENG(IVRQ...) DS3270(____) DSSCS(____)
EWASUPP(Y) EXTD(Y) FACILITY(0000) FCI(X'01') GCHARS(00000) GCODES(00000)
GM1(____) HIGHLIGHT(Y) INITPARM(LOWVALUE) INITPARMLEN(000) INPARTN(____) KATAKANA(____)
LDCMNEM(IVRQ...) LDGNUM(____) MAPCOLUMN(IVR) MAPHEIGHT(IVR) MAPLINE(IVR)
MAPWIDTH(IVR) MSCRCONTROL(____) NATLANGINUSE(E) NETNAME(NETA0000) NEXTTRANSID(____)
NUMTAB(____) OPCCLASS(X'000000') OPERKEYS(X'0000000000000001') OPID(X'535953')
OPSECURITY(X'000001') ORGACODE(____) OUTLINE(Y) PAGENUM(IV) PARTNPAGE(IV)
PARTNS(____) PARTNSET(____) PRINSYSID(____) PROGRAM(DFHZG1Q) PS(Y) QNAME(IVRQ)
RESSEG(____) RESTART(____) SCRNRHT(024) SCRNRWD(080) SIGDATA(00000) SOSI(Y)
STARTCODE(TD) STATIONID(____) SYSID(CS2) TASKPRIORITY(255) TCTUALENG(000)
TELLERID(____) TERMPCODE(X'9132') TERMPRIORITY(000) TEXTKYBD(____) TEXTPRINT(____)
TRANSPRIORITY(000) TWALENG(0000) UNATTEND(____) USERID(CICSUSER)
USERNAME(____) USERPRIORITY(255) VALIDATION(____)
```

- 6) 画面クリア後に下記トランザクションを実行します。確認するために画面左上にインスタンス名を表示しています。同じように出力するためには BMS ファイルに項目の追加が必要になります。

トランザクション)ACCT

CICS2 インスタンスで指定トランザクションが起動されています。

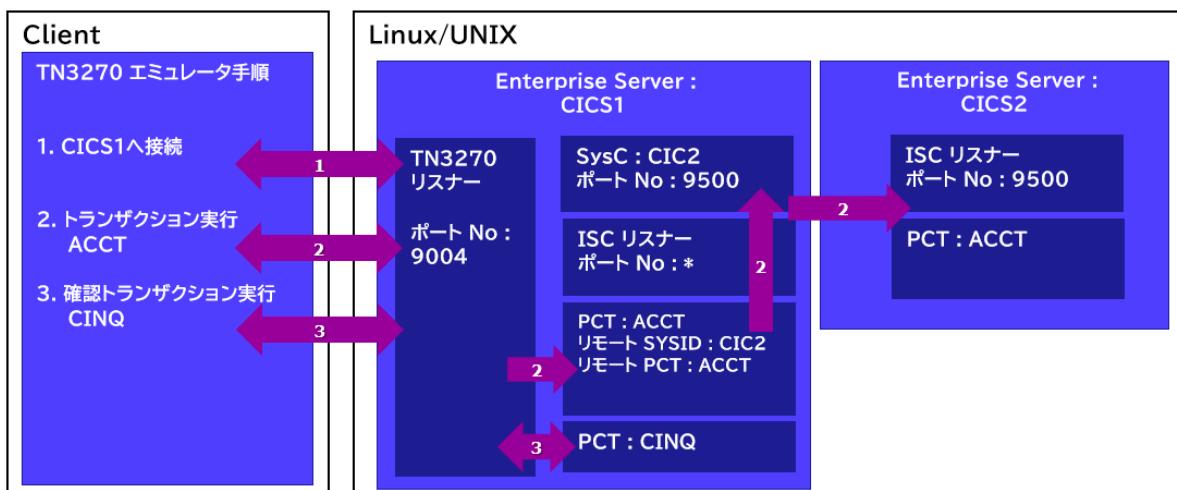


- 7) 確認後は接続を切斷してください。

4.11 シナリオ B に対するインスタンスの準備

シナリオ)CICS1 インスタンスへログインして、特定トランザクションだけを CICS2 インスタンスで実行させる。

【概要図】



- 1) CICS1インスタンスへPCT定義を作成します。CICS1インスタンスを【編集】アイコンで開き、画面上部の【モニター】プルダウンメニューから【CICS】>【リソース】>【タイプ別】を選択し、左側ペインの【PCT】をクリックします。

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、【コピー】ボタンをクリックします。

プログラム管理テーブル - ACCT (DFH\$ACCT) |

- 2) 下記を変更して【保存】ボタンをクリックします。

| 項目名 | 説明 |
|------------|--|
| 名前 | ACCT を入力します。 |
| グループ | SIT で指定している DBCS を選択します。 |
| リモート SYSID | SysC 定義の名前である CIC2 を入力します。 |
| リモート PCT | CICS2 インスタンスで実行する PCT 定義名を入力します。ここでは既存の同名 PCT を実行しますので ACCT を入力してください。 |

名前* グループ

説明

TranClass

プログラム名 リモート SysID リモート PCT

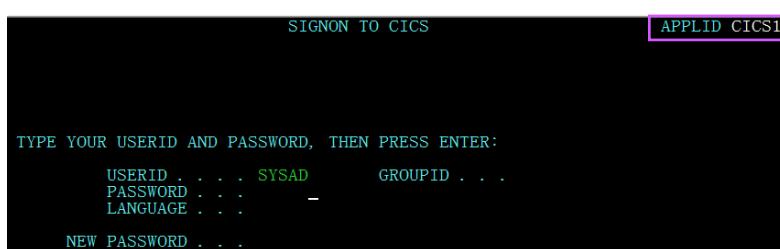
- 3) 追加を反映させるために【インストール】ボタンをクリックします。

プログラム管理テーブル - ACCT (DBCS) |

4.12 シナリオ B に対するルーティングの実施

シナリオ) CICS1 インスタンスへログインして、特定トランザクションだけを CICS2 インスタンスで実行させる。

- 1) リモートとして作成した PCT 定義の ACCT だけを CICS2 インスタンスへルーティングします。TN3270 エミュレータを CICS1 インスタンスへ接続します。



- 2) [USERID] に SYSAD を、[PASSWORD] に設定したパスワードを入力してログインします。

- 3) 画面クリア後に下記トランザクションを実行します。この例では画面左上にインスタンス名を表示しています。

トランザクション)ACCT

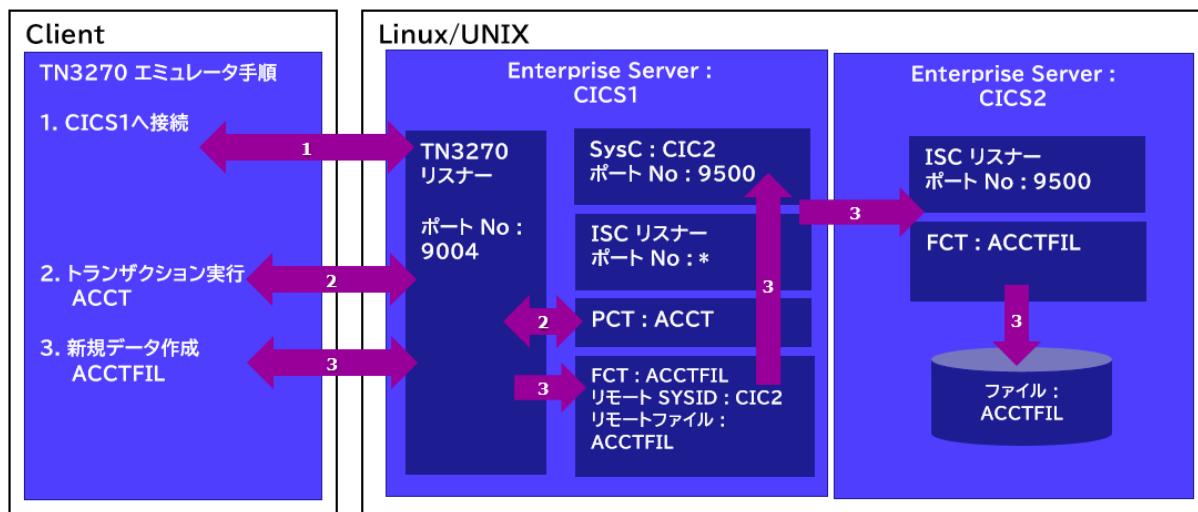
CICS2 インスタンスで指定トランザクションが起動されています。

- 4) 確認後は接続を切斷してください。

4.13 シナリオ C に対するインスタンスの準備

シナリオ) CICS1 インスタンスへログインして、トランザクションから出力するデータを CICS2 インスタンスに存在するファイルへ出力する。

【概要図】



- 1) CICS1 インスタンスへ前項でリモート PCT として作成した ACCT を削除するため、[DBCS] グループの [ACCT] ヘカーソルを合わせ、[削除] アイコンをクリックします。

| ア... | 名前 | グループ | 説明 | アクション |
|------|------|-----------|-------------------------|-------|
| ⌚ | ACCT | DBCS | CICS primer transaction | |
| ⌚ | ACCT | DFH\$ACCT | CICS primer transaction | |

- 2) 既存の ACCT を有効にするため、[DFH\$ACCT] グループにある ACCT の [編集] アイコンをクリックし、[インストール] ボタンをクリックします。

- 3) CICS1 インスタンスへ FCT 定義を作成します。CICS1 インスタンスを [編集] アイコンで開き、画面上部の [モニター] プルダウンメニューから [CICS] > [リソース] > [タイプ別] を選択し、左側ペインの [FCT] をクリックします。

FCT 一覧が表示されますので、[DFH\$ACCT] グループに属する [ACCTFIL] の [編集] アイコンをクリックし、[コピー] ボタンをクリックします。

ファイル管理テーブル - ACCTFIL (DFH\$ACCT) | [インストール](#) | [コピー](#)

- 4) 下記を入力して [保存] ボタンをクリックします。

| 項目名 | 説明 |
|------|--------------------------|
| 名前 | ACCTFIL を入力します。 |
| グループ | SIT で指定している DBCS を選択します。 |

名前*  グループ 

- 5) 一旦保存した [DBCS] グループの [ACCTFIL] を再度開き、下記を入力して、[適用] ボタンをクリックします。

| 項目名 | 説明 |
|------------|---|
| リモート SYSID | SysC 定義の名前である CIC2 を入力します。 |
| リモートファイル | CICS2 インスタンスで実行する FCT 定義名を入力します。ここでは既存の同名 FCT を実行しますので ACCTFIL を入力してください。 |

リモート SysID  リモート ファイル 

- 6) CICS2 インスタンスのファイルにアクセスするため、CICS2 インスタンスのセキュリティ設定を一旦停止します。

- ① CICS2 インスタンスを停止します。



- ② ESCWA から CICS2 をクリックし、[一般] > [セキュリティ] を選択後、VSAM ESM を削除し、[適用] ボタンをクリックします。再度適用する場合には [+追加] ボタンをクリックして VSAM ESM を選択後、[適用] ボタンをクリックします。

セキュリティ マネージャリスト

[+ 追加](#) [定義の表示](#)

1 ✓ VSAM ESM

↓

セキュリティ マネージャリスト

[+ 追加](#) [定義の表示](#)

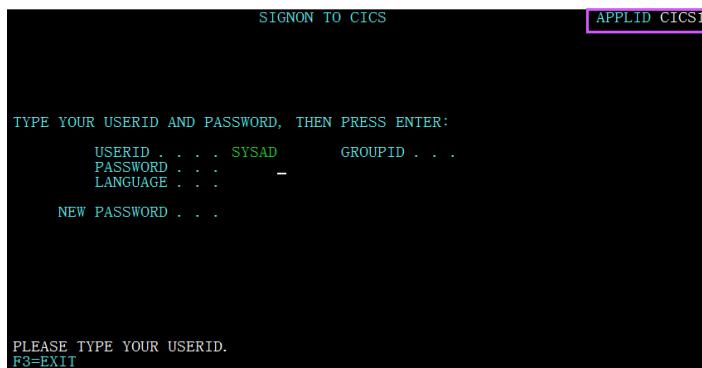
構成情報 

- 7) 追加した FCT を反映させるため、CICS1 を再起動、CICS2 インスタンスを起動します。

4.14 シナリオ C に対するルーティングの実施

シナリオC) CICS1インスタンスへログインして、トランザクションから出力するデータをCICS2インスタンスに存在するファイルへ出力する。

- 1) TN3270エミュレータをCICS1インスタンスへ接続します。



SIGNON TO CICS APPLID CICS1

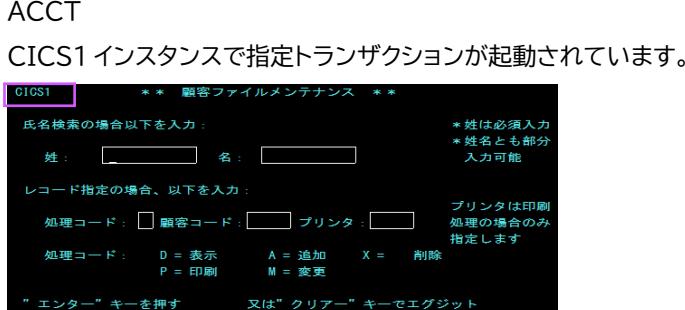
TYPE YOUR USERID AND PASSWORD, THEN PRESS ENTER:

USERID SYSAD GROUPID
 PASSWORD -
 LANGUAGE -

NEW PASSWORD

PLEASE TYPE YOUR USERID.
 F3=EXIT

- 2) [USERID]にSYSADを、[PASSWORD]に設定したパスワードを入力してログインします。
- 3) 画面クリア後に下記トランザクションを実行します。この例では画面左上にはインスタンス名を表示しています。



CICS1 * * 顧客ファイルメンテナンス * *

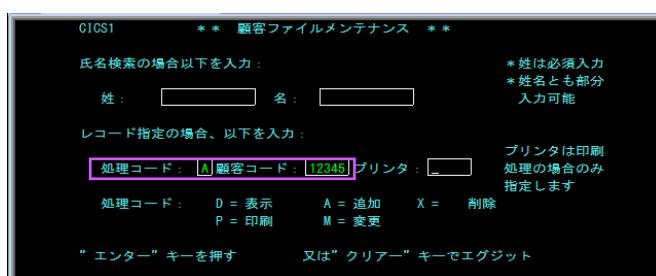
氏名検索の場合以下を入力 : * 姓は必須入力
 姓 : [] 名 : [] * 姓名とも部分
 入力可能

レコード指定の場合、以下を入力 : プリンタは印刷
 処理コード : [] 顧客コード : [] プリンタ : [] 業務の場合のみ
 指定します

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

- 4) データを追加するためタブキーで項目間を移動させ、処理コードにA、顧客コードに12345を入力して実行キーを押下してください。



CICS1 * * 顧客ファイルメンテナンス * *

氏名検索の場合以下を入力 : * 姓は必須入力
 姓 : [] 名 : [] * 姓名とも部分
 入力可能

レコード指定の場合、以下を入力 : プリンタは印刷
 処理コード : 顧客コード : [12345] プリンタ : [] 業務の場合のみ
 指定します

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

- 5) 入力画面が表示されますので、タブキーで移動しながら下記項目を入力後、実行キーを押下してください。

| 項目名 | 入力値 |
|-----|-----|
| 姓 | 山田 |
| 名 | 太郎 |
| 住所1 | 東京都 |

| | |
|---------|--------|
| 住所2 | 港区 |
| 発行カード枚数 | 1 |
| 発行日 | 101010 |
| 理由 | N |
| カードコード | 1 |
| 承認者 | 001 |



- 6) 画面左下に表示される下記メッセージで正常に追加されたことを確認します。

追加要求完了

- 7) TN3270 エミュレータを切断します。

- 8) CICS1 インスタンスと CICS2 インスタンスの DATAFILE ディレクトリに存在する ACCTFIL ファイルを Windows 環境へ転送します。

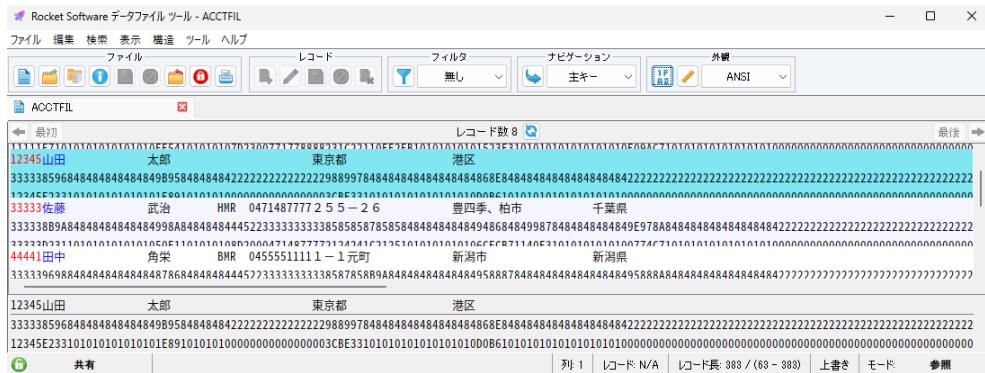
- 9) Enterprise Developer に含まれるデータファイルツールを起動して、出力されたファイル内容を確認します。

 Data File Tools

- 10) [ファイル] プルダウンメニュー > [開く] > [データファイル] を選択してウィンドウを開きます。[ファイル名] の [参照] ボタンをクリックし、[ファイルのタイプ] に [すべてのファイル] を指定して CICS2 インスタンスに存在した ACCTFIL ファイルを共有モードで開きます。

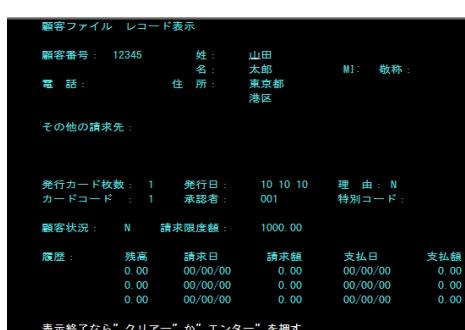
- 11) 前項で作成したデータが、CICS1 のリモート FCT を通じて CICS2 配下のファイルに追加されたことが確認できます。

CICS2 の ACCTFIL データ)



補足) CICS2 インスタンスへログインしてデータを確認することもできます。

処理コード:D、顧客番号:12345



12) CICS1 インスタンスに存在するファイル内容も同様に確認します。追加したデータは存在していません。

CICS1 の ACCTFIL データ)

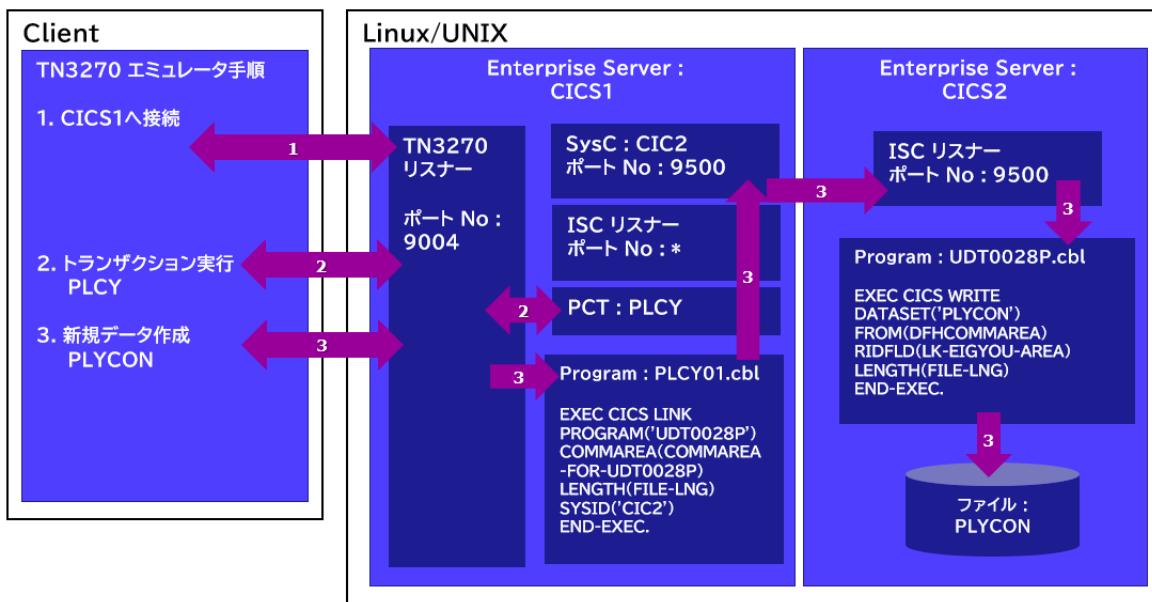


4.15 シナリオ D に対するコードと実施方法の紹介

シナリオ) CICS1 へログインして、トランザクションのプログラムから CICS2 に存在するプログラムへリンクさせる。

CICS1 インスタンスに存在するプログラム PLCY01.cbl から SysC に定義してある CIC2 を利用して CICS2 インスタンスに存在するプログラム UDT0028P.cbl を呼び出し、CICS2 に存在するファイルへデータを出力します。

【 概要図 】



- 1) TN3270 エミュレータを CICS1 インスタンス(9004)へ接続し、PLCY トランザクションを実行します。

PLGY

- 2) CICS1 インスタンスで起動された画面の各項目へ入力後、実行キーを押下します。

| | | | |
|---|-----|-----------------|-----|
| GI051 | ☆☆☆ | 自家用普通乗用車年間保険料試算 | ☆☆☆ |
| <p>お車の初度登録年月日 : * <input type="text" value="2010"/> 年 <input type="text" value="10"/> 月 <input type="text" value="02"/> 日</p> <p>お車の登録番号の種類 : * <input type="text" value="2"/></p> <p>(1 - 普通自動車 , 2 - 小型自動車 , 3 - 小型貨物車 4 - 軽自動車 , 5 - 軽貨物車)</p> <p>主な使用目的 : <input type="text" value="B"/></p> <p>(A - 主に家庭用 , B - 主に業務用)</p> <p>契約距離区分 : * <input type="text" value="5"/> (< 今後一年間のお車の最大走行距離)</p> <p>(1 - 3000KM 以下 , 2 - 5000KM 以下 , 3 - 7000KM 以下 , 4 - 9000KM 以下 , 5 - 11000KM 以下 , 6 - 無制限)</p> <p>記名保険者の氏名 : <input type="text" value="東京"/> <input type="text" value="太郎"/></p> <p>記名保険者の住所 : <input type="text" value="〒111-2222 都道府県 (東京都) 市区町村 (港区) 千代田区 六本木"/></p> <p>書類番号 : <input type="text" value="六本木"/></p> <p>記名保険者の生年月日 : * <input type="text" value="1975"/> 年 <input type="text" value="09"/> 月 <input type="text" value="15"/> 日</p> <p>記名被保険者の免許証の色 : * <input type="text" value="2"/></p> <p>(1 - ゴールド , 2 - ブルー , 3 - グリーン)</p> <p>運転者の範囲 : <input type="text" value="2"/></p> <p>(1 - 本人のみ , 2 - 本人と配偶者 , 3 - 家族のみ , 4 - 限定なし)</p> <p>年間保険料試算額 : <input type="text"/> 円</p> <p>* 計算要素を入力してください</p> | | | |

- 3) プログラムコードから、CICS2 インスタンスに存在するプログラム UDT0028P.cbl が呼び出します。

```
029800* 顧客情報ファイル書き出し機能の呼び出し
029900*
030000  EXEC CICS LINK PROGRAM('UDT0028P')
030100    COMMAREA(COMMAREA-FOR-UDT0028P)
030200    LENGTH(FILE-LNG) SYSID('CIC2')
030300  END-EXEC.
```

- 4) CICS2 インスタンスに存在するプログラム UDT0028P.cbl から値が返却され、同時にファイルに入力内容が出力されます。

年間保険料試算額： 77,175 円
試算された保険料が表示されました

- 5) 入力データが出力された CICS2 インスタンスに存在する PLYCON ファイル内容をデータツールから確認します。最後尾に追加したデータが存在しています。

| ANSI | Prime Key | IEEE | |
|--------------|-----------|------|---------------|
| K51951051731 | 2K2 | | |
| K51951051731 | 1K2 | 山田 | 太郎 |
| K51951051731 | 2K2 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | 3K3 | | |
| K51951051731 | 3K5 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | 2K2 | | |
| K51951051731 | 2K2 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | 2K1 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | 2K2 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | 1K2 | | |
| K51951051731 | 2G3 | 東京 | 太郎 |
| | | | 1112222東京都 港区 |
| | | | 六本木 |

- 6) CICS1 インスタンスに存在する PLYCON ファイル内容をデータツールから確認します。追加データは存在していないため、CICS2 に存在するプログラムへのリンクが成功していることが確認できます。

| ANSI | Prime Key | IEEE | |
|--------------|-----------|------|---------------|
| K51951051731 | 2K2 | | |
| K51951051731 | 1K2山田 | 太郎 | 1001000東京都 港区 |
| K51951051731 | 2K2 | | |
| K51951051731 | / 2K3 | | |
| K51951051731 | 3G5 | | |
| K51951051731 | / 2K3 | | |
| K51951051731 | 2K2 | | |
| K51951051731 | 2K2 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | / 2K1 | | |
| K51951051731 | 2K3 | | |
| K51951051731 | 2K2 | | |
| K51951051731 | / 2K3 | | |
| K51951051731 | 1K1 | | |

4.16 Enterprise Server インスタンスの停止

- 1) CICS1 と CICS2 インスタンスを停止します。



- 2) インスタンスの停止を確認後、Eclipse を終了します。

4.17 リモートマシンの切断

必要であれば、リモートマシンで使用したポートの遮断をルートユーザーで行います。

ポート遮断コマンド) \$COBDIR/remotedev/stoprdodaemon 5000

5. 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。