

Micro Focus メインフレームソリューション

スターターズキット

7. Enterprise Developer データベース連携

7.1 チュートリアル準備

本チュートリアルで使用する例題プログラムは、キットに添付されている DBtutorial.zip に圧縮されています。これを C:¥ の直下に解凍しておきます。

また、作業用に C:¥work というフォルダを作成しておきます。

ここでは JCL バッチからの使用方法として紹介していますが、CICS や IMS プログラムで Oracle, DB2, SQL Server などとのデータベース連携を XA リソース方式で行う場合は同様の設定となります。

なお、対象のデータベースクライアントは同じ環境にインストール済みで、プリコンパイラも含め接続が確認された環境であることが前提です。

本書作成時に使用したデータベース環境は下記の通りです。

Oracle

Oracle クライアント 11g 11.2.0.4.0 64bit
DB: リモート環境 (Oracle 10g 10.2.0.1.0 64bit)
(Oracle Pro*COBOL プリコンパイラ使用)

DB2

DB2 クライアント 9.7.900.250 64bit
DB: ローカル環境 (DB2 9.7.900.250 64bit)
(DB2 PREP プリコンパイラ使用)

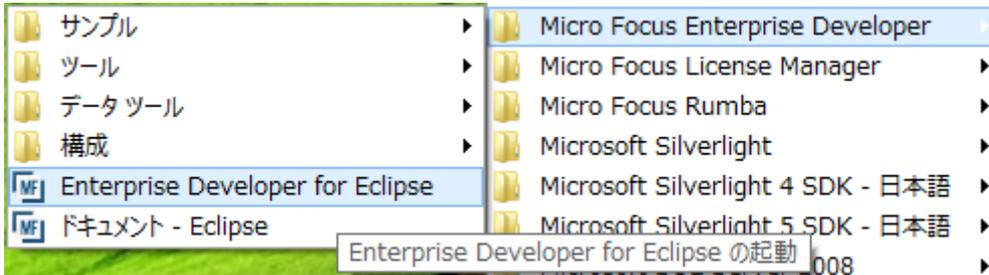
SQL Server

ODBC ドライバー: SQL Server Native Client 11.0
DB: ローカル環境 (SQL Server 2012 R1 64bit)
(ODBC 使用)

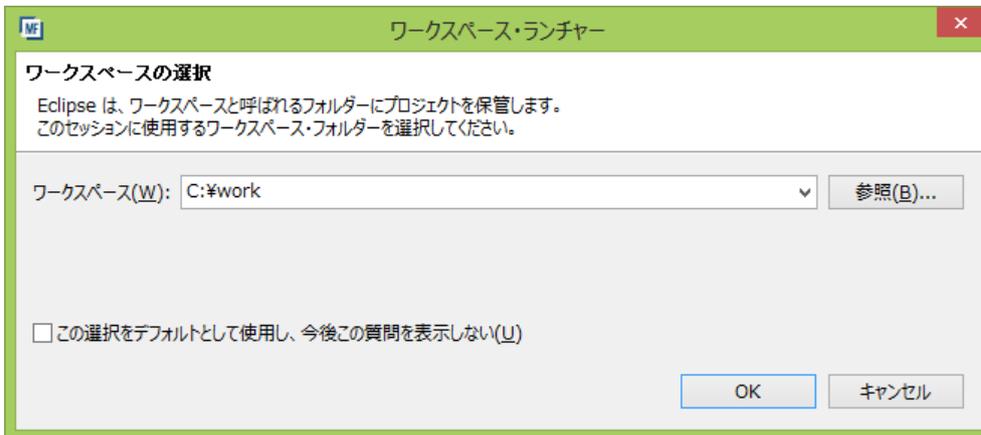
7.2 Enterprise Developer の起動

まず、Enterprise Developer を起動し、新たなワークスペースを作成します。

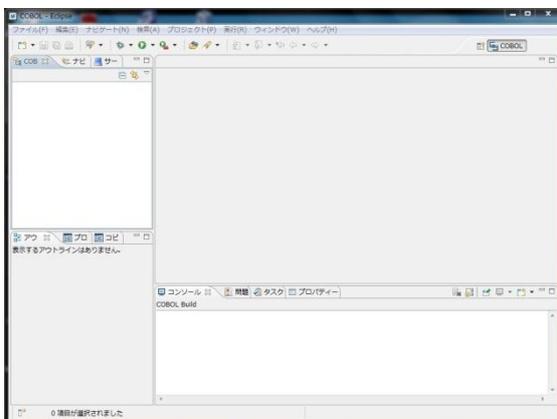
- 1) メニューから Enterprise Developer for Eclipse を選択して起動します。



- 2) 以下のダイアログでは C:¥work を指定し [OK] をクリックします。



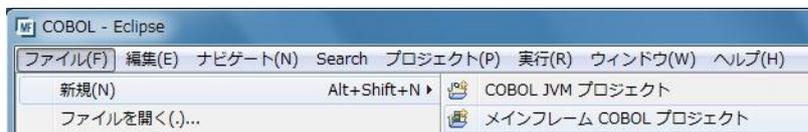
- 3) 「ようこそ」タブの右の X をクリックしてパナールを閉じます。以下のように Eclipse の COBOL パースペクティブが開きます。



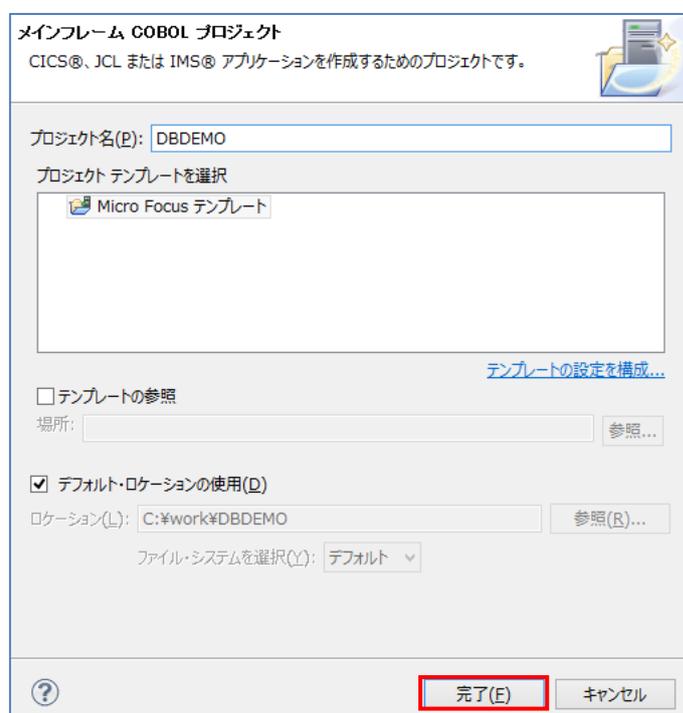
7.3 プロジェクトの新規作成

作成されたワークスペースに新たなプロジェクトを作成します。

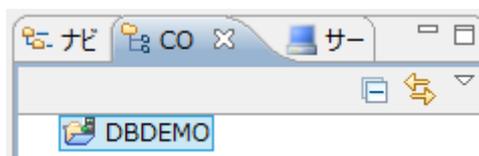
- 1) [ファイル] > [新規] > [メインフレーム COBOL プロジェクト] を選択します。



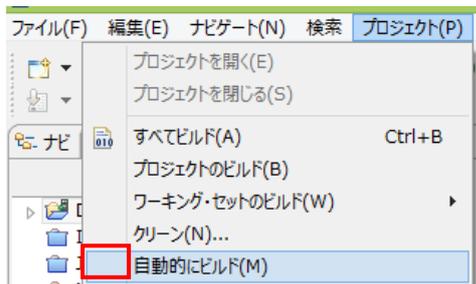
- 2) 以下のダイアログでプロジェクト名を指定します。ここでは“DBDEMO”と命名します。[完了] をクリックします。



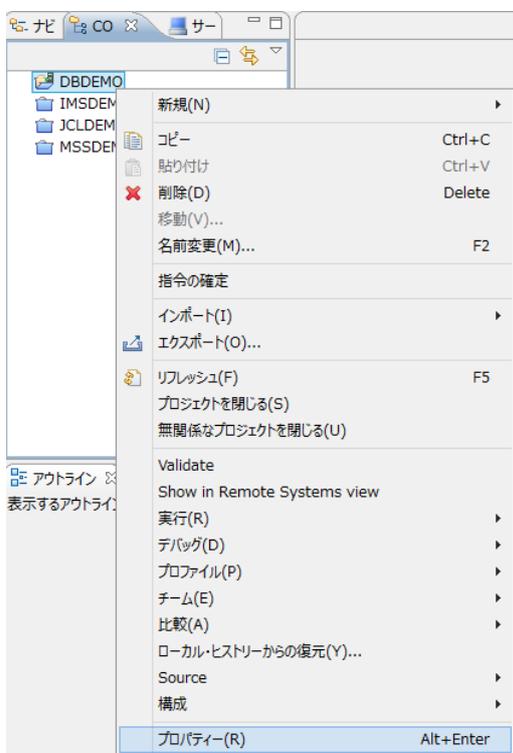
- 3) 空のメインフレームプロジェクトが作成されます。



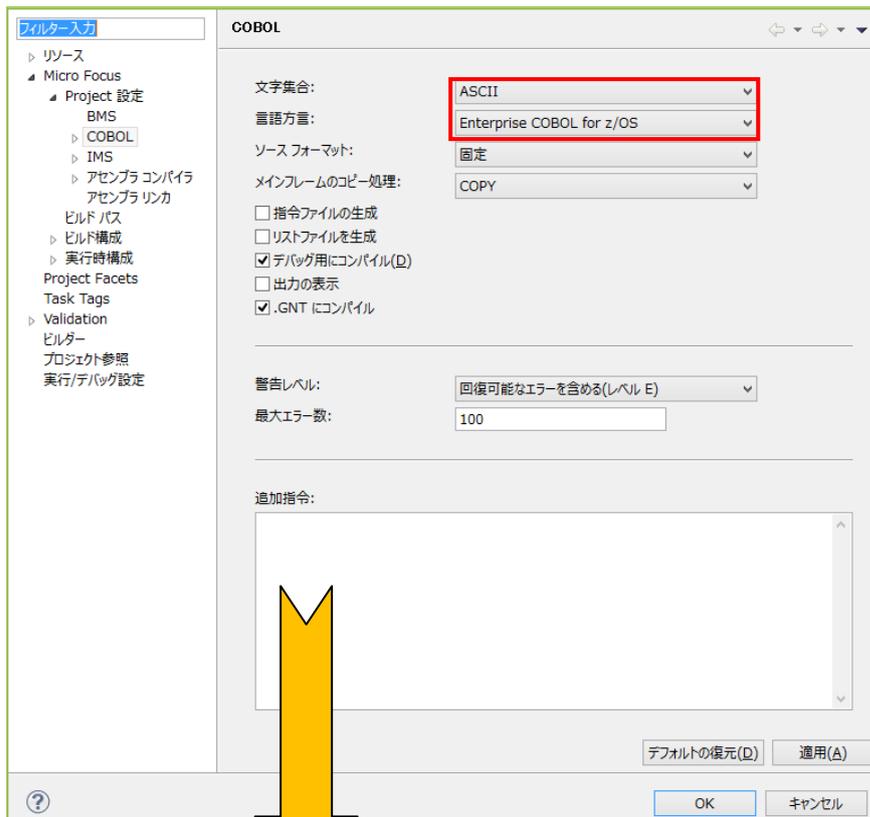
4) [プロジェクト] > [自動ビルド] をオフにしておきます。



5) 作成されたプロジェクトに必要なプロパティの設定を行います。COBOL エクスプローラ内で DBDEMO を右クリックして [プロパティ] を選択します。



6) 以下のようにプロジェクトのプロパティダイアログが開きます。ここでメインフレームプロジェクトの各種設定を行うことができます。左側ペインのツリービューにて [Micro Focus] > [Project 設定] > [COBOL] を開き、以下のように設定してください。本チュートリアルで使用する DB 例題プログラムは IBM Enterprise COBOL の方言を使用しています。



- 7) 使用するデータベース製品に合わせ、[追加指令]欄へ埋め込み SQL 対応のプリプロセッサの設定を追加します。

<Oracle の例>

COBSQL プリプロセッサを使用しますので、以下の指令を追加します。

入力値) P(COBSQL) ENDP

追加指令:

P(COBSQL) ENDP

<DB2 の例>

DB2 ECM プリプロセッサを使用しますので、以下の指令を追加します。

入力値) DB2(DB=SAMPLE,VALIDATE=RUN)

追加指令:

DB2(DB=SAMPLE,VALIDATE=RUN)

<SQL Server の例>

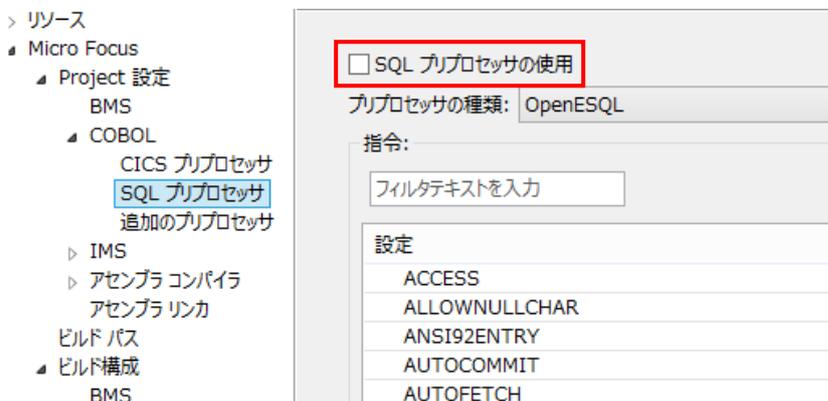
OPENESQL プリプロセッサを使用しますので、以下の指令を追加します。

入力値) SQL(DBMAN=ODBC,BEHAVIOR=JCL,TARGETDB=MSSQLSERVER)

追加指令:

SQL(DBMAN=ODBC,BEHAVIOR=JCL,TARGETDB=MSSQLSERVER)

- 8) 既に前項で埋め込み SQL 対応のプリプロセッサを使用するために追加指令を行っていますので、左側ペインのツリービューにて [Micro Focus] > [Project 設定] > [COBOL] > [SQL プリプロセッサ] を開き、以下のように [SQL プリプロセッサに使用] のチェックをオフにして、[OK] をクリックします。(必要に応じて使い分け可能です)



- 9) 左側ペインのツリービューにて [Micro Focus] > [ビルド構成] > [COBOL] を開き、以下のように [ターゲットの種類]として[すべて INT/GNT ファイル]を選択かつ、[プラットフォーム ターゲット]は [64 ビット]を選択し、一旦、[OK] をクリックし、プロパティダイアログを閉じます。



- 10) 再度、COBOL エクスプローラ内で DBDEMO を右クリックして [プロパティー] を選択します。左側ペインのツリービューにて [Micro Focus] > [Project 設定] > [COBOL] を開き、以下のように [GNT にコンパイル] のチェックをオンにし、[OK] をクリックします。

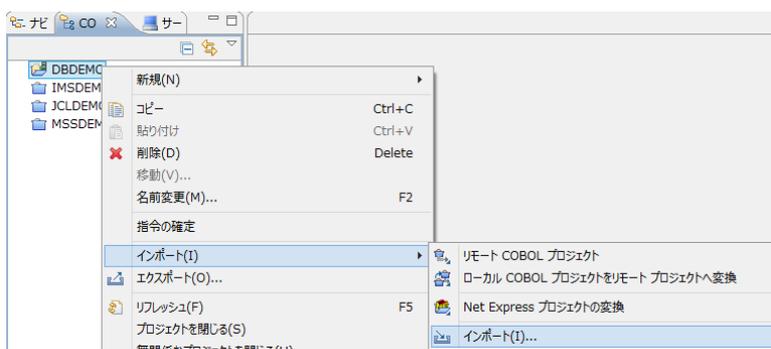


7.4 埋め込み SQL 付き COBOL ソースのインポート

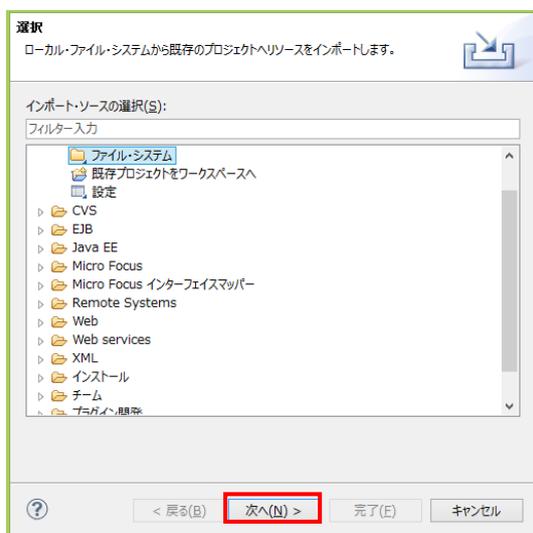
作成されたプロジェクトに例題プログラムをインポートします。

オリジナルの埋め込み SQL 付き COBOL ソースは予め Micro Focus 形式の COBOL ソースにプリコンパイルしてから使用することも可能ですが、製品機能であるプリプロセッサ機能からプリコンパイラを呼び出して内部的にプリコンパイルすることにより、オリジナルのソースのイメージのままデバッグが可能となり管理もし易くなります。ここでは後者の方法を紹介します。

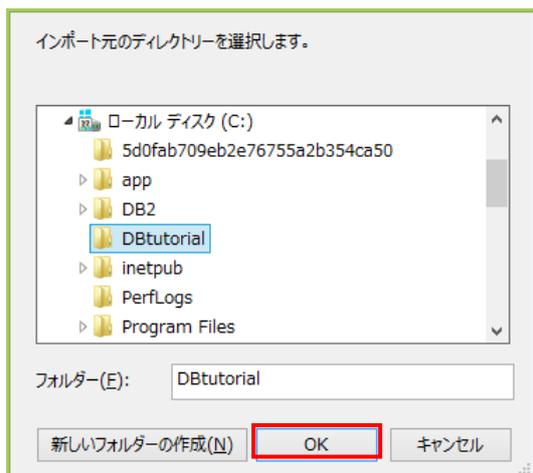
- 11) COBO エクスプローラ内で DBDEMO を右クリックして [インポート] > [インポート] を選択します。



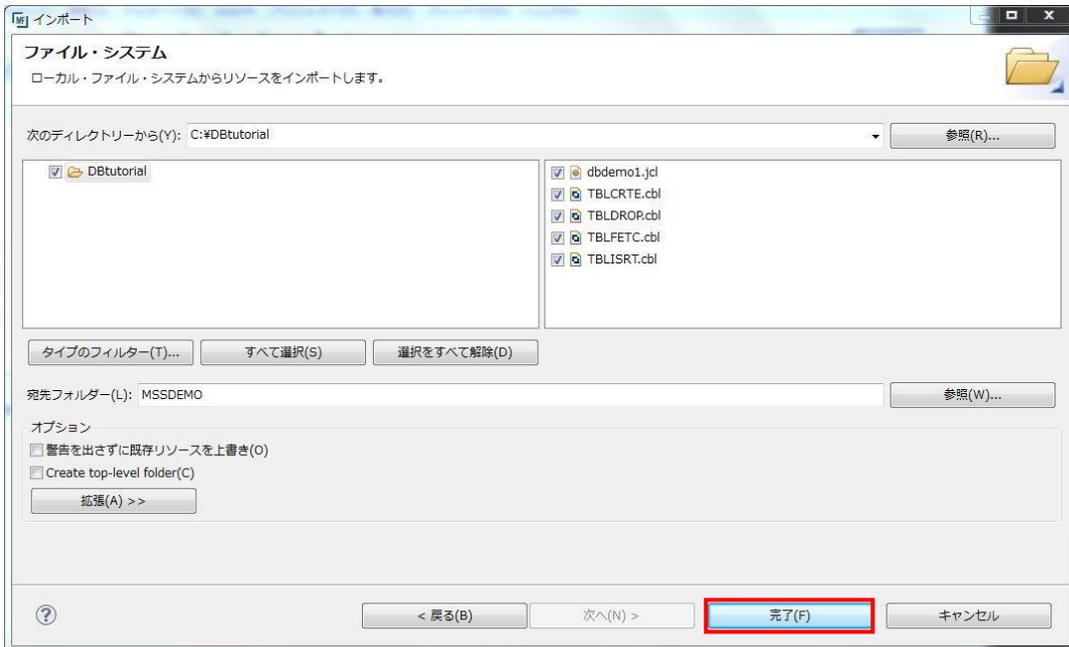
12) 以下のダイアログで [一般] > [ファイルシステム] を選択し、[次へ] をクリックします。



13) 参照ボタンをクリックし、以下のダイアログで C:\DBtutorial を選択し、[OK] をクリックします。

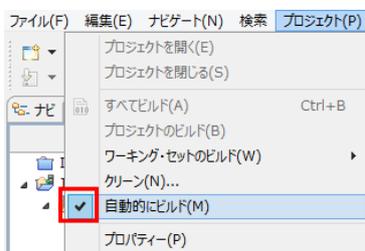


14) C:\¥DBtutorial 直下に解凍されているファイルをすべて選択し、[完了] をクリックします。



7.5 プログラムのコンパイル

1) [自動的にビルド] をオンにします。



2) 自動ビルドが実行され、以下のように再コンパイルが完了します。なお、警告が出力される場合がありますが、この例題では無視して構いません。



7.6 XA スイッチモジュールの生成

この環境で実行するプログラム内では、XA スイッチモジュール経由でデータベースと接続することになります。ここでは使用するデータベース製品に合わせた XA スイッチモジュールを作成します。作成手順は以下の通りです。

- 1) プリコンパイルを行うため、下記フォルダを書き込み権限があるフォルダ配下へコピーします。当チュートリアルでは C:¥直下へコピーします。

【理由1】 Oracle のプリコンパイラはパスに英数字とアンダースコア以外は許容しない

【理由2】 製品関連フォルダの書き込み権限によるトラブルを避ける

【コピー元フォルダ例】: Micro Focus 製品があるパス

C:¥Program Files (x86)¥Micro Focus¥Enterprise Developer¥src¥enterpriseserver¥xa

【コピー先フォルダの例】: 書き込み権限があるパス

C:¥xa



- 2) スタートメニューで、[Micro Focus Enterprise Developer] > [ツール] > [Enterprise Developer コマンドプロンプト(64-bit)] を管理者として実行します。この例題では 64 ビットですが、DB に合わせて 32 ビットもしくは 64 ビットを選択してください。

C:¥xa ディレクトリに移動します。

```
C:¥Users¥Administrator¥Documents>cd c:¥xa
c:¥xa>
```

3) 使用するデータベース製品に合わせた XA スイッチモジュールを build コマンドで作成します。

<Oracle の例>

build < ora11 または ora12 > を実行します。ここでは下記コマンドを実行します。

build ora11

```
C:\Users\tarot\Documents>cd c:\xa
c:\xa>build ora11
Building 64-bit switch module...
Micro Focus COBOL
Version 2.2.02170 Copyright (C) Micro Focus 1984-2014. All rights reserved.

* Cobsql Integrated Preprocessor
* CSQL-I-018: Oracle プリコンパイラトランスレータを起動します。
* CSQL-I-020: Oracle プリコンパイラの出力を処理中。
* CSQL-I-001: COBSQL: チェッカへの引き渡しを完了しました。
* チェック終了: エラーはありません- コード生成を開始します
* Generating ESORAXA
* Data:      8704      Code:      34336      Literals:      2224
Micro Focus COBOL - CBLLINK utility
Version 2.2.2.44 Copyright (C) Micro Focus 1984-2014. All rights reserved.

Microsoft (R) Incremental Linker Version 10.00.40219.01
Copyright (C) Microsoft Corporation. All rights reserved.

ESORAXA.obj
cbllids00001130.obj
  Creating library ESORA11XA.lib and object ESORA11XA.exp
Microsoft (R) Manifest Tool version 5.2.3790.2076
Copyright (c) Microsoft Corporation 2005.
All rights reserved.

c:\xa>
```

<DB2 の例>

ここでは下記コマンドを実行します。

```
build db2
```

```
C:\%xa>build db2
Building 64-bit switch module...
Micro Focus COBOL - CBLLINK utility
Version 2.2.2.39 Copyright (C) Micro Focus 1984-2014. All rights reserved.

Micro Focus COBOL
Version 2.2.02161 Copyright (C) Micro Focus 1984-2014. All rights reserved.
* チェック終了：エラーはありません - コード生成を開始します
* Generating ESDB2XA
* Data:      3792      Code:      8051      Literals:      992
Microsoft (R) Incremental Linker Version 10.00.40219.01
Copyright (C) Microsoft Corporation. All rights reserved.

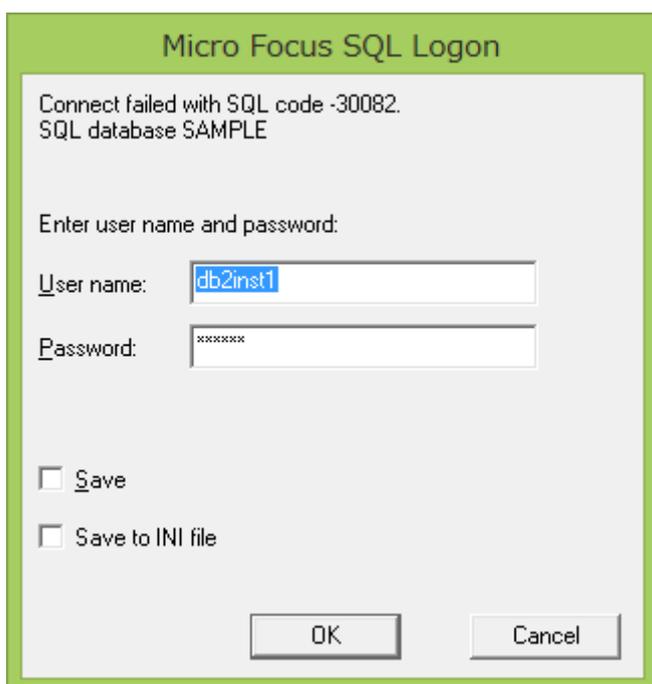
ESDB2XA.obj
cbllids0000D640.obj
  Creating library ESDB2XA.lib and object ESDB2XA.exp
Microsoft (R) Manifest Tool version 5.2.3790.2076
Copyright (c) Microsoft Corporation 2005.
All rights reserved.

C:\%xa>
```

JCL 機能で IKJEFT01 ユーティリティを使用する場合は、xxx1pc.dll サブモジュール生成も必要のため、build db21pc < DB のエイリアス > を実行します。ここでは下記コマンドを実行します。

```
build db21pc sample
```

DB2 へアクセスするためのログイン画面が表示されますので、ユーザ ID とパスワードを入力します。
“build db21pc sample ユーザ ID パスワード”を実行するとログイン画面は出力されません。



```
C:¥xa>build db21pc sample
Building 64-bit switch module...
Micro Focus COBOL
Version 2.2.02161 Copyright (C) Micro Focus 1984-2014. All rights reserved.
* チェック終了：エラーはありません- コード生成を開始します
* Generating ESDB20PC
* Data:          5992      Code:          11078      Literals:          976
Micro Focus COBOL - CBLLINK utility
Version 2.2.2.39 Copyright (C) Micro Focus 1984-2014. All rights reserved.

Microsoft (R) Incremental Linker Version 10.00.40219.01
Copyright (C) Microsoft Corporation. All rights reserved.

ESDB20PC.obj
cblls00010648.obj
  Creating library ESDB20PC.lib and object ESDB20PC.exp
Microsoft (R) Manifest Tool version 5.2.3790.2076
Copyright (c) Microsoft Corporation 2005.
All rights reserved.

If you have upgraded your installation from an earlier version of Net
Express or Studio, then you may need to additionally rebuild the main
switch module, that is, execute :

    build db2

Unable to locate the COBOL bin64 directory.

If you intend to execute JES-initiated transactions under Enterprise Server
you must copy ESDB20PC.DLL to a directory on your PATH, such as your COBOL
bin64 directory.

If you do not do so, then such transactions will not be able to communicate
with the database server.

C:¥xa>
```

正常に作成されました。メッセージにあるように “ESDB20PC.DLL” ファイルを製品の bin64 フォルダへコピーします。32 ビットの場合は bin フォルダへコピーします。

例) C:¥Program Files (x86)¥Micro Focus¥Enterprise Developer¥bin64

<SQL Server の例>

ここでは下記コマンドを実行します。

build mssql

```
C:\xa>build mssql
Building 64-bit switch module...
Micro Focus COBOL - CBLINK utility
Version 2.2.2.39 Copyright (C) Micro Focus 1984-2014. All rights reserved.

Micro Focus COBOL
Version 2.2.02161 Copyright (C) Micro Focus 1984-2014. All rights reserved.
* チェック終了：エラーはありません- コード生成を開始します
* Generating ESMSSQL
* Data:      16      Code:      22844      Literals:      2544
Microsoft (R) Incremental Linker Version 10.00.40219.01
Copyright (C) Microsoft Corporation. All rights reserved.

ESMSSQL.obj
cblllds0001F1E4.obj
  Creating library ESMSSQL.lib and object ESMSSQL.exp
Microsoft (R) Manifest Tool version 5.2.3790.2076
Copyright (c) Microsoft Corporation 2005.
All rights reserved.

Unable to locate the COBOL bin64 directory.

If you intend to execute JES-initiated transactions under Enterprise Server
you must copy ESMSSQL.DLL to a directory on your PATH, such as your COBOL
bin64 directory.

If you do not do so, then such transactions will not be able to communicate
with the database server.

C:\xa>
```

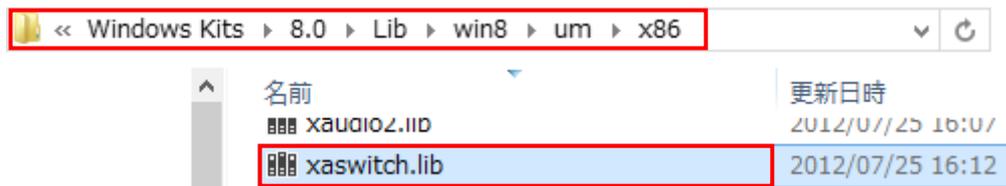
正常に作成されました。メッセージにあるように “ESMSSQL.DLL” ファイルを製品の bin64 フォルダへコピーします。32 ビットの場合は bin フォルダへコピーします。

例) C:\Program Files (x86)\Micro Focus\Enterprise Developer\bin64

xaswitch.lib の読み込み失敗でリンクエラーとなる場合は、環境変数「LIB」へ左記ファイルパスを追加し、コマンドプロンプトを再起動後に再ビルドしてください。

32 ビット例: C:\Program Files (x86)\Windows Kits\8.0\Lib\win8\um\x86

64 ビット例: C:\Program Files (x86)\Windows Kits\8.0\Lib\win8\um\x64

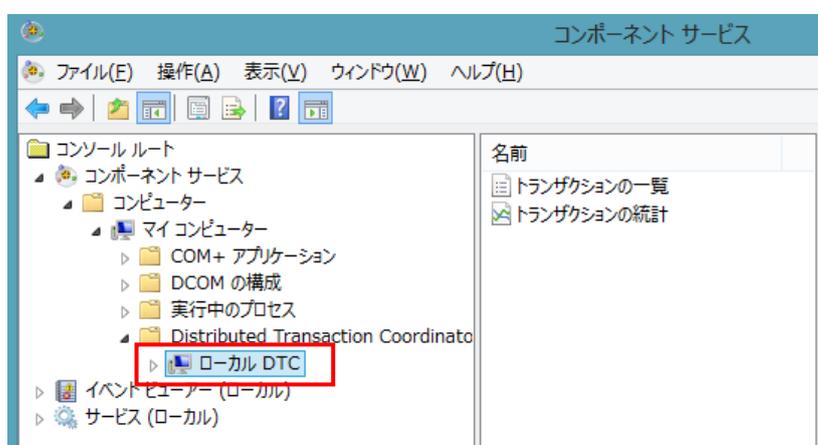


- 4) 使用するビット数に合わせた ODBC データソースを [コントロールパネル]>[管理ツール]>[ODBC データソース] から追加します。

注意)ODBC データソース名称が Enterprise Server へ設定する XA リソース定義の OPEN 文字列で使用する DSN 名称となります。

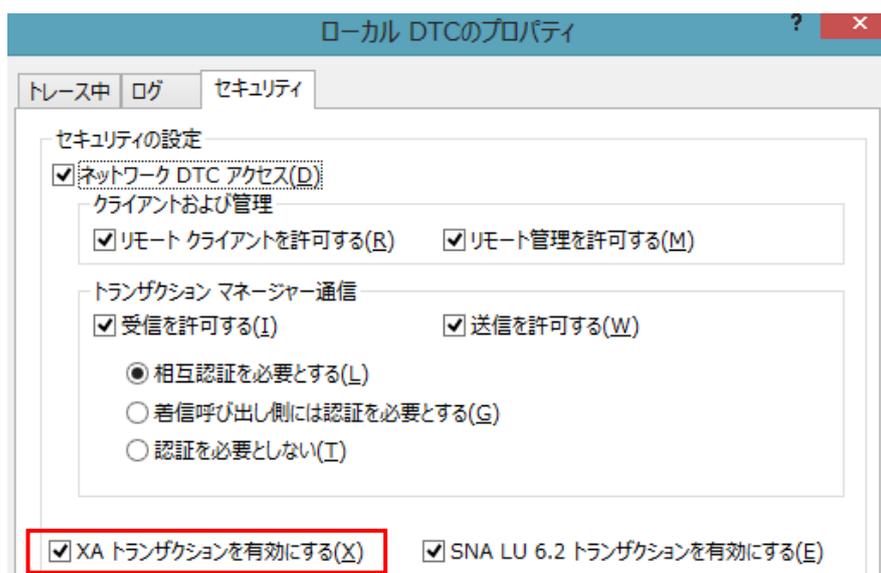
- 5) XA スイッチモジュールを有効にするため下記設定を確認します。

[コントロールパネル]>[管理ツール]>[コンポーネントサービス]>[コンピューター]>[マイコンピュータ]>[Distributed Transaction Coordinator]>[ローカル DTC]まで展開します。



[ローカル DTC]を右クリックして[プロパティ]を選択し、[セキュリティ]タブへ移動します。

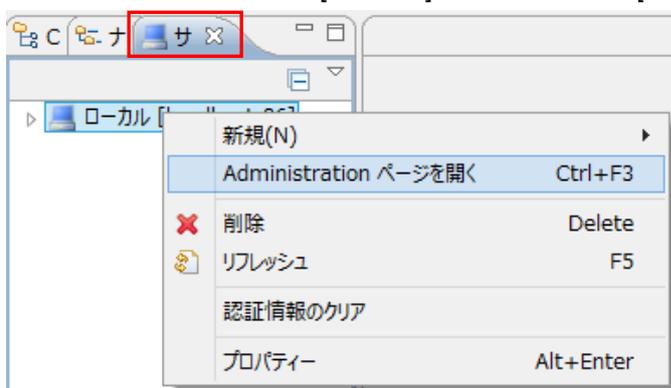
[XA トランザクションを有効にする]にチェックされていることを確認、もしくはチェックして有効にします。



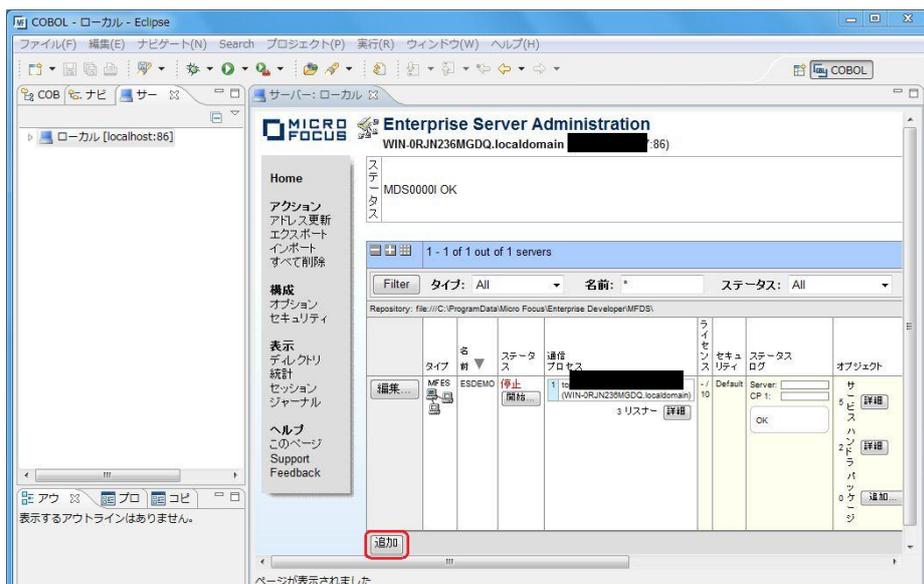
7.7 リージョンの作成と XA リソースの設定

本チュートリアルでは例題 JCL を Enterprise Server にサブミットして実行します。これには Enterprise Developer に内蔵されているテスト用のメインフレームランタイム環境を使用します。これは Enterprise Server と呼ばれるミドルウェアであり、Enterprise Developer には開発用の Enterprise Server が内蔵されています。これがメインフレームアプリケーションのテスト・デバッグのために使用されます。またマイグレーションにおいては本番実行用の Enterprise Server 製品を使用します。

- 1) Enterprise Developer 内で開発用の Enterprise Server を操作するにはサーバーエクスプローラを使用します。サーバーエクスプローラは COBOL エクスプローラの後ろに隠れていますので、このタブをクリックします。[ローカル] を右クリックして [Administration ページを開く] を選択します。



- 2) Enterprise Server の管理コンソールが開きます。メインフレームアプリケーションの実行のためには新たなサーバー (リージョン) を定義する必要があります。画面下部の [追加] ボタンをクリックします。



- 3) 以下の画面に遷移します。新規に作成するサーバー名として DBDEMO を入力し、[動作モード] は[64-bit]を選択後、[次へ] をクリックします。

Home | ステータス | MDS0000I OK | Wed Oct 15 11:40:04 2014
[mfuser] [Page id: a001]

アクション
アドレス更新
エクスポート
インポート
すべて削除

構成
オプション
セキュリティ

表示
ディレクトリ
統計
セッション

サーバー追加 (Page 1 of 3):

サーバー名:

動作モード:

32-bit 64-bit

You cannot change your choice of working mode once a server is created, although you can change it when copying or importing a server.

- 4) 以下の画面では“Micro Focus Enterprise Server with Mainframe Subsystem Support” のラジオボタンを選択し、[次へ] をクリックします。

Home | ステータス | MDS0000I OK | Wed Oct 15 11:42:28 2014
[mfuser] [Page id: a101]

アクション
アドレス更新
エクスポート
インポート
すべて削除

構成
オプション
セキュリティ

表示
ディレクトリ
統計
セッション
ジャーナル

ヘルプ
このページ
Support
Feedback

サーバー追加 (Page 2 of 3):

サーバー名:

サーバータイプ:

MFES (MSS) **Micro Focus Enterprise Server**
An enterprise server that provides an execution environment for COBOL application programs running as services in a service orientated architecture.

MFES (MSS) **Micro Focus Enterprise Server with Mainframe Subsystem Support**
An enterprise server that also provides an execution environment for CICS applications that have been migrated from the mainframe.

You can change your choice of server type later.

- 5) 以下の画面では、[TN3270リスナーの作成] のチェックをオフにし、[追加] をクリックします。

生成オプション:

TN3270リスナーの作成

構成情報

説明

Micro Focus Enterprise Server

- 6) 以下のように64ビット対応の DBDEMO が新規に追加されました。作成された DBDEMO のプロパティを設定します。左端の [編集...] ボタンをクリックします。



- 7) [サーバー] > [プロパティ] > [MSS...] > [JES] > [General] のタブを開き、以下のように各プロパティを設定し、[Apply] をクリックします。

なお、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。

- ① [ジョブ入力サブシステム有効] のチェックをオンにします。
- ② [JES プログラムパス] はジョブステップで実行される COBOL アプリケーションの探索先パスですので、開発プロジェクトの bin ディレクトリを指定しています。
- ③ [システムカタログ] は、JES リージョンで仮定されるマスターカタログの置き場所です。
- ④ [データセットの省略時ロケーション] はジョブの実行とともに生成されるスプールデータやカタログされるデータセットの置き場所です。
- ⑤ [システムプロシージャライブラリ] は、ジョブの実行時に使用されるプロシージャライブラリの名前です。

プロパティ... 構成 診断... 過去の統計

一般 XAリソース (1) **MSS... (✓)** MQ... スクリプト アクセス権 セキュリティ

メインフレーム サブシステム サポート有効:

CICS (✓) **JES... (✓)** IMS... PL/I

General Initiators (1) Printers (0)

ジョブ入力サブシステム有効:

JESプログラムパス:
C:\work\DBDEMO\New_Configuration.bin

システムカタログ:
C:\work\DBDEMO\DATAFILE\CATALOG.DAT

データセットの省略時刻ケーション:
C:\work\DBDEMO\DATAFILE

システムプロシージャライブラリ:
SYS1.PROCLIB

Fileshare Configuration Location:

Apply

 要事項

パスは全て半角英数字で指定してください。
以降も同様です。

8) 「JES...」 > [Initiators] タブを開き、[追加] をクリックします。

CICS (✓) **JES... (✓)** IMS... PL/I

一般 **Initiators (0)** Printers (0)

追加

9) 以下のように入力し [追加] ボタンをクリックします。

一般 **Initiators (0)** Printers (0)

▲ Add Initiator...

名前:
INITABC

Class:
ABC

説明:
クラスABCのイニシエーター

キャンセル **追加**

10) 以下のようにジョブクラス A, B, C に対する JES イニシエータが定義されます。

一般	Initiators (1)	Printers (0)	
	名前	クラス	説明
<input type="button" value="編集..."/>	INITABC	ABC	クラスABCのイニシエーター
<input type="button" value="追加"/>			

11) リージョンで指定したデータセットを格納する DATAFILE フォルダを C:¥work¥DBDEMO 配下に作成しておきます。

12) JES リージョンで指定したシステムプロシジャライブラリのフォルダを C:¥work¥DBDEMO¥DATAFILE 配下に作成しておきます。

13) XA リソースを使用可能とするための準備を行います。[サーバー] > [プロパティ] > [XA リソース] を開き、[追加] ボタンをクリックします。



14) 使用するデータベース製品に合わせた XA リソースを設定します。

[ID]: IKJEFT ユーティリティに渡す DSN TSO コマンドの SYSTEM パラメタで渡すサブシステム名 (このチュートリアルでは XADB とします)

[名前]: XA リソース名として任意の名前を指定しますが、Oracle は "ORACLE_XA" 固定です。

[モジュール]: XA スイッチメインモジュールのパスと DLL ファイル名

<<指定 DLL 名>> Oracle:ESORA11XA.DLL、DB2:ESDB2XA.DLL、SQLServer:ESMSSQL.DLL

[OPEN 文字列]: DB オープン用の指定 (設定済みの DB 環境設定に依存)

<Oracle の例>

例えば以下のように設定し、[追加] ボタンをクリックします。

一般	XAリソース (3)	MSS... (✓)	MQ...	スクリプト	アク
----	-------------------	------------	-------	-------	----

ID:

名前:

モジュール:

OPEN文字列:

CLOSE文字列:

説明:

有効:

<DB2 の例>

例えば以下のように設定し、[追加] ボタンをクリックします。

一般	XAリソース (0)	MSS... (✓)	MQ...	スクリプト	アク
----	-------------------	------------	-------	-------	----

ID:

名前:

モジュール:

OPEN文字列:

CLOSE文字列:

説明:

有効:

<SQL Server の例>

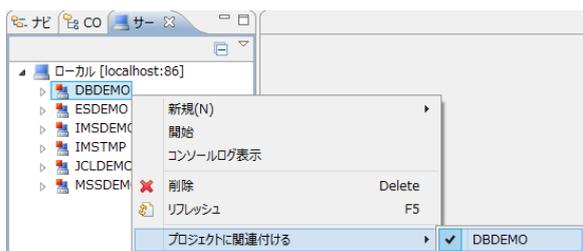
例えば以下のように設定し、[追加] ボタンをクリックします。

前項で作成した ODBC データソース名称を OPEN 文字列に指定します。

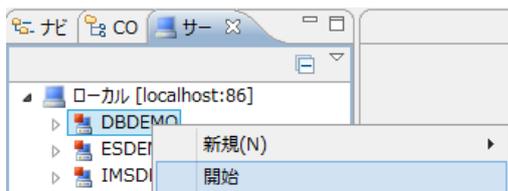
・DSN= ODBC データソース名称

一般	XAリソース (2)	MSS... (✓)	MQ...	スクリーン
ID: XADB				
名前: SQLSVR_XA				
モジュール: C:\xa\ESMSSQL.DLL				
OPEN文字列: DSN=SQLSVR				
CLOSE文字列: 				
説明: 				
有効: <input checked="" type="checkbox"/>				
<input type="button" value="追加"/> <input type="button" value="キャンセル"/>				

15) DBDEMO にプロジェクトを関連付けます。



16) DBDEMO を開始します。



17) XA スイッチモジュールが Enterprise Server に正しくロードできると、Enterprise Server Administration のコンソールログには以下のメッセージが出力されます。(DB2 の場合、NAME(DB2 for WINDOWS) となり、SQL Server の場合、NAME(Micro Focus SQL Server XA struct) となります。)

サーバー...
リスナー (2)
サービス (4)
ハンドラ (3)
パッケージ (0)

プロパティ...
構成
診断...
過去の統計

トレース
ダンプ
ESコンソール
CSコンソール

Show entries from to of 50 total entries
 Show last lines

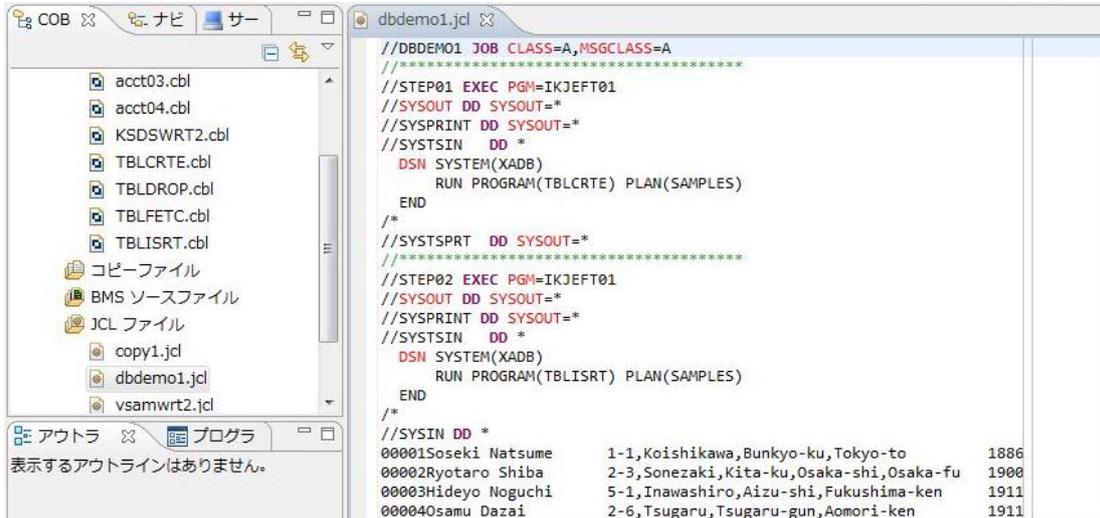
Entry	Event
41	141015 11542790 62280 DBDEMO JES000042I SSTM not enabled: CICS 11:54:27
42	141015 11542813 62280 DBDEMO CASXO0020I XADB XA interface loaded. Name(Oracle_XA), Registration Mode(Static) 11:54:27
43	141015 11542813 62280 DBDEMO CASXO0021I XADB XA interface local transaction support enabled 11:54:28
44	141015 11542836 62280 DBDEMO CASXO0015I XADB XA interface initialized successfully 11:54:28
45	141015 11542857 62280 DBDEMO CASSI5001I PLTPI Phase 1 - No PLT Specified 11:54:28
46	141015 11542857 62280 DBDEMO CASSI5040I Active SEP memory strategy set to x'00000001', retain count 100 11:54:28
47	141015 11542899 63208 DBDEMO CASBJ0005I Batch initiator started for job classes "ABC" 11:54:28
48	141015 11542921 62888 DBDEMO CASSI1800I SEP initialization completed successfully 11:54:29
49	141015 11542943 61384 DBDEMO CASSI1800I SEP initialization completed successfully 11:54:29
50	141015 11542964 62888 DBDEMO CASSI5021I PLTPI Phase 2 - No PLT Specified 11:54:29

 **重要事項**

「プラットフォーム ターゲット = XA リソース = Enterprise Server の動作モード = DB」のビット数が一致することが正常に稼働するために必要な条件です。

7.8 データベースアクセスを含む COBOL バッチプログラムの実行

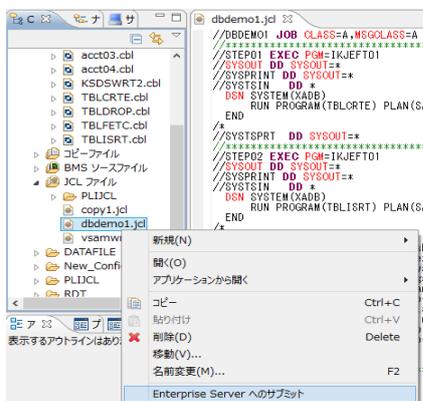
1) COBOL エクスプローラ内で dbdemo1.jcl をダブルクリックしエディタで開きます。



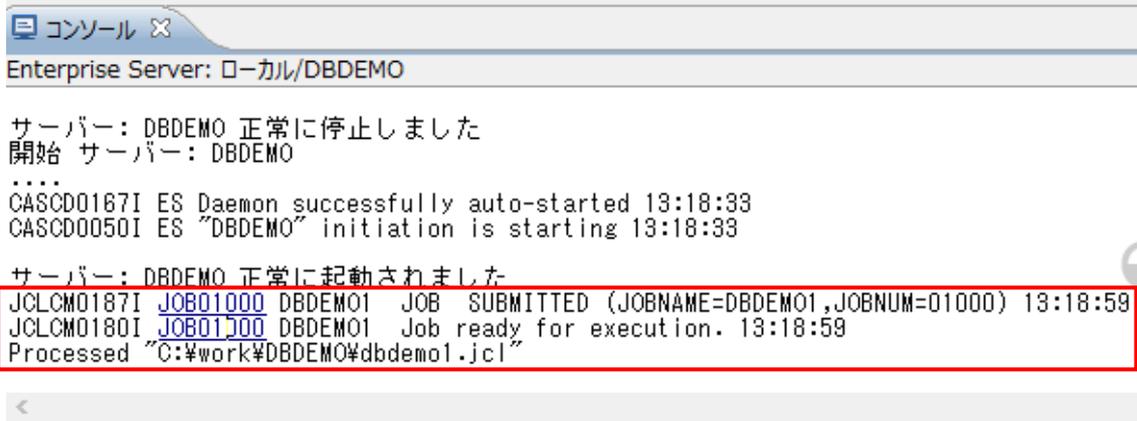
このジョブは4つのステップから構成され下記の様に連携されています。

- STEP01:
DB にテーブルを新規作成
- STEP02:
JCL に記述された SYSIN データをテーブルに追加
- STEP03:
テーブルからデータを全件読み出し SYSOUT へ出力
- STEP04:
テーブルを削除

2) COBOL エクスプローラ内で dbdemo1.jcl を右クリックして [Enterprise Server へのサブミット] を選択します。



3) 以下のように DBDEMO1 ジョブが実行された旨のメッセージがコンソールに表示されます。



```
コンソール x
Enterprise Server: ローカル/DBDEMO

サーバー: DBDEMO 正常に停止しました
開始 サーバー: DBDEMO
.....
CASC00167I ES Daemon successfully auto-started 13:18:33
CASC00050I ES "DBDEMO" initiation is starting 13:18:33

サーバー: DBDEMO 正常に起動されました
JCLCM0187I JOB01000 DBDEMO1 JOB SUBMITTED (JOBNAME=DBDEMO1,JOBNUM=01000) 13:18:59
JCLCM0180I JOB01000 DBDEMO1 Job ready for execution. 13:18:59
Processed "C:\work\DBDEMO\dbdemo1.jcl"
```

ハイパーリンクをクリックすると自動的にスプールが表示されますので、以降は、JCL チュートリアルを参考にして、スプールビューの Out Hold キューおよび Output キューを開き、実行された DBDEMO1 ジョブの結果を確認するなど、バッチプログラムのデバッグも同様に実行してみてください。また、各 DB 付属の管理ツールを使用すると、各ステップにおける DB テーブルの状況が確認できます。

以上、