

Enterprise Developer チュートリアル

メインフレーム COBOL 開発 : データベース連携

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、データベース接続を使用した JCL の実行までを行い、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 Enterprise Developer 11J for Visual Studio 2022 がインストールされていること
- 使用マシンに対象のデータベースクライアントがインストール済みで、プリコンパイラを含め動作確認済みであること

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. XA スイッチモジュールの生成
7. 文字エンコーディングの設定
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. 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
11. Enterprise Server インスタンスの停止

4. 免責事項

3.1 チュートリアル準備

例題プログラムに関連する資源を用意します。

- 1) 使用する例題プログラムは、キットに添付されている DBtutorial.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 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 を含む [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。

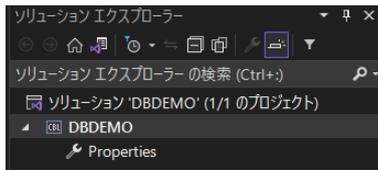


- 2) 下記項目を入力後、[作成] ボタンをクリックします。

項目名	説明
プロジェクト名	任意ですが、ここでは DBDEMO 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。



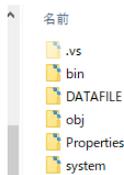
- 3) [ソリューション エクスプローラー] へ作成したプロジェクトが表示されます。



- 4) プロジェクトを作成したことにより C:\¥VS¥DBDEMO フォルダが作成されています。このフォルダ配下に JES 機能で使用するフォルダをあらかじめ用意しておきます。

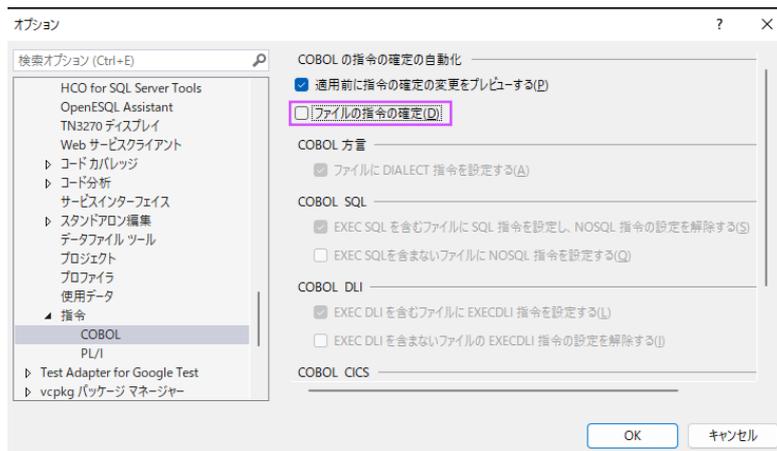
カタログファイルやスプールファイルを配置するための DATAFILE フォルダと、実行時にログなどを格納する system フォルダを C:\¥VS¥DBDEMO 配下へ作成します。

レディスク (C:) > VS > DBDEMO >



- 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。[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。

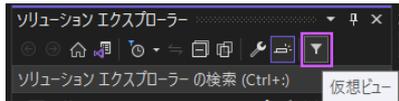
左側ツリービューの [Rocket ツール] > [指令] > [COBOL] > [ファイルの指令の確定] チェックボックスをオフにして [OK] ボタンをクリックします。



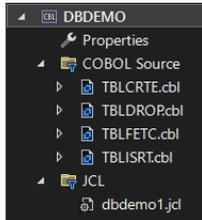
- 6) 用意した例題プログラム類をインポートします。DBDEMO プロジェクトを右クリックして [追加] > [既存の項目] を選択し、既存項目の追加ウィンドウにて C:\¥DBtutorial を指定すると内容が表示されますので、全ファイルを選択後 [追加] ボタンをクリックします。この実行により、プロジェクトフォルダへ例題プログラムが配置されます。

名前	種類
dbdemo1.jcl	JCL ソースコード
TBLCRTE.cbl	COBOL ソースファイル
TBLDROP.cbl	COBOL ソースファイル
TBLFETC.cbl	COBOL ソースファイル
TBLISRT.cbl	COBOL ソースファイル

- 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 DBDEMO 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

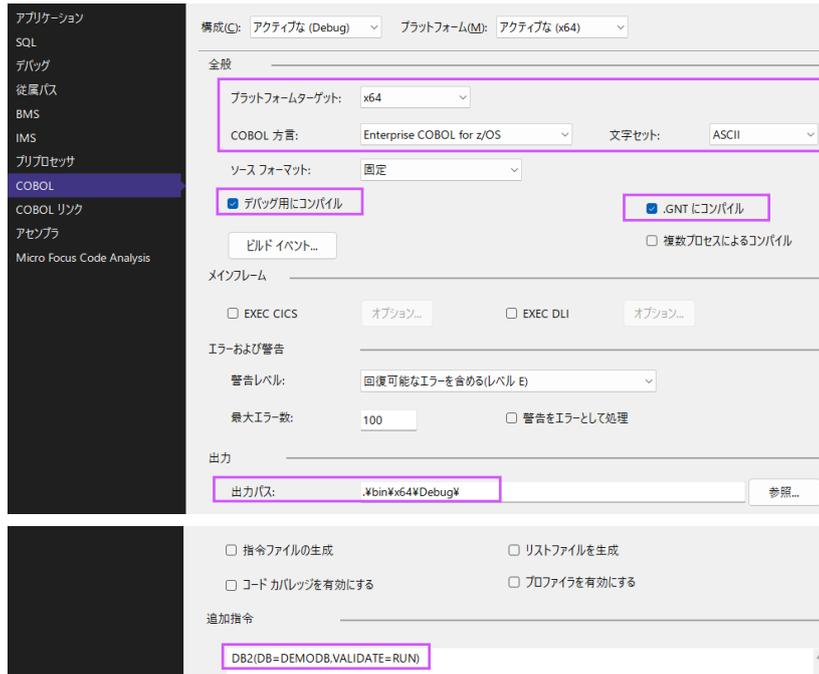
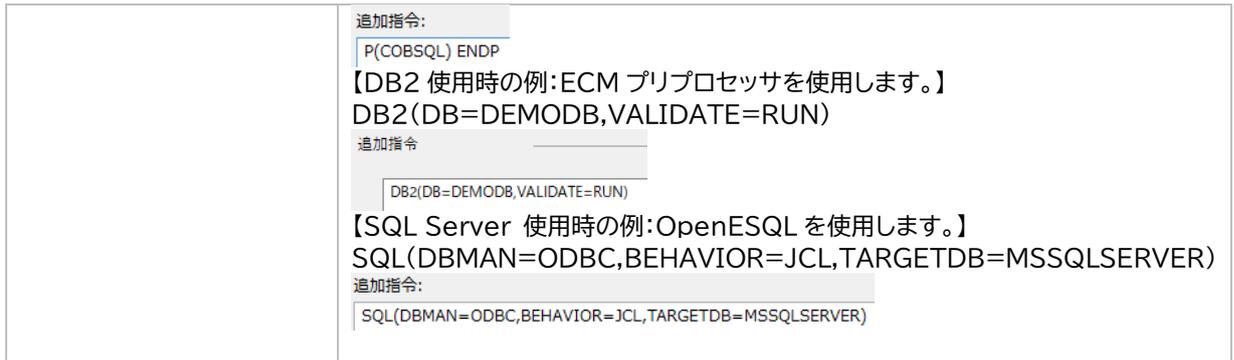
プログラム内容に沿ったプロジェクトのプロパティを設定します。製品には、プリプロセッサ機能からプリコンパイラを呼び出して、埋め込み SQL 付き COBOL ソースを内部的にプリコンパイルする機能があります。これを使用することにより、オリジナルソースイメージのままデバッグが可能となり管理も容易になります。

- [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。
- 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] ^ [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。 例題プログラムは IBM Enterprise COBOL の方言を使用しているため、ここでは [Enterprise COBOL for z/OS] 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。
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	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 使用時の例:COBSQL プリプロセッサを使用します。】 P(COBSQL) ENDP



- 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 DBDEMO、[起動] には JCL を、[メインフレーム TN3270 ディスプレイの起動] はオフに指定します。



- 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 DBDEMO 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。

```
出力
出力元(S): ビルド
9:28 でビルドが開始されました...
----- ビルド開始: プロジェクト: DBDEMO, 構成: Debug x64 -----
* C:\¥$DBDEMO\TBLCRTE.cb1 のコンパイル中
* Generating C:\obj\¥$64\Debug\TBLCRTE
* Data: 188864 Code: 2482 Literals: 528
* C:\¥$DBDEMO\TBLDROP.cb1 のコンパイル中
* Generating C:\obj\¥$64\Debug\TBLDROP
* Data: 188864 Code: 2450 Literals: 448
* C:\¥$DBDEMO\TBLFETC.cb1 のコンパイル中
* Generating C:\obj\¥$64\Debug\TBLFETC
* Data: 188844 Code: 5594 Literals: 880
* C:\¥$DBDEMO\TBLSRT.cb1 のコンパイル中
* Generating C:\obj\¥$64\Debug\TBLSRT
* Data: 187440 Code: 4324 Literals: 880
COBOL コンパイル: 4 個 正常終了または最新の状態 0 個 失敗。
DBDEMO -> C:\¥$DBDEMO\obj\¥$64\Debug\
***** ビルド: 1 正常終了または最新の状態、0 失敗、0 スキップ *****
***** ビルドは 9:28 で完了し、01.262 秒掛かりました *****
```

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



3.6 XA スイッチモジュールの生成

ここで実行するプログラムは XA スイッチモジュール経由でデータベースと接続するため、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。本チュートリアルでは 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 プログラムからのデータベース連携を XA リソース方式で行う場合も同様の手順となります。

1) ビルドを行うため、製品フォルダに含まれている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。本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

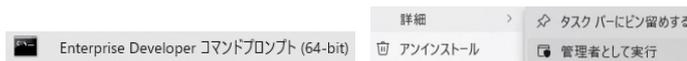
【コピー元フォルダ例】

C:¥Program Files (x86)\¥Rocket Software¥Enterprise Developer¥src¥enterpriseserver

【コピー先フォルダの例】 C:¥xa



2) Windows のプログラムメニューから [Rocket Enterprise Developer] > [ツール] > [Enterprise Developer コマンドプロンプト(64-bit)] を右クリックして [管理者として実行] を選択します。



3) コマンドプロンプトで、コピーした C:¥xa パスへ移動します。

```
C:\Users\¥arot¥Documents>cd c:¥xa
c:¥xa>
```

- 4) DLL を生成するために、Windows SDKが必要になります。リンクエラーを避けるために、これがインストールされているかご確認ください。また、複数の SDK や Microsoft Build Tools がインストールされている場合には、COBOL 環境が使用するバージョンを指定することもできます。

使用可能な SDK と Microsoft Build Tools の確認コマンド)

cblms -L

```
c:\%xa>cblms -L
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
10.0.0.82 (C) Copyright 1984-2024 Micro Focus or one of its affiliates.

Windows SDK
Id バージョン 場所
0] 10.0.19041.0 c:\Program Files (x86)\Windows Kits\10

Microsoft Build Tools
Id バージョン 場所
0] 14.32.31326 c:\Program Files\Microsoft Visual Studio\2022\Professional
```

最新バージョンに設定するコマンド)

cblms -U

```
c:\%xa>cblms -U
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
10.0.0.82 (C) Copyright 1984-2024 Micro Focus or one of its affiliates.

Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.19041.0

Microsoft Build Tools
場所 = c:\Program Files\Microsoft Visual Studio\2022\Professional
バージョン = 14.32.31326
```

特定のバージョンを指定するコマンド例):以降は -L で表示された番号を指定します。

Windows SDK を指定する場合)cblms -US:0

Build Tools を指定する場合)cblms -UB:0

```
c:\%xa>cblms -UB:0
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
10.0.0.82 (C) Copyright 1984-2024 Micro Focus or one of its affiliates.

Microsoft Build Tools
場所 = c:\Program Files\Microsoft Visual Studio\2022\Professional
バージョン = 14.32.31326
```

COBOL 環境が使用するバージョンを表示するコマンド)

cblms -Q

```
c:\%xa>cblms -Q
Micro Focus COBOL - Configuration Utility for the Microsoft Build Tools & SDK
10.0.0.82 (C) Copyright 1984-2024 Micro Focus or one of its affiliates.

Windows SDK
場所 = c:\Program Files (x86)\Windows Kits\10
バージョン = 10.0.19041.0

Microsoft Build Tools
場所 = c:\Program Files\Microsoft Visual Studio\2022\Professional
バージョン = 14.32.31326
```

利用可能なオプションを表示するコマンド)

cblms -H

- 5) 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。正常終了すると C:\%xa 配下に対象データベースの XA スイッチモジュールが作成されます。

① Oracle

コマンド)build ora19(対象バージョンにより ora21)

ディスク (C:) ▶ xa

名前



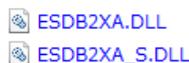
ファイル名.dll は静的登録用、ファイル名_D.dll は動的登録用です。

② Db2

コマンド)build db2

ディスク (C:) ▶ xa

名前



ファイル名_S.DLL は静的登録用、ファイル名.DLL は動的登録用です。

③ SQL Server

A) ビルドの実行

SQL Server に使用する ODBC 17.3 ドライバー以降、Microsoft 社は XA プロトコルを変更したため、ODBC17.3 以降を使用する場合は、mssql17 ビルドオプションを使用します。また、ご利用の Windows に合わせた SDK をインストールしておいてください。

本チュートリアルでは ODBC17.8 を利用するため mssql17 を使用してビルドします。

コマンド)build mssql17

ディスク (C:) ▶ xa

名前



ファイル名.dll は静的登録用、ファイル名_D.dll は動的登録用です。

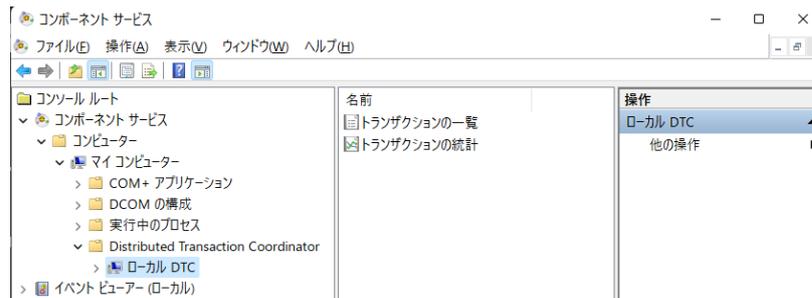
B) ODBC の追加

使用ビット数に合わせた ODBC データソースを Windows の [コントロールパネル] > [Windows ツール] > [ODBC データソース] から追加します。ここで指定する ODBC データソースの名前が Enterprise Server インスタンスへ登録する XA リソース定義の OPEN 文字列で使用する DSN 名となります。



C) XA トランザクションの有効化

Windows の [コントロールパネル] > [Windows 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] を展開します。



[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] のチェックがオンであることを確認、もしくはオンにして [OK] ボタンをクリックします。



XA スイッチモジュールのビルド詳細に関しては製品マニュアルをご参照ください。

3.7 文字エンコーディングの設定

Enterprise Server インスタンスを運用、管理する Enterprise Server Common Web Administration (以降 ESCWA) 機能では、スプールやデータ内容などに含まれる日本語を正しく表示させるために、事前に文字セットを所定のフォルダへ展開します。製品マニュアルの「リファレンス > コードセットの変換 > CCSID 変換テーブルのインストール > CCSID 変換テーブルをインストールするには」を参照しながら進めてください。

- 1) CCSID 変換テーブルをインストールします。

製品マニュアルにリンクされている下記の IBM CCSID 変換テーブルを、Web ブラウザから任意のフォルダへダウンロードします。アドレスは変更される可能性がありますので、製品マニュアルにてご確認ください。
<https://download.boulder.ibm.com/ibmdl/pub/software/dw/java/cdctables.zip>

- 2) 製品インストールフォルダ配下の etc フォルダに CCSID フォルダがない場合はこれを作成します。
例) C:\Program Files (x86)\Rocket Software\Enterprise Developer\etc\ccsid

- 3) ダウンロードファイルに含まれている Package2.zip を展開します。

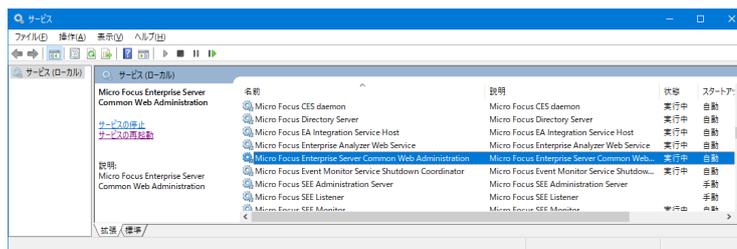
- 4) 展開した Package2 フォルダに含まれる IBM-932.zip を展開します。

- 5) 展開した IBM-932 フォルダを切り取り、作成した CCSID フォルダ配下へ貼り付け、14 ファイルが含まれていることを確認します。

名前	種類	サイズ
03A43480.MU-R-A2	MU-R-A2 ファイル	28 KB
03A43480.MU-R-A3	MU-R-A3 ファイル	28 KB
03A43480.MU-R-D	MU-R-D ファイル	28 KB
03A43480.PACKAGE	PACKAGE ファイル	5 KB
03A43480.TPMAP11A	TPMAP11A ファイル	329 KB
03A43480.TPMAP12A	TPMAP12A ファイル	329 KB
03A43480.TPMAP110	TPMAP110 ファイル	329 KB
03A43480.UPMAP12A	UPMAP12A ファイル	446 KB
03A43480.UPMAP13A	UPMAP13A ファイル	490 KB
03A43480.UPMAP120	UPMAP120 ファイル	447 KB
34B003A4.RPMAP12A	RPMAP12A ファイル	336 KB
34B003A4.RPMAP120	RPMAP120 ファイル	336 KB
34B003A4.UM-E-A21	UM-E-A21 ファイル	54 KB
34B003A4.UM-E-D12	UM-E-D12 ファイル	54 KB

詳細については、製品マニュアルの「デプロイ > 構成および管理 > Enterprise Server の構成および管理 > Enterprise Server Common Web Administration > [Native] > [Directory Server] > リージョンとサーバー > リージョン > エンタープライズ サーバー リージョンの文字エンコーディングのサポート」をご参照ください。

- 6) Windows サービスとして起動している Micro Focus Enterprise Server Common Web Administration を再起動し、インストールした CCSID をロードさせます。



3.8 Enterprise Server インスタンスの設定

Enterprise Server インスタンスには JES 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) 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。

2) Enterprise Server インスタンスを作成します。

Visual Studio の [サーバー エクスプローラー] タブの [Rocket Enterprise Server] を右クリックして [管理] 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。



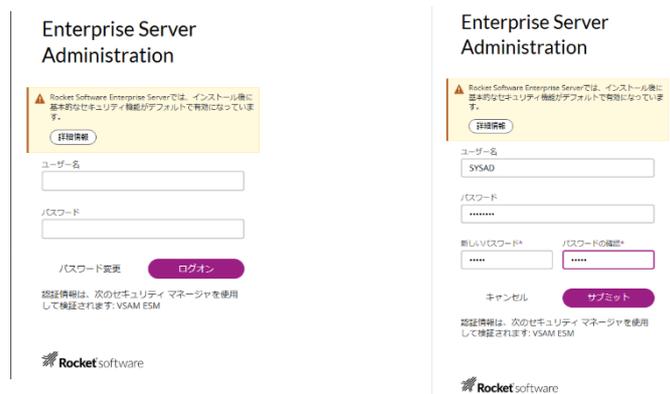
3) Web ブラウザが立ち上がり、ユーザー認証を求められます。

パスワードを変更してログオンします。

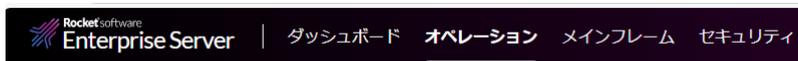
[パスワード変更] をクリックし、前項で取得したデフォルトユーザーと初期パスワード、新しいパスワードを入力して [サブミット] をクリックします。

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、ESCWA 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。



7) 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。



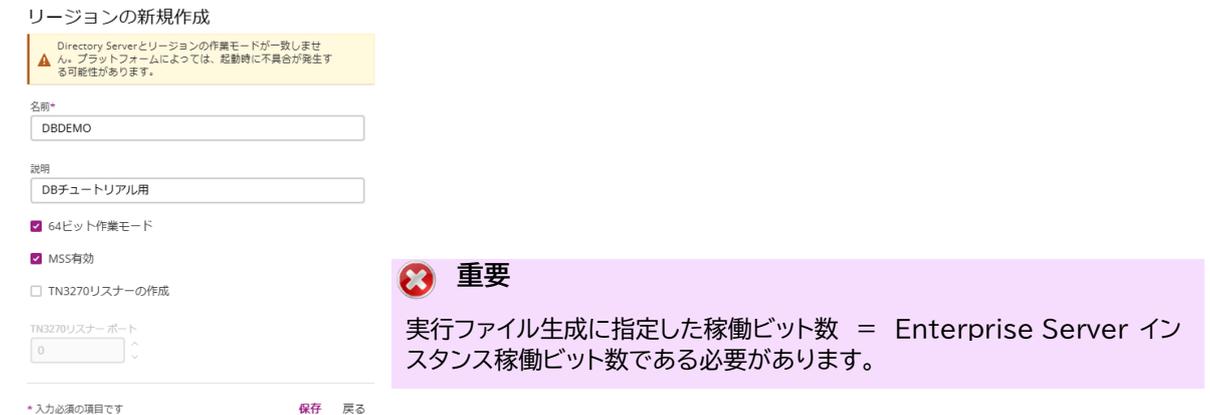
追加ウィンドウの [名前] は任意ですが、ここでは Default を、
[ホスト] は ホスト名またはマシンの IP アドレスを、
[ポート] は Micro Focus Directory Server の
デフォルトポートである 86 を入力して [保存] ボタンをクリックします。

4) 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。



5) [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 DBDEMO、説明に DB チュートリアル用と入力します。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] に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] にチェックが入っていることを確認し、[TN3270 リスナーの作成] のチェックを外して [保存] ボタンをクリックします。



6) 64 ビットアプリケーション稼働用の DBDEMO インスタンスが作成され、一覧に表示されます。DBDEMO 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...	ステータス	64ピ...	MSS有効	セキュ...	アクション
DBDEMO	Region	Stopped	✓	✓	デフォルト	  



- 7) DBDEMO 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 フォルダを指定して、[リージョンの機能] の [JES 有効] をチェックします。

開始オプション * 入力必須の項目です

名前* システムディレクトリ

リージョンの機能

MSS有効 JES有効

IMS有効 MQ有効

- 8) 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、Visual Studio からの動的デバッグが可能になります。

ローカルコンソールを表示 動的デバッグを許可

64ビット作業モード 以前のログを削除

- 9) [追加設定] の [構成情報] 欄に、文字エンコーディングを指定する MFACCCGI_CHARSET 環境変数に IBM-932 を認識させるための値である Shift_JIS と、プロジェクトのパスを指定する環境変数を設定し、最後に [適用] ボタンをクリックします。

入力値)

[ES-Environment]

proj=C:\VS\DBDEMO
MFACCCGI_CHARSET=Shift_JIS

追加設定

構成情報

```
[ES-Environment]
proj=C:\VS\DBDEMO
MFACCCGI_CHARSET=Shift_JIS
```

- 10) 画面上部の [JES] 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 proj 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
JES プログラム パス	COBOL アプリケーションの実行ファイルが存在するパスを指定します。
システムカタログ	カタログファイルを出力するパスと、そのファイル名称を指定します。
データセットの省略時ロケーション	ジョブ実行時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デフォルトパスを指定します。

JESの構成 適用

JES プログラムパス システムカタログ

データセットの省略時ロケーション システムプロシージャライブラリ

Fileshare 構成ロケーション



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。

11) [イニシエータ] の [新規作成] ボタンをクリックします。



12) 下記画面のように入力して [保存] ボタンをクリックします。この指定により DBDEMO インスタンスが開始時にイニシエータが稼働し、ジョブクラス A,B,C のジョブが実行可能になります。

JESイニシエータ

名前*

クラス

説明

* 入力必須の項目です 保存 戻る

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。

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

重要

バージョン 7.0 では、パフォーマンス向上の観点から JES 関連ファイルである SPLJOB.DAT のフォーマットが改善されています。そのため、旧バージョンのファイルを 7.0 以降で利用する場合は mfsplcnv コマンドを使用して新フォーマットにコンバートする必要があります。コンバートを実行すると、古いフォーマットのファイルは SPLJOB.bak として保存されます。

対象ファイルの特定には MFSYSCAT 環境変数を利用して、カタログファイルを指定します。

例)

```
set MFSYSCAT=C:¥VS¥DBDEMO¥DATAFILE¥catalog.dat
mfsplcnv -2
```

詳しくは製品マニュアルをご参照ください。

14) 前項で作成した XA リソースを登録します。画面上部の [一般] プルダウンメニューから [XA リソース] を選択し、表示される画面で [新規作成] ボタンをクリックします。

項目名	説明
ID	プログラムや JCL の IKJEFT ユーティリティに渡す DSN TSO コマンドの SYSTEM パラメータへ指定する ID を指定します。ここでは XADB を指定します。 ID <input type="text" value="XADB"/>
名前	XA リソース名として任意の名前を指定します。Oracle の場合は Oracle_XA 固定です。 名前 <input type="text" value="ORACLE_XA"/>

モジュール	<p>前項で作成した XA スイッチモジュールのパスとファイル名を指定します。</p> <p>【Oracle 使用時の例】 動的登録の C:\xa\ESORAXA_D.dll を入力します。 モジュール名 <u>C:\xa\ESORAXA_D.dll</u></p> <p>【Db2 使用時の例】 動的登録の C:\xa\ESDB2XA.dll を入力します。 モジュール名 <u>c:\xa\ESDB2XA.DLL</u></p> <p>【SQL Server 使用時の例】 動的登録の C:\xa\ESMSSQL_D.dll を入力します。 モジュール名 <u>c:\xa\ESMSSQL_D.dll</u></p>
OPEN 文字列	<p>対象データベースのオープン文字列を指定します。</p> <p>【Oracle 使用時の例】 ORACLE_XA+SesTm=100+SqlNet=orcl-19c+Acc=P/scott/tiger を入力します。</p> <p>【Db2 使用時の例】 DB=DEMO DB,uid=tarot,pwd=password,AXLIB=casaxlib.dll を入力します。</p> <p>Windows の場合)AXLIB に指定するモジュールのパスが PATH 環境変数に指定されていない場合はフルパスを指定してください。ビット数により bin64, bin のパスが異なりますのでご注意ください。ファイル拡張子は省略可能です。 64 ビットフルパスの例: %COBDIR%\bin64\CASAXLIB.dll</p> <p>静的登録の場合は末尾に SREG=T を指定します。</p> <p>【SQL Server 使用時の例】 DSN=SQLSVR を入力します。(=ODBC 名)</p>
有効	有効、無効切り替えチェックを指定します。ここではオンを指定します。

XA リソースの構成

ID* 名前*

モジュール*

有効

OPEN 文字列

CLOSE 文字列

説明

*入力必須の項目です 保存 戻る

15) 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] 内の [Rocket Enterprise Server] を展開します。ESCWA 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) [サーバーエクスプローラー] 内に表示された [Default] を展開すると、再度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。



- 3) DBDEMO 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 [Rocket Enterprise 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。
- 4) [サーバー エクスプローラー] 内の DBDEMO 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] > [DBDEMO] を選択します。これにより DBDEMO プロジェクトから実行されるアプリケーションは DBDEMO インスタンスで処理されることになります。



- 5) [サーバーエクスプローラー] 内の DBDEMO インスタンスを右クリックし、[開始] を選択して開始します。

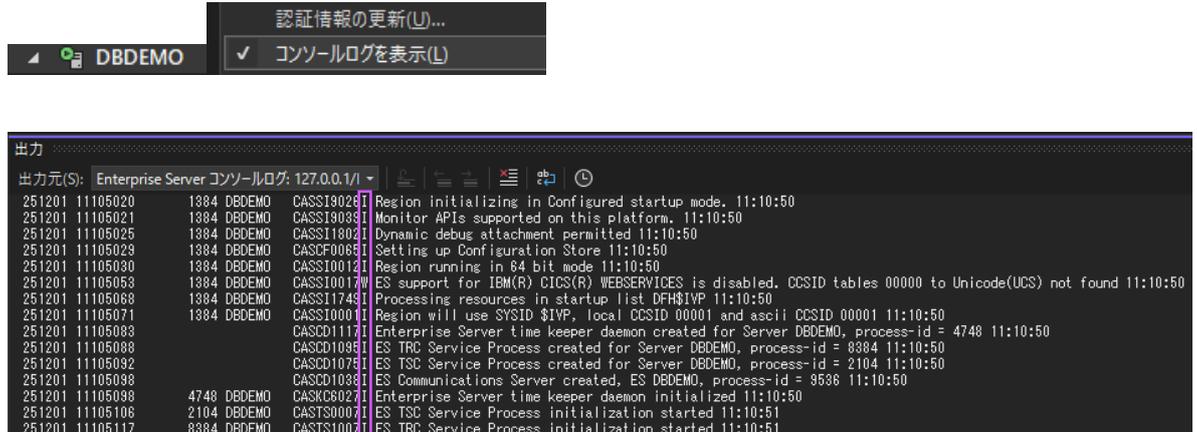


- 6) 開始ユーザーの認証ウィンドウが表示されますので、前述で指定した SYSAD ユーザーとそのパスワードを指定して [OK] ボタンをクリックします。



- 開始された DBDEMO インスタンスを右クリックし、[コンソールログを表示] を選択して、[出力] タブの開始ログを確認します。

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。



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。

- XA モジュールが正常にロードされ、オープンされると以下のようなログが出力されます。下記は SQL Server の例ですが、動的登録を指示しているため Dynamic と出力されています。静的登録の場合は Static が出力されます。

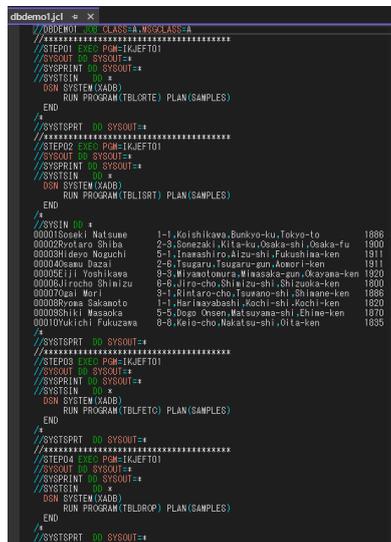


- 画面左側ペインの [Default] をクリックして一覧画面に戻ります。

3.10 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

現在 DBDEMO 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。プログラム内容を各自ご確認いただき、JCL を実行します。

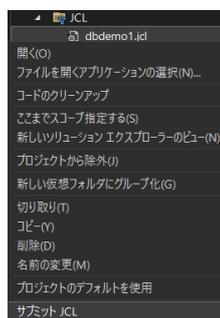
- Visual Studio の [ソリューション エクスプローラー] 内にある DBDEMO プロジェクト配下の dbdemo1.jcl をダブルクリックし、エディタで内容を確認します。



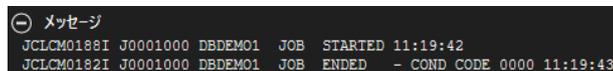
この JCL は 4 ステップから構成されています。

- ① STEP01
データベーステーブルを新規作成します。
- ② STEP02
SYSIN データを作成したテーブルへ挿入します。
- ③ STEP03
挿入したデータをテーブルから全件読み込み、SYSOUT へ出力します。
- ④ STEP04
作成したテーブルをデータベースから削除します。

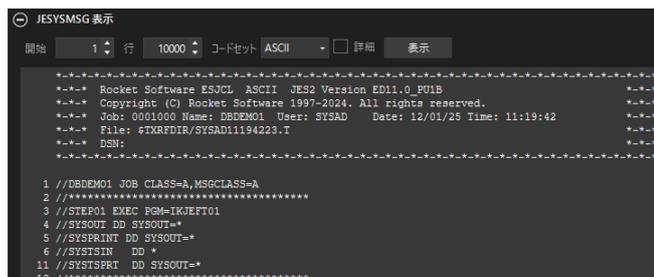
- 2) [ソリューション エクスプローラー] 内の dbdemo1.jcl を右クリックして [サブミット JCL] を選択し、この JCL を実行します。



- 3) [DBDEMO スプール] タブが表示されますので、[メッセージ] で COND CODE を確認します。



- 4) [JESYSMSG 表示] を展開すると各ステップの内容が確認できます。



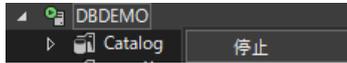
- 5) STEP03 の SYSOUT を選択し、[SYSOUT 表示] を表示すると、作成したテーブルからインサート済データが正常に FETCH できていることが確認できます。



6) JCL チュートリアルに記載しているように、バッチプログラムのデバッグも可能です。

3.11 Enterprise Server インスタンスの停止

1) [サーバーエクスプローラー] 内の DBDEMO インスタンスを右クリックし、[停止] を選択して停止します。



2) DBDEMO インスタンスの停止を確認後、Visual Studio を終了します。

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。