Micro Focus 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 2017 編

1. 目的

本チュートリアルでは、Visual Studio 2017 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 10 Enterprise
- 使用マシンに Microsoft Visual Studio 2017 がインストールされていること
- 使用マシンに Micro Focus Enterprise Developer 4.0 for Visual Studio 2017 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
- 使用したデータベースは以下の通り
 - ➢ Oracle サーバー : Oracle 10g 10.2.0.1.0 64bit

クライアント: Oracle クライアント 11g 11.2.0.4.0 64bit (Pro*COBOL プリコンパイラ使用)

▶ DB2 サーバー: DB2 11.1.0.1527 64bit

クライアント: DB2 クライアント 9.7.900.250 64bit (DB2 PREP プリコンパイラ使用)

> SQL Server サーバー : SQL Server 2012 R1 64bit

クライアント: ODBC ドライバー SQL Server Native Client 11.0

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. 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. 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
- 10. Enterprise Server インスタンスの停止



3.1 チュートリアルの準備

例題プログラムに関連する資源を用意します。

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。

```
-カル ディスク (C:) ▶
名前
```

2) Visual Studio のソリューションを保存する VS フォルダーを C:¥ 直下に作成します。

3.2 Visual Studio の起動

1) Visual Studio 2017 を起動します。

Visual Studio 2017

3.3 メインフレーム COBOL 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、下記項目を指定後 [OK] ボタンをクリックします。

項目名	説明
左側ツリービュー	[インストール済] > [テンプレート] > [COBOL プロジェクト] > [Native] を選択します。
中央リスト	[メインフレーム サブシステム アプリケーション] を選択します。
名前	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。
ソリューションのディレクトリ ーを作成	ここではチェックをオフにします。

新しいプロジェクト								
▶ 最近使用したファイル		並べ替え	; 既定	· # 🗉		検索 (Ctrl+E)		ρ-
▲ インストール済み		CBL	Enterprise Server アプリケー	シ…COBOL プロジェクト	Î	種類: COBOL プロジェクト		
▲ COBOL プロジェクト Managed		CBL	Windows アプリケーション	COBOL プロジェクト	J	Micro Focus Server 配下で実行され CICS®、JCL または IMS® アプリケーシ 成するためのプロジェクトです。	行される リケーション	。 コンを作
■ Native ▷ PL/I ▷ その他のプロジェクト	の種類		メインフレーム サブシステム ア	プ COBOL プロジェクト				
▶ オンライン		C:\	コンソール アプリケーション	COBOL プロジェクト	1			
			リンク ライブラリ	COBOL プロジェクト				
			空のプロジェクト	COBOL プロジェクト				
採しているものかり Visual Studio イ	もつからない場合 ンストーラーを開く	CBL	Micro Focus INT/GNT	COBOL プロジェクト	Ŧ			
名前(<u>N</u>):	DBDEMO							
場所(<u>L</u>):	C:¥VS¥			•		参照(<u>B</u>)		
ソリューション名(<u>M</u>):	DBDEMO					ソリューションのディレクトリを作!	戊(<u>D</u>)	
						ソース管理に追加(U)		
						ОК	キャン	ยม

2) [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。



メインフレーム COBOL 開発:データベース連携 Visual Studio 2017 編



3) プロジェクトを作成したことにより C:¥VS¥DBDEMO フォルダーが作成されています。このフォルダー配下に JES 機能で 使用するフォルダー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 DATAFILE フォルダーを C:¥VS¥DBDEMO 配下へ作成します。 (C:) 、 VS 、 DBDEMO 、



4) 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] プロダウンメニューの [オプション] を選択してオプションウィンドウを表示します。

左側ツリービューの [Micro Focus] > [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。



5) 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ーへ例題プログラムが配置されます。

🔀 既存項目の追加 - DBD	DEMO				×	<
← → • ↑ 📑 >	PC → ローカル ディス	ク(C:) > DBtutorial	ٽ ~	DBtutorialの検索	م	
整理 ▼ 新しいフォル	ダー				- 🔳 🔞	
a OneDrive	^	名前	● 更新日時	種類	サイズ	
🤱 tarot		at dbdemo1.jcl	2013/09/19 16:57	JCL ソースコード	2	KB
PC		TBLCRTE.cbl	2013/09/20 12:02	CBL ファイル	1	KE
🧊 3D オブジェクト		🛅 TBLDRORcbl	2013/09/20 12:02	CBL ファイル	1	KE
🖊 ダウンロード		TBLFETC.cbl	2013/10/07 15:06	CBL ファイル	2	KE
デスクトップ		TBLISRT.cbl	2013/10/07 15:06	CBL ファイル	2	KE
🏥 ドキュメント	~	<				>
77	ッイル名(<u>N</u>): "TBLISR	T.cbl" "dbdemo1.jcl" "TBl	.CRTE.cbl" "TBLDRORcbl" "TBLFETC.cbl" 🗸	すべてのファイル (*.*)	~	
				追加(<u>A</u>) 🔽	キャンセル	



6) [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。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でデバッグが可能とな り管理も容易になります。ここでは後者の方法を紹介します。

- 1) [ソリューション エクスプローラー] 内の [Properties] をダブルクリックしてプロパティウィンドウを表示します。
- 左側ツリービューの [アプリケーション] を選択して、生成する実行ファイルを GNT にするため [出力の種類] へ [INT/GNT] を選択します。

アプリケーション	構成(C): なし	● 情報
SQL		
従属パス		GNT は Micro Focus 独自のオブジェクトで、Micro Focus
BMS	出力の名前:	COBOL ランタイム環境下で実行可能とかります
IMS	DBDEMO	この日のビ リンノーム境況 「 こ天日 可能になります。
プリプロセッサ	田力の種類: INT/GNT	

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 言語方言を指定します。
COBOL 方言	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。
出カパス	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。



	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:COBSQL プリプロセッサを使用します。】
	P(COBSQL) ENDP
	追加指令: P(COBSQL) ENDP
追加指令	【DB2 使用時:ECM プリプロセッサを使用します。】
	DB2(DB=SAMPLE,VALIDATE=RUN)
	追加指令:
	DB2(DB=SAMPLE,VALIDATE=RUN)
	【SQL Server 使用時 : OpenESQL を使用します。】
	SQL(DBMAN=ODBC,BEHAVIOR=JCL,TARGETDB=MSSQLSERVER)
	追加指令:
	SQL(DBMAN=ODBC,BEHAVIOR=JCL,TARGETDB=MSSQLSERVER)
アプリケーション SQL	✓ ブラットフォーム(M): アクラィブ (×86)

SQL					
従属パス	全般				
BMS IMS プリプロセッサ	プラットフォームターグット: COBOL 方言:	x64 V Enterprise COBOL for 2	z/05 ¥	文字セット:	ASCII V
COBOL	ソース フォーマット:	固定	¥		
COBOL リンク アセンブラ	デバッグ用にコンパイル		✓ .GNT にコンパイ	JU	
デバッグ	ビルド イベント				
Micro Focus Code Analysis	メインフレーム				
	EXEC CICS	オプション	EXEC DLI	オプション	
	エラーおよび警告				
	警告レベル:	回復可能なエラーを含める(L/VILE)	~	
	最大エラー数:	100	□ 警告をエラーとし	て処理	
	出力				
	出カパス:	.¥bin¥x86¥Debug¥			参照
	□ 指令ファイルの生成		🗌 リストファイルを生	成	
	□ コード カバレッジを有効に	1 98	🗌 プロファイラを有効	かにする	
	追加指令 —				
	DB2(DB=SAMPLE,VAL	IDATE=RUN)			^

4) 左側ツリービューの [デバッグ] を選択して、[アクティブ設定を選択] へ [JCL] を選択します。

アクティブ設定を選択:	JCL	~
-------------	-----	---

5) プロパティファイルを上書き保存します。

ファイル(F)	編集(E)	表示(V)	
0 - O	ት - 省	•	



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。					
出力元の表示(S):	ビルド				• <u> </u>
* Generating .	¥bin¥x8	Sidon opinion S¥Debug¥TBLFE	ETC		
* Data:	3272	Code:	6275	Literals:	720
* C:#VS#DBDEMU * Generating .	J¥IBLISK ,¥bin¥x8I	i.coi のコン。 S¥Debug¥TBLIS	SRT		
* Data:	3776	Code:	5317	Literals:	768
	ル:418	町正常終了ま)	には最新の利	天態 U 1固 失敗。	
======================================	てリビル	ド:1 正常終	了、0 失敗、	. 0 スキップ ===	

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) プリコンパイルを行うため、製品フォルダーに含まれている下記フォルダーを書き込み権限があるフォルダー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】製品関連フォルダーの書き込み権限によるトラブルを避ける

【コピー元フォルダー例】

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

【コピー先フォルダーの例】 C:¥xa

Enterprise Dev	eloper → src →	enterprises erver \rightarrow		ーカル ラ	[£] ィスク (C:) →
^	名前	~		^	名前
	🎍 exits] ⊐Ľ– →		鷆 xa



- 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¥tarot¥Documents>cd	c:¥xa
c:¥xa>	

- 4) 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 ora11 (対象バージョンにより ora12)

c:¥xa>build ora11 Building 64-bit switch module Micro Focus COBOL Version 4.0 Copyright (C) Micro Focus 1984-2018. All rights reserved.	
* Cobsql Integrated Preprocessor * CSOL-Lally・Drasla ブリコンパイニトニンフレータズ提動します	ディスク (C:) → xa
* CSQL-100: Oracle ブリコンバイラの出力を処理中。 * CSQL-1-001: COBSQL:チェッカへの引き渡しを完了しました。	名前
* 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 * Dete: 15052 Code: 62500 Literale: 2076	SORAXA.dll
Micro Focus COBOL - CELLINK utility Version 4.0.0.79 Copyright (C) Micro Focus 1984-2018. All rights reserved. ————————————————————————————————————	SORAXA_D.dll

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。

DB2

コマンド) build db2

c:¥xa≻build db2		
Building 64-bit switch module		
Micro Focus COBOL		= 17 h (C) > > >
Version 4.0 Copyright (C) Micro Focus 1984-2018. All rights reserved.		71X9(C;) ⊫ xa
* 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		A7
*Data: 21408 Code: 46154 Literals: 2768		治則
Micro Focus COBOL - CBLLINK utility		
Version 4.0.0.79 Copyright (C) Micro Focus 1984-2018. All rights reserved.		
		S ESDB2XA.DLL
Microsoft (R) Incremental Linker Version 14.12.25830.2		
Copyright (C) Microsoft Corporation. All rights reserved.	\rightarrow	ESDB2XA_S.DLL

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

コマンド) build mssql

c:¥xa>build mssql Buildine 6d-bit switch module		
Micro Focus COBOL Version 4.0 Copyright (C) Micro Focus 1984-2018. All rights reserved.		イスク (C:) ト xa
*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: 51427 Literals: 9538		名前
Micro Focus COBOL - CBLLINK utility Version 4.0.0.79 Copyright (C) Micro Focus 1984-2018. All rights reserved.		RESINGS OF ALL
Microsoft (R) Incremental Linker Version 14.12.25830.2 Copyright (C) Microsoft Corporation. All rights reserved.	\rightarrow	ESMSSQL_D.dll

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) リンクエラー時

環境変数「LIB」へ下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。 xaSwitch.Lib を利用するためには Windows SDK が必要になります。 32 ビット例: C:¥Program Files (x86)¥Windows Kits¥8.0¥Lib¥win8¥um¥x**86**

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

	システム変数の編集
変数名(<u>N</u>):	LIB
変数値(⊻):	¥Windows Kits¥8.0¥Lib¥win8¥um¥x64;C:¥
	OK キャンセル

ESMSSQL.obj cbllds00015740.obj	
Creating library ESMSSQL.lib and object ESMSSQL.exp	
Copyright (c) Microsoft Corporation 2012.	
All rights reserved.	
1 個のファイルをコビーしました。	
Micro Focus COBOL	
Version 2.3.00351 Copyright (C) Micro Focus 1984-2015. 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_D	
*Data: 16 Code: 55176 Literals:	3536
Micro Focus COBOL - CBLLINK utility	
Version 2.3.0.95 Copyright (C) Micro Focus 1984-2015. All	rights reserved.



C) 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] > [管理ツール] > [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 インスタンスへ設定する XA リソース定義の OPEN 文字列で使用する DSN 名となります。

ユーザー DSN システム DSN ファイル DSN ドライバー トレース 接続

システム データ ソース(<u>S</u>):			
	名前	プラットフォーム	ห็ว่า//-
	SQLSVR	64 ビット	ODBC Driver 13 for SQL Server

D) XA トランザクションの有効化

Windows の [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] を展開します。

(e).	コンボーネント サービス
シランション ション・ション・ション・ション・ション・ション・ション・ション・ション・ション・	ブ(且)
	名前 日 トランザウションの一覧 M トランザクションの統計

[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。

ローカル DTCのプロパティ ? 💻
レース中 ログ セキュリティ
セキュリティの設定
- ソフ1 // ノ「わよい目荘 ■ リモート クライアントを許可する(R) ■ リモート管理を許可する(M)
 ⑦ 受信を許可する(I) ⑦ 送信を許可する(W)
 相互認証を必要とする(L)
○ 着信呼び出し側には認証を必要とする(G)
○ 認証を必要としない(工)
✓ XA トランザクションを有効にする(X) ✓ SNA LU 6.2 トランザクションを有効にする(E)

XA スイッチモジュールのビルド詳細に関しては製品ヘルプをご参照ください。



3.7 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) Enterprise Server インスタンスを作成します。 [サーバー エクスプローラー] タブの [Micro Focus Server] を右クリ ックして [管理] を選択します。



2) Enterprise Server Administration 画面に Enterprise Server インスタンス一覧が表示されますので、画面の左 下にある [追加] ボタンをクリックします。

編集	MFES (MSS)	STAFF64	<u>停止</u> 開始…
追加			

3) サーバー名には [DBDEMO] を入力、動作モードは 64-bit を指定して [次へ] ボタンをクリックします。

サ <i>ー</i> バー追加 (Page 1 of 3):
サーバー名: DBDEMO
動作モード:
○ 32-bit
You cannot change your choice of work
キャンセル 次へ >>

() 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 = XA リソースビット数 =
データベーフクライアント対応ビット数 であろいの更があります

 画面の Page 2/3 では、CICS や JCL を実行可能な機能を持つ [Micro Focus Enterprise Server with Mainframe Subsystem Support] が選択されていることを確認後、[次へ] ボタンをクリックします。

サーバ	・ータイナ	h.
0	MFES	Micro Focus Enterprise Server An enterprise server that provides an execution environment for COBOL application programs running as services in a service orientated architecture.
۲	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.



5) Page 3/3 では [TN3270 リスナーの作成] のチェックをオフにして [追加] ボタンをクリックすると、[DBDEMO] という 名前の 64 ビットアプリケーション稼働用 Enterprise Server インスタンスが追加されます。

生成オプション:		編集	MFES (MSS)	DBDEMO	<u>停止</u> 開始…
TN3270リスナーの作成 🗌 using port	\rightarrow		64 9		

6) 左にある [編集] ボタンをクリックします。

編集	MFES (MSS) 64	DBDEMO	停止 開始	
----	---------------------	--------	----------	--

- 7) [サーバー] > [プロパティ] > [一般] タブ内の下記項目を設定します。
 - ① [動的デバッグを許可] チェックボックスをオンにします。この指定により、Visual Studio からの動的デバッグが可能 になります。

開始オブション:		
共有メモリページ数:	512	サービス実行プロセス: 2
共有メモリクッション:	32	要求ライセンス: 10
ローカルコンソールを表示:		動的デバッグを許可: 🔽
Start on System Start:	6	4-Bit Working Mode: 🗹
以前のログを削除:	עב 🗆	ソールログサイズ (K): 0

- ② [適用] ボタンをクリックします。
- 8) [サーバー] > [プロパティ] > [MSS] > [JES] タブで表示される画面の各項目を設定します。入力後は [Apply] ボタ ンをクリックします。

項目名	説明
メインフレーム サブシステム サポート有効	[MSS] タブ配下の設定をオン、オフ指定します。ここではオンに指定します。
ジョブ入力サブシステム 有効	[JES] タブ配下の設定をオン、オフ指定します。ここではオンに指定します。
JES プログラム パス	COBOL アプリケーション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	カタログファイルが存在するパスと、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。
システムプロシージャライブラリ	プロシージャライブラリの名前を指定します。 ここでは SYS1.PROCLIB を入力します。



メインフレーム サブシステム サポート有効: 🔽

CICS (✔) JES (✔) IMS PL/I						
一般 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:\VS\DBDEMO\bin\x86\Debug						
C:\VS\DBDEMO\DATAFILE\catalog.dat						
データセットの省略時ロケーション:						
C:\VS\DBDEMO\DATAFILE						
システムフロシージャライフラリ: SYS1.PROCLIB						
Fileshare 構成 ロケーション:						



入力値は全て半角英数字で指定してください。

これらのフィールドでは改行を入れないように注意してください。

9) [サーバー] > [プロパティ] > [MSS] > [JES] > [Initiators] タブを表示し、左下の [追加] ボタンをクリックします。

CICS (✔)	JES (✔)	IMS	PL/I
一般	nitiators (0)	Printers (0)	
追加			

10) 下記画面のように入力して [追加] 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。



11) 前項で作成した XA リソースを定義します。[サーバー] > [プロパティ] > [XA リソース] タブを表示して、左下の[追加] ボタンをクリックします。



12) 前項で作成した、使用するデータベース製品のXA リソースを設定します。下記項目を入力後 [追加] ボタンをクリックしま す。

項目名	説明
	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 パラメタヘ指
	定する ID を指定します。
ID	ここでは XADB を指定します。
	XΔ IIV-7名として任音の名前を指定します
	Oracle は Oracle XA 固定です。
名前	名前:
	Oracle_XA
	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。
	【Oracle 使用時の例】
	動的登録の C:¥xa¥ESORAXA_D.dll を入力します。
	C:\xa\ESORAXA_D.dll
モジュール	【DB2 使用時の例】
	動的登録の C:¥xa¥ESDB2XA.dll を入力します。
	【SQL Server 使用時の例】
	動的登録の C:¥xa¥ESMSSQL_D.dll を入力します。
	C:\xa\ESMSSQL_D.dll
OPEN 文字列	対象データベースのオープン文字列を指定します。
	【Oracle 使用時の例】
	Oracle_XA+SesTm=100+SqlNet=tok-par+Acc=P/scott/tiger を入力します。
	OPENX 4 % In the set of the set o
	ondele_withdestina toologineer tok partitiee a viseotatiger
	【DB2 使用時の例】
	DB=SAMPLE,uid=db2inst1,pwd=ibmdb2,AXLIB=casaxlib を入力します。
	静的登録の場合は末尾に SREG=T を指定します。デフォルトは動的です。
	OPEN文字列: DB=SAMDLE_uid=dh2inst1_mud=ibmdh2_AYLIB=sessaylib
	【SQL Server 使用時の例】
	DSN=SQLSVR を入力します。(=ODBC 名)
	OPEN文字列:
月幼	1月301、無効切り増スナエッツを指正します。 ここではオンを指正します。

メインフレーム COBOL 開発: データベース連携 Visual Studio 2017 編



ID: XADB 名前: Oracle_XA	D: XADB 名前: DB2_XA	ID: XADB 名前: SQLSVR_XA
モジュール:	モジュール: 	モジュール:
C:\xa\ESORAXA_D.dll	C:\xa\ESDB2XA.DLL	C:\xa\ESMSSQL_D.dll
 OPEN 文字列:	OPEN文字列:	OPEN文字列:
Oracle_XA+SesTm=100+SqlNet=tok-par+Acc=P/scott/tiger	DB=SAMPLE,uid=db2inst1,pwd=ibmdb2,AXLIB=casaxlib	DSN=SQLSVR
CLOSE 文字列:	CLOSE文字列:	CLOSE文字列:
L	L	
有効: 🔽	有効: 🕑	有効: 🔽

13) 画面左上の [Home] をクリックして一覧画面に戻ります。



3.8 Enterprise Server インスタンスの開始と確認

- 1) [サーバー エクスプローラー] 内に 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 [Micro Focus Server] を右クリックし、[最新の情報に更新] を選択してリフレッシュしてください。
- [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。

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



4) 下記ウィンドウが表示された場合は、ここではユーザーによる制限を行わないため [OK] ボタンをクリックします。

Enterprise Server サイ	יעזע ×
サーバーの接続詳細を	:入力: DBDEMO
🗌 セキュア サーバー	
ユーザー名:	
パスワード:	
グループ:	既定グループの場合は空白
☑ 認証の保存	
	OK キャンセル



5) Enterprise Server Administration 画面へ移動して開始状態であることを確認後、[詳細] ボタンをクリックします。



6) [サーバー] > [診断] > [ES コンソール] で [DBDEMO] インスタンスのコンソールログをリアルタイムにチェックすることがで きます。また [Show Entire Log] をクリックしてログ全体を表示させることも可能です。

正常に開始されたことを確認します。

トレース	ダンプ 】 <mark>ESコンソール</mark> CSコンソール	
<u>ا</u> ف	更新 ○ Show entries from 1 to 10 of 55 total entries ● Show last 10 lines	
Entry	Event	Show Entire Log
46	151112 15081195 123560 DBDEMO CASX00015I XADB2 XA interface initialized successfully 15:08:11	
47	151112 15081216 123560 DBDEMO CASXO0020I XASQL XA interface loaded. Name(Micro Focus SQL Server XA switch), Re	egistration Mode(Static) 15:06:12
48	151112 15081251 123580 DBDEMO CASXO0021I XASQL XA interface local transaction support enabled 15:08:12	
49	151112 15081342 124044 DBDEMO CASCS5100I Communications Process instance 01 is ready to accept requests 15:08:1	3
	151110 15081570 102580 DRDENO CARVODDEL VAROL VA interfere initialized successfully 15:08-15	
00	151112 15001572 123500 DBDEMO CASAGOUTSI ARGOL AR Interface Initialized Subsestibility 15:00.15	
50	151112 15061677 123560 DBDEMO CASSI50011 PLTPI Phase 1 - No PLT Specified 15:06:16	
50 51 52	10112 100015/2 12500 DEDEBID CHRADOVID AGAL An Interface minutes soccessing 1500,15 15112 1000167 12580 DEDEMIC CASSIOUT LPTP Phase 1 - No PLT Specified 1500;16 151121 25061671 123580 DEDEMIC CASSIOUT Active SEP memory strategy set to x00000001, retain count 100 15:06:16	1
50 51 52 53	151112 1506157 12550 DBDEMO CASSIG001 PASEL XA interface minated subcessi unit 100.15 151112 15061657 125560 DBDEMO CASSIS0011 PLTPI Pase 1 - No PLT Specified 15.06:16 151112 1506171 12580 DBDEMO CASSIS001 Active SEP memory strategy set to x000000011, retain count 100 15:06:16	
50 51 52 53 54	10112 1000157 12300 DBCBM CASSI0010 ASL An imitatos minatos subassi subassi un 100.10 10112 1000157 12360 DBCBM CASSI0011 FLPT Phase 1. No FLT Specified 1500:10 15112 1000157 12360 DBCBM CASSI0401 Active SEP memory strategy set to X0000001', retain count 100 15:00:16 15112 1000170 121310 DBCBM CASSI0001 Bach initiator started for job classes 'ABC' 15:00:17 151112 1000170 124784 DBCBM CASSI0001 Bit hinitiator started for job classes 'ABC' 15:00:17	

いくつかのサービス開始や XA モジュールのロード、オープン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。

7) 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 Oracle 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。

123560 DBDEMO	CASXC00201	XADB XA	interface	loaded. Name(Oracle_XA), Registration Mode(Dynamic) 15:06:	07
123560 DBDEMO	CASXCOO21I	XADB XA	interface	local transaction support enabled 15:06:08	
123560 DBDEMO	CASXC0015I	XADB XA	interface	initialized successfully 15:06:09	



8) 画面左上の [Home] をクリックして一覧画面に戻ります。

3.9 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。まずは簡単な JCL を実行して みます。

1) [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

データベースへテーブルを新規作成します。

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。

			👌 dbdemo1.jcl
¢	關<(0)		
	ファイルを開くアプリケーションの選択(N)		
	ここまで検索(S)		
Ē	新しい ソリューション エクスプローラー のビュー(N)		
	プロジェクトから除外(J)		
	新しい仮想フォルダにグループ化(G)		
ж	切り取り(工)	Ctrl+X	
ŋ	⊐ピ-(Y)	Ctrl+C	
\mathbf{X}	削除(D)	Del	
X:::	名前の変更(M)		
	サブミット JCL		



3) [出力] タブの [出力元] に [Enterprise Server] を選択すると JOB 実行ログと JOB 番号が表示されますので、コ ントロールキーを押しながらリンクをクリックします。



4) この JOB 番号にかかわるスプール一覧が表示されます。先頭の [JESYSMEG] をクリックしてジョブログを確認します。

J0001001	Name	DBDEMO1	Status:	Complete			
Hold	Clas	s: A	Priority	00			
Update	Use	JESUSER	COND	0000			
Delete	File	s \$TXRFDIR/t0	000000022.t				
UCLCM0188I J0001001 DBDEM01 JOB STARTED 14:47:05 JCLCM0182I J0001001 DBDEM01 JOB ENDED - COND CODE 0000 14:47:07							
	Status	Class DD Name	Step	Nbr.			
Details	Hold	A JESYSM	ISG	0			

5) ジョブログの内容を確認すると、この JOB が正常に終了していることが確認できます。

JCLCM0182I JOB ENDED - COND CODE 0000

6) [web ブラウザー 戻る] アイコンをクリックし、スプール一覧から各ステップの出力内容を確認することができます。

たとえば、STEP03 の SYSOUT 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。

	FETCH:	00001,Soseki Natsume	,1-1,Koishikawa,Bunkyo-ku,Tokyo-to	,1886
	FETCH:	00002,Ryotaro Shiba	,2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu	,1900
	FETCH:	00003,Hideyo Noguchi	,5-1,Inawashiro,Aizu-shi,Fukushima-ken	,1911
	FETCH:	00004,Osamu Dazai	,2-6,Tsugaru,Tsugaru-gun,Aomori-ken	,1911
	FETCH:	00005,Eiji Yoshikawa	,9-3,Miyamotomura,Mimasaka-gun,Okayama-ken	,1920
	FETCH:	00006,Jirocho Shimizu	,6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken	,1800
	FETCH:	00007,Ogai Mori	,3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken	,1886
8 🦱 👝 v 🔊 s	FETCH:	00008,Ryoma Sakamoto	,1-1,Harimayabashi,Kochi-shi.Kochi-ken	,1820
ି 🖸 🖉 🖉	FETCH:	00009,Shiki Masaoka	,5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken	,1870
	FETCH:	00010,Yukichi Fukuzawa	,8-8,Keio-cho,Nakatsu-shi,Oita-ken	,1835
Neb フラウサー 戻る	→ FETCH:	**END OF JOB**		

7) 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。



3.10 Enterprise Server インスタンスの停止

1) DBDEMO インスタンスを停止します。



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



WHAT'S NEXT

- メインフレーム COBOL 開発 : JCL Visual Studio 2017 編
- メインフレーム COBOL 開発 : CICS Visual Studio 2017 編
- 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。