# 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 チュートリアルの準備

本チュートリアルを実行する前に「5. Enterprise Developer JCL チュートリアル」を終了している必要 があります。ここ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。

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 32bit) (ODBC 使用)

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

### 7.2 埋め込み SQL 付き COBOL ソース

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でのデバッグが 可能であり、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。

# 1) [プロジェクト] > [自動ビルド] をオフにしておきます。



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

▶ 🛃 MSSDEM	0				
		新規(N)	+		
	D	コピー	Ctrl+C		
	1	貼り付け	Ctrl+V		
	×	<b>削除(D)</b> 移動(V)	Delete		
		名前変更(M)	F2		
		指令の確定			
		インポート(I)	•	Ê.,	リモート COBOL プロジェクト
	4	エクスポート(0)		8	ローカル COBOL プロジェクトをリモート プロジェクトへ変換
	8	リフレッシュ(F)	F5	2	Net Express プロジェクトの変換
語アウ ※		プロジェクトを閉じる(S) 毎期後わプロミナクトを明じる(D)		è	インポート(I)

# 3) 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。

22+17		9 <del>8</del> - 26 - 26
<b>囲小</b> ル・ファイル・	システムから既存のプロジェクトヘリソースをインボートします。	2
インポート・ソースの道	鼎択(S):	
フィルター入力		
4 🗁 一般		*
👰 アーカイブ	・ファイル	
🛄 ファイル・	システム	
😂 既存プロジ	ェクトをワークスペースへ	=
💷 設定		
D 🗁 CVS		
D 📴 EJB		
👂 🗁 Java EE		
Micro Focus		
Micro Focus 1	<sup>、</sup> ンターフェイスマッパー	
Remote Syste	ms	-
K (m Wah		
?	< 戻る(B) 次へ(N) > 完了(F)	キャンセル
J		

# 4) 以下のダイアログの上の [参照] ボタンをクリックします。

77/11.8.7=1			
ソースは空にできません。			
欠のディレクトリーから(Y):		•	参照(R)
タイプのフィルター(T)) すべ	:て選択(5) 選択をすべて解除(D)		
洗フォルダー(L): MSSDEMO			参照(W)
電先フォルダー(L): MSSDEMO オプション			参照(W)
R先フォルダー(L): MSSDEMO オブション ■ 警告を出さずに既存リソースを上書き(O) ■ Create top-level folder(C) 拡張(A) >>			参照(W)

5) 以下のダイアログで C:¥DBturotial を選択し、[OK] をクリックします。



6) C:¥DBturotial 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。

「「」 インポート			
ファイル・システム ローカル・ファイル・システムからリソースをインボートします。			
次のディレクトリーから(Y): C:¥DBtutorial		ं 🗸	参照(R)
	V e dbdemol.jd V TBLCRTE.cbl V TBLDROP.cbl V TBLFETC.cbl V TBLISRT.cbl		
タインのノイルター(1) 9へて違択(5) 違 宛先フォルダー(L): MSSDEMO	Kを9へ(発烁(U)		参照(W)
オプション ■ 警告を出さずに既存リソースを上書き(O) ■ Create top-level folder(C) 拡張(A) >>			
?	< 戻る(B) 次へ(N) >	完了(F)	キャンセル

7) 今回追加インポートしたソースに対して個別の設定を行います。Shift キーまたは Ctrl キーを押しな がら対象ファイルを複数選択したら、右クリックしメニューから [プロパティー] を選択します。



8) 先の JCL チュートリアルでのプロパティー設定以外に、ここでは使用するデータベース製品に合わせ、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。 DIALECT(ENTCOBOL) P(COBSQL) ENDP

「」」プロパティー: TBLCRTE.cbl,	TBLDROR.cbl, TBLFETC.cbl			
フィルター入力	COBOL			$\phi \bullet \phi \bullet \bullet \bullet$
COBOL	マイルの固有な設定を可能にする(F)			
	文字集合:	ASCII	-	
	言語方言:	Enterprise COBOL for z/OS	•	
	ソース フォーマット:	固定	•	
	メインフレームのコピー処理:	СОРҮ	•	
	<ul> <li>□ 指令ファイルの生成</li> <li>□ リストファイルを生成</li> <li>☑ デバッグ用にコンパイル(D)</li> <li>□ 出力の表示</li> <li>☑ .GNT にコンパイル</li> </ul>			
	警告レベル: 最大エラー数: 	回復可能なエラーを含める(レベル E) 100	•	
	追加指令: DIALECT(ENTCOBOL) p(cobsql) endp			
			デフォルトの復元(D)	適用(A)
?			ОК	キャンセル

# 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。

DIALECT(ENTCOBOL) DB2(DB=SAMPLE, VALIDATE=RUN)

IN プロパティー: TBLCRTE.cbl,	, TBLDROP.cbl, TBLFETC.cbl			
フィルター入力	COBOL			↓ ↓ ↓ ↓
COBOL	アーイルの固有な設定を可能にする(F)			
	文字集合:	ASCII		
	言語方言:	Enterprise COBOL for z/OS	•	
	ソース フォーマット:	固定	•	
	メインフレームのコピー処理:	СОРУ		
	□ 指令ファイルの生成			
	□ リストファイルを生成			
	デバッグ用にコンパイル(D)			
	警告レベル:	回復可能なエラーを含める(レベル E)	•	
	最大エラー数:	100		
	追加指令:			100
	DIALECT(ENTCOBOL) db2(db=sample,va	lidate=run)		^
				<b>T</b>
			デフォルトの復元(D)	適用(A)
?			OK \$	ヤンセル

# 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。

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

プロパティー: TBLCRTE.cbl	, TBLDROP.cbl, TBLFETC.cbl	_ <b>_</b> ×
フィルター入力	COBOL	⇔•≎••
▷ COBOL	図ファイルの固有な設定を可能にする(F)	<u>^</u>
	文字集合: ASCII	
	amp/a・ ソースフォーマット: 国定 ・	
	×インフレームのコピー処理: □ 指令ファイルの生成 □ リストファイルを生成 ☑ デバッグ用にコンパイル(D) □ 出力の表示 ☑ .GNT にコンパイル	E
	警告レベル: 回復可能なエラーを含める(レベル E) ▼ 100	
	通知语令: DIALECT(ENTCOBOL) SQL(DBMAN=ODBC,BEHAVIOR=JCL,TARGETDB=MSSQLSERVER)	*
?	ОК	キャンセル

9) CICS プリプロセッサは不要ですので以下のようにチェックを外します。

フィルター入力	CICS ブリブロセッサ	⇔ ▼ ⇔ ▼ ▼
▲ COBOL CICS プリプロセッサ SQL プリプロセッサ 追加のプリプロセッサ	ファイルの固有な設定を可能にする(E)	^
	CICS プリプロセッサの使用	
	サポートされない機能の処理	
	○ 実行時にアベンド ○ コメント扱い ● コンパイルエラー	
	サポートされないオプションの処理 ○ 実行時にアペンド ○ コメント扱い ● コンパイルエラー ○ 実行時は無視	
		^
		~
	٢	>
?	ОК	キャンセル

10) 既に 8) 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、 ここでは[EXEC SQL] 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)

あり、そのため、ビルド構成またはプロジェクト プロパティでのみ設定で	きます。)
値	^
	•
	^
	$\sim$
L プリプロヤッサの使用" オブションを選択して有効にしてください。	
	<b>'</b>
	あり、そのため、ビルド構成またはプロシェクト プロパライでのみ設定で 値 し、プリプロセッサの使用"オプションを選択して有効にしてください。

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

# 7.3 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。



#### 7.4 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

🎉 « src 🕨 enterpriseserver 🕨	く C enterpriseserverの検討
<b>^</b> 名前 <sup>^</sup>	更新日時    種類
📔 exits	2014/02/19 15:01 ファイル フォルダー
👔 xa	2014/04/03 10:23 ファイル フォルダー
🊢 ト コンピューター ト ローカル ディスク (C:) ト	✓ C □-カル ディスク (C:)の検索
▲ 名前	更新日時 ▼ 種類
🚳 msdia80.dll	── 2006/12/01 23:37 アプリケーション拡張
🔐 work	2014/04/03 10:56 ファイル フォルダー
た場所 🔰 👔 😺	2014/04/03 10:47 ファイル フォルダー

 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(32-bit)] を管理者ユーザで起動します。C:¥xa ディレクトリに移動します。



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。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です。

#### Oracle の例:

build < ora10 または ora11 > を実行します。ここでは下記コマンドを実行します。

#### build ora11



#### 次に下記コマンドを実行します。

#### build ora1pc



#### 正常に作成されました。メッセージにあるように "ESORAOPC.DLL" ファイルを製品の bin フォルダへ

#### コピーします。

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

#### DB2 の例:

#### ここでは下記コマンドを実行します。

#### build db2



次に 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 ヘ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。

Micro Focus SQL Logon
Connect failed with SQL code -30082. SQL database SAMPLE
Enter user name and password:
User name: db2inst1
Password:
□ <u>S</u> ave
Save to INI file
OK Cancel

# 正常に作成されました。メッセージにあるように "ESDB2OPC.DLL" ファイルを製品の bin フォルダへ コピーします。

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



#### SQL Server の例:

### ここでは下記コマンドを実行します。

#### build mssql



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成してください。

例:C:¥Program Files (x86)¥Windows Kits¥8.0¥Lib¥win8¥um¥x86

퉬 « Windows Kits	8.0 → Lib → win8 → um → x86	~ ¢
^	名前 翻 Xaudio2.IID	更新日時 2012/0//25 16:0/
	🏙 xaswitch.lib	2012/07/25 16:12

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 の XA リソース定義で使用する 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。

ローカル DTCのプロパティ	×
トレース中 ログ セキュリティ	
セキュリティの設定	
▼ネットワーク DTC アクセス(D)	
- クライアントおよび管理 ■ 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)	
トランザクション マネージャー通信	_
✓ 受信を許可する(I) ✓ 送信を許可する(W)	
<ul> <li>相互認証を必要とする(L)</li> </ul>	
○ 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( <u>T</u> )	
<ul> <li>✓ XA トランザクションを有効にする(X)</li> <li>✓ SNA LU 6.2 トランザクションを有効にする</li> </ul>	( <u>E</u> )

# 7.5 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。ここでは CICS および JCL チュートリアルで使用した MSSDEMO リージョンを使用しますので、XA リソースを使用可能と するための準備を行います。

1) MSSDEMO が開始状態の場合にはいったん停止します。

2) Enterprise Server Admin にて MSSDEMO の [編集] ボタンをクリックします。



3) [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。

4) 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 DB2 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" 固定値 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(xxx1pc.dll は指定無し)
<<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:					
XAD	DB				
名前	:				
OR	ACLE_XA				
モジ	ב− <i>ו</i> ר:				
C:\x	a\ESORA11XA.DL	.L			
OPE	N文字列:				
OR	ACLE_XA+SesTm=	=100+SqlNet=te	ok-par+A	cc=P/scott/tig	jer
CLO	SE文字列:				
説明	:				
有効	:				
追	加 キャンセル				

### 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 XAUY-ス (1) MSS (✔) MQ スクリプト アク
ID:
XADB
名前:
DB2_XA
モジュール:
C:\xa\ESDB2XA.DLL
OPEN文字列:
DB=SAMPLE,uid=db2inst1,pwd=ibmdb2
CLOSE文字列:
有効: 🗹
追加キャンセル

# 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

─般 XAUV-ス (3) MSS (✔) MQ スクリプト アク・
ID:
XADB
名前:
SQLSVR_XA
モジュール:
C:\xa\ESMSSQL.DLL
OPEN文字列:
DSN=JCLTEST1
CLOSE文字列:
記約17: 
有効: 🗹
追加キャンセル

5) MSSDEMO を開始します。



 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。)

ナーバ・	<mark>ー</mark> ┃リスナー (3) ┃ サ <i>ー</i> ビス (4) ┃ ハンドラ (3) ┃ パッ	ケージ (0)
ロバテ	イ 構成 <b>診断</b> 過去の統計	
トレース	ເ ] ອັນສ໌ <mark>ES⊐ນນ−ル</mark> CS⊐ນນ−ル ]	
画面	● Show entries from 1 to 10 ● Show last 10 lines	of 45 total entries
Entry	Event	Show Entire Log
Entry 38	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS	Show Entire Log
Entry 38 37	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS 130927 18090538 1976 MSSDEMO CASXO0020I XADB XA interface loade	Show Entire Log 18:09:04 d. Name(Oracle_XA), Registration Mode(Static) 18:09:0
Entry 38 37 38	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS 130927 18090538 1976 MSSDEMC CASXO0020I XADB XA interface loade 130927 18090558 1976 MSSDEMC CASXO0021I XADB XA interface local	Show Entire Log 18:09:04 d. Name(Oracle_XA), Registration Mode(Static) 18:09:0 transaction support enabled 18:09:05
Entry 38 37 38 39	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS 130927 18090538 1976 MSSDEMC CASXO0020I XADB XA interface loade 130927 18090558 1976 MSSDEMC CASXO0021I XADB XA interface local 130927 18090601 1976 MSSDEMC CASXO0015I XADB XA interface initial	Show Entire Log 18:09:04 d. Name(Oracle_XA), Registration Mode(Static) 18:09:05 transaction support enabled 18:09:05 ized successfully 18:09:05
Entry 38 37 38 39 40	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS 130927 18090538 1976 MSSDEMC CASXO0020I XADB XA interface loade 130927 18090558 1976 MSSDEMC CASXO0021I XADB XA interface local 130927 18090601 1976 MSSDEMC CASXO0015I XADB XA interface initial 130927 18090621 1976 MSSDEMO CASSI50011 PLTPI - No PLT Specified	Show Entire Log 18:09:04 d. Name(Oracle_XA), Registration Mode(Static) 18:09:0 transaction support enabled 18:09:05 ized successfully 18:09:05 18:09:06
Entry 38 37 38 39 40 41	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS 130927 18090538 1976 MSSDEMC CASX00020I XADB XA interface loade 130927 18090558 1976 MSSDEMC CASX00021I XADB XA interface local 130927 18090601 1976 MSSDEMC CASX00015I XADB XA interface initial 130927 18090621 1976 MSSDEMO CASSI5001I PLTPI - No PLT Specified 130927 18090621 1976 MSSDEMO CASSI5040I Active SEP memory stra	Show Entire Log 18:09:04 d. Name(Oracle_XA), Registration Mode(Static) 18:09:0 transaction support enabled 18:09:05 ized successfully 18:09:05 18:09:06 tegy set to x'00000001', retain count 100 18:09:06
Entry 38 37 38 39 40 41 42	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS 130927 18090538 1976 MSSDEMC CASXO0020I XADB XA interface loade 130927 18090558 1976 MSSDEMC CASXO0021I XADB XA interface local 130927 18090601 1976 MSSDEMC CASXO0015I XADB XA interface initial 130927 18090621 1976 MSSDEMC CASSI50011 PLTPI - No PLT Specified 130927 18090621 1976 MSSDEMC CASSI5040I Active SEP memory stra 130927 18090621 1976 MSSDEMC CASSI5040I Active SEP memory stra 130927 18090621 1280 MSSDEMC CASSI1600I SEP initialization complete	Show Entire Log 18:09:04 d. Name(Oracle_XA), Registration Mode(Static) 18:09:0 transaction support enabled 18:09:05 ized successfully 18:09:05 18:09:06 tegy set to x'00000001', retain count 100 18:09:06 ed successfully 18:09:06
Entry 38 37 38 39 40 41 42 43	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS 130927 18090538 1976 MSSDEMO CASXO0020I XADB XA interface loade 130927 18090558 1976 MSSDEMC CASXO0021I XADB XA interface local 130927 18090601 1976 MSSDEMO CASXO0015I XADB XA interface initial 130927 18090621 1976 MSSDEMO CASSI5001I PLTPI - No PLT Specified 130927 18090621 1976 MSSDEMO CASSI5001I PLTPI - No PLT Specified 130927 18090621 1976 MSSDEMO CASSI5001I SEP initialization complete 130927 18090687 4404 MSSDEMO CASSI18001 SEP initialization complete 130927 18090687 4404 MSSDEMO CASBJ0005I Batch initiator started for	Show Entire Log 18:09:04 d. Name(Oracle_XA), Registration Mode(Static) 18:09:0 transaction support enabled 18:09:05 ized successfully 18:09:05 118:09:06 tegy set to x'00000001', retain count 100 18:09:08 de successfully 18:09:08 r job classes "ABC" 18:09:08
Entry 38 37 38 39 40 41 42 43 44	Event 130927 18090515 1976 MSSDEMO JES000042I SSTM not enabled: CICS 130927 18090538 1976 MSSDEMO CASXO0020I XADB XA interface loade 130927 18090558 1976 MSSDEMC CASXO0021I XADB XA interface local 130927 18090601 1976 MSSDEMO CASXO0015I XADB XA interface initial 130927 18090621 1976 MSSDEMO CASSI5001I PLTPI - No PLT Specified 130927 18090621 1976 MSSDEMO CASSI5001I PLTPI - No PLT Specified 130927 18090621 1976 MSSDEMO CASSI5001I SEP initialization complete 130927 18090621 1280 MSSDEMO CASSI50015 EP initialization complete 130927 18090687 4404 MSSDEMO CASSI50015 Batch initiator started for 130927 18090707 1280 MSSDEMO CASSI5021I PLTPI Phase 2 - No PLT	Show Entire Log 18:09:04 d. Name(Oracle_XA), Registration Mode(Static) 18:09:0 transaction support enabled 18:09:05 ized successfully 18:09:05 18:09:06 tegy set to x'00000001', retain count 100 18:09:08 tegy set to x'00000001', retain count 100 18:09:08 specified 18:09:08

# 💢 重要事項

「① 作成するオブジェクト;ビット数 = ② XA リソース;ビット数 = ③ Enterprise Server;ビット数 = ④ DB;許容するビット数 」 が正常に稼働するために必要な条件です。

### 7.6 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
- 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。



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

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 🛙 🔪 問題 🖉 タスク	
nterprise Server	
開始リージョン: MSSDEMO	
CASCD0167I ES Daemon successfully auto-started 18:08:56	
ASCD0050I ES "MSSDEMO" initiation is starting 18:08:56	
リージョン: MSSDEMO 正常に起動されました	
リージョン: M <u>SSDEMO 正常に起動されました</u> JCLCM0187I JOB01026 DBDEMO1 JOB SUBMITTED (JOBNAME=DBDEMO1,JOBNUM=01026)	18:22:22
リージョン: M <u>SSDEMO 正常に起動されました</u> JCLCM0187I JOB01026 DBDEMO1 JOB SUBMITTED (JOBNAME=DBDEMO1,JOBNUM=01026) JCLCM0180I JOB01026 DBDEMO1 Job ready for execution. 18:22:23	18:22:22

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したり、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、