

Visual COBOL チュートリアル

1

OpenESQL アシスタントを利用したデータベースアクセス

1 目的

レガシーな COBOL 言語で開発された企業システムからデータベースのアクセスを行うには繁雑な技術を要するというイメ ージをお持ちの方も多いかもしれませんが、Visual COBOL を使えば驚くほど簡単に開発が可能です。一般的にリレーショ ナル・データベースのデータ操作には SQL を利用します。SQL は、COBOL と異なるデータベース言語となるため、 COBOL コンパイラは SQL 文を解釈できません。

そこで、COBOL 上で SQL 文を利用する場合、EXEC SQL と END-EXEC で囲んだ部分のみに SQL 文を記述 し、COBOL と区別させます。この SQL 文が埋め込まれたソースを Oracle などデータベースベンダが提供するプリコンパ イラに渡すことで、SQL 文の部分をデータベースベンダが提供する API コールに展開した COBOL プログラムを生成しま す。プログラマーが当初作成するソースはこのソースではなくプリコンパイル展開する前のソースになるためデバッグ時のコードと 作成時のコードに相違が生まれてしまいます。

Visual COBOL はプログラマーが実際にメンテナンスする埋め込み SQL 文が入ったソースを IDE に認識させるべく、プ リコンパイルとコンパイルをシングルステップで処理できるようにしました。プリコンパイル、もしくはそれに相当する処理を Visual COBOL が内部的に処理します。

これにより、プリコンパイル後のソースは扱うことがなくなるため、このソースに関する考慮は不要となりました。

Visual COBOL はこのプリコンパイルを意識させない技術に関して技術や利用する DBMS に応じて柔軟に技術選択い ただけるよういくつかのオプションを用意しています。その中でも今回紹介する OpenESQL は、製品搭載のプリプロセッサで す。また、OpenESQL によるデータベースアクセスを行う場合、「OpenESQL アシスタント」という開発補助ユーティリティ を利用できます。

このチュートリアルでは、OpenESQL アシスタントを使用して OpenESQL によるデータベースアクセスの方法を学びます。



2 前提

本チュートリアルは、下記の環境を前提に作成されています。

OS	Windows 10
COBOL 製品	Visual COBOL 10.0 for Eclipse
DBMS 製品	SQL Server 2022

また、SQL 実行やレコード確認のため、SQL Management Studio を使用します。

下記のリンクから事前にチュートリアル用のサンプルデータベースの SQL ファイルをダウンロードして、任意のフォルダに解凍しておいてください。

サンプルデータベースの SQL スクリプトダウンロード

© Rocket Software, Inc. or its affiliates 1990–2024. All rights reserved. Rocket and the Rocket Software logos are registered trademarks of Rocket Software, Inc. Other product and service names might be trademarks of Rocket Software or its affiliates.



内容

- 1 目的
- 2 前提
- 3 チュートリアルについて
 - 3.1 データベースの準備
 - 3.2 Eclipse プロジェクトの作成と設定
 - 3.3 COBOL プログラムの作成とオプション設定
 - 3.4 ODBC データソースの設定
 - 3.5 クエリーの組み立てとテスト
 - 3.6 SQL プログラムの生成と埋め込み
 - 3.7 追加のコードを記述
 - 3.8 プログラムの実行

Rocket software

3 チュートリアルについて

3.1 データベースの準備

- 1) チュートリアル用データベースの作成
 - チュートリアルで使用するデータベースとサンプルテーブル、データのインポートを行う SQL を実行します。
 - ① SQL Server Management Studio の実行

Microsoft SQL Server Management Studio を起動し、管理者でログインを行います。※Microsoft SQL Server Management Studio は別途ダウンロードが必要です。

② SQL スクリプトの実行

ダウンロードした SQL スクリプトの内容をコピー/ペーストしてクエリーを実行します。

③ 実行結果の確認

初めて実行する場合はデータベースが存在しないので drop database 文だけが失敗します。[オブジェクトエクスプ ローラー] ツリーにてデータベースとして「OESQLDemo」が作成されており、EMP テーブルが存在することを確認し ます。

- 🖃 📕 Databases
 - 🗄 🔳 System Databases
 - 🗄 🛑 Database Snapshots
 - 🖃 🛢 OESQLDemo
 - 🗄 🛑 Database Diagrams
 - Tables
 System Tables
 - ∃ = System Tables
 - External Tables
 - 🗄 📕 Graph Tables
 - 🗄 🎞 dbo.ACT
 - 🗄 🎞 dbo.authors
 - 🗄 🎞 dbo.DEPT
 - 🗄 🏛 dbo.EACT
 - ⊞ dbo.EDEPT
 - ⊞ dbo.EEMP
 - 🕀 🎹 dbo.EEPA
 - 🗄 🎹 dbo.EMP
- ④ SQL Server Management Studio の終了

閉じるボタンにて「Microsoft SQL Server Management Studio」を終了します。

3.2 Eclipse プロジェクトの作成と設定

- 1) Visual COBOL for Eclipse の起動とプロジェクトの作成
 - ① スタートメニューより [Micro Focus Visual COBOL] > [Visual COBOL for Eclipse] を選択します。
 - ② ワークスペースの選択画面は任意のワークスペースを指定して、[起動(L)] ボタンをクリックします。
 - ③ Eclipse メニューより、[ファイル(F)] > [新規(N)] > [COBOL プロジェクト] を選択し、プロジェクト名に
 "OESQLAssistantTutorial"を指定して、[終了(F)] ボタンをクリックします。

4



COBOL プロジェクト

ワークスペースまたは外部の場所にCOBOL プロジェクトを作成します。

プロジェクト名(P) OESQLAssistantTutorial	
プロジェクト テンプレートを選択	
^ピ Micro Focus テンプレート [32 ビット] ピ Micro Focus テンプレート [64 ビット]	
	<u> ナソノレートの設定を構成</u>
□ テンフレートの参照	
場所:	参照
ファイルシステムを選択: default ~	
✓ デフォルト・ロケーションの使用(D)	
ロケーション(L): C:¥workspace_openesql_native¥OESQLAssistantTutorial	参照(R)
ファイル・システムを選択(Y): <mark>デフォルト</mark> 〜	
? 終7	(F) キャンセル

3.3 COBOL プログラムの作成とオプション設定

- 1) COBOL プログラムの作成
 - COBOL エクスプローラーにて作成した「OESQLAssistantTutorial」プロジェクトを右クリックし、コンテクストメニュ ーから [新規作成(N)] > [COBOL プログラム] を選択します。
 - ② [含まれるプロジェクト] に「OESQLAssistantTutorial」が指定されていること、新規ファイル名が 「Program1.cbl」になっていることを確認し、[終了(F)] ボタンをクリックします。
- 2) プロジェクトプロパティの設定
 - プロジェクトの構成を変更します。
 COBOL エクスプローラーにて作成した「OESQLAssistantTutorial」プロジェクトを右クリックし、コンテクストメニュ ーから「プロパティ(R)]を選択します。
 - ② プロパティ設定ダイアログが表示されます。
 [Micro Focus] > [プロジェクト設定] > [COBOL] > [SQL プリプロセッサ] をクリックし、以下の設定をお k なったうえで、[適用して閉じる] をクリックします。
 SQL プリプロセッサの使用: チェック プリプロセッサの種類: "OpenESQL"を選択
 BEHAVIOR: "ANSI" を選択



フィルタ入力	SQL プリプロセッサ			← → ⇒ ∛
 > リソース Coverage > Micro Focus ビルダー ビルダー ビルド構成 > ブロジェクト設定 > COBOL > SQL プリプロセッサ	 ✓ SQL ブリブロセッサの使用 ブリブロセッサの種類: OpenESQL 指令: フイルクテキストを入力 設定 ALLOWNULLCHAR ALLOWSERVERSELECT ANSI92ENTRY AUTOCCOMMIT AUTOFETCH BEHAVIOR BGP BEHAVIOR メインフレームマイグレーション指令を制 	値 ANSI 御します。注意: この指令は TARGET		~ ~
サーバー タスク・タグ > タスク・リボジトリー ビルダー プロジェクト・ネーチャー プロジェクト・ネーチャー プロジェクト・ファセット プロジェクト参照 > 検証 実行/デバッグ設定	SQL 指令: SQL(BEHAVIOR=ANSI)		デフォルトの復元(1)	》 《 》 適用(L)
?			適用して閉じる	キャンセル

- 3) OpenESQL アシスタントの設定変更
 - Eclipse メニューより [ウィンドウ(W)] > [設定(P)] を選択し、設定ダイアログを開き、[Micro Focus] > [データ ベース] > [OpenESQL アシスタント] をクリックします。

以下の設定を行ったのち、[適用して閉じる]をクリックします。

[DCLGEN] > [COBOL 変数の型] : "SQL TYPE"を選択

[一般] > [モード] : "ODBC"を選択

フィルタ入力	OpenESQL アシスタント		← → ⇒ §
> JS/TS (Wild Web Developer)	OpenESOL アシスタントの設定		
> Language Servers			
> Maven			
✓ Micro Focus			
> AWM			
> COBOL	設定	值	^
Enterprise Server	✓ DCLGEN		
コンテナー	ホスト変数の接頭語	表名	
> サービスインターフェイス	インジケータ変数	コピーブック末尾に生成する	
デバッグ	構造名	DCL + 表名	
テンプレート	COBOL 変数の型	SQL TYPE	
✓ データベース	レベル 49を使う	False	
DR2 バックグランド パース	▼ 一般		
OpenESQL アシスタント	自動的にカタログ	False	
OpenESQL バッククフノト パー	コードの生成	そのまま	
> ビルダー	項目を表名順に表示する	True	
ヘルプ	モード	ODBC	
リモート JRE	出力言語	COBOL	~
> 一般	x _ u [*]		
検索	していた。 したした「した」の1-3を指定します。	(必須)	
> 統合化トレース機能		(30.74)	
> Mylyn	デフォルト値: ODBC		\sim
> Oomph			
> TextMate ~		デコナルトの復元(五)	(室田(1)
< >		ノフィルトの復几(1)	旭用(L)
? 🕍 🖆 💿 😨		適用して閉じる	キャンセル

© Rocket Software, Inc. or its affiliates 1990–2024. All rights reserved. Rocket and the Rocket Software logos are registered trademarks of Rocket Software, Inc. Other product and service names might be trademarks of Rocket Software or its affiliates.

3.4 ODBC データソースの設定

- 1) ODBC データソースアドミニストレーター (32 ビット)の起動
 - スタートメニュー > 検索にて "ODBC" とタイプし、検索候補から「ODBC Data Sources (32 ビット)」を起動します。
- 2) データソースの設定と接続確認
 - ① ODBC データソースアドミニストレーター (32 ビット) 画面より [システム DSN] タブをクリックし、[追加(D)…] ボタ ンをクリックします。
 - ② [データソースの新規作成] ダイアログが表示されるので [ODBC Driver 17 for SQL Server] を選択し、[完了] ボタンをクリックします。

SOUDBC データ ソース アドミニストレーター (32 ビット)				
ユーザー DSN システム DSN	ファイル DSN ドライバー トレース 接続ブール バージョン情報			
システム データ ソース(S):				
名前 プラットフォーム ドライ	-71/-	追加(D)		
		削除(R)		
データントフのギ担作成		×		
) -)) - 入切利1兆1日以		~		
	セットアップするデータ ソースのドライバーを選択してください(S)			
	名前	^		
TTT B	Microsoft Excel-Treiber (*.xls)			
	Microsoft ODBC for Oracle			
	Microsoft Paradox-Treiber (*.db)			
	Microsoft Text Driver (*.txt; *.csv)			
	Microsoft Text-Treiber (*.txt; *.csv)			
	ODBC Driver 17 for SQL Server			
	< <	>		
		N		
	~ 床 ン (ロ) 元 」 ギヤ,			

 ③ [Name] に "OESQLDemo" を入力し、[Server] に接続する SQL Server アドレスを指定し、[次へ(N)]を クリックします。もし、不明な場合は、自社の DBMS 管理者に確認してください。



Create a New Data So	purce to SQL Server	×
Monet	This wizard will help you create an ODBC data source that you can use to connect SQL Server. What name do you want to use to refer to the data source?	to
SQLServer	Name: OESQLDemo	
	How do you want to describe the data source?	
	Description:	
81	Which SQL Server do you want to connect to?	
	Server: SQLServerAddr	\sim
	完了 次へ(N) > キャンセル ヘルプ	

 ④ SQL Server 接続のための認証方式を選択し [次へ(N)] ボタンをクリックします。不明な場合は、自社の DBMS 管理者に確認してください。(以下は SQL Server 認証の例です)

5	How should SQL Server verify the authenticity of the login ID?
Mogast	○ With Integrated Windows authentication .
SQL Server	SPN (Optional):
	\bigcirc With Azure Active Directory Integrated authentication .
	${\scriptstyle \sc black \below \sc black \s$
	\bigcirc With Azure Active Directory Password authentication using a login ID and password entered by the user.
	O With Azure Active Directory Interactive authentication using a login ID entered by the user.
	With Azure Managed Service Identity authentication.
	○ With Azure Service Principal authentication .
	Login ID: Sa
	Password:
	< 戻る(B) 次へ(N) > キャンセル ヘルプ

⑤ [Change the default database to] にチェックを入れて、作成したデータベース「OESQLDemo」を指定し、
 [次へ(N)] ボタンをクリックします。



	Change the default database to:
	OESQLDemo 🗸
Microsoft	Nurror server.
SQL Server	SPN for mirror server (Optional).
	Attach database filename:
	Use ANSI quoted identifiers. Use ANSI nulls, paddings and warnings. Application intent
	READWRITE
	│ Multi-subnet failover. ✓ Transparent Network IP Resolution. │ Column Encryption.
	Enclave Attestation Info:
	Keystore Configuration
	Use FMTONLYmetadata discovery.
	< 戻る(B) 次へ(N) > キャンセル ヘルプ

⑥ [完了] ボタンをクリックして設定を保存します。

8	Change the language of SQL Server system messages t	o:
Sto	(Default)	\sim
SQL Server	Use strong encryption for data.	
	Trust server certificate.	
	Perform translation for character data.	
14	Use regional settings when outputting currency, number	rs, dates and times.
	Save long running queries to the log file:	
	C:¥Users¥tarot¥AppData¥Local¥Temp¥QUERYLOG	Browse
	Long query time (milliseconds):	30000
	Log ODBC driver statistics to the log file:	
	C:¥Users¥tarot¥AppData¥Local¥Temp¥STATSLOG	Browse
	Connect retry count:	1
	Connect retry interval (seconds):	10
	< 戻る(B) 完了 キャン	セル ヘルプ

[Test Data Source] をクリックして接続確認を行ってください。

ODBC Microsoft SQL Server Setup		×
A new ODBC data source will be created wi	th the followin	g configuration:
Microsoft ODBC Driver for SQL Server Version Data Source Name: OESQLDemo Data Source Description: Server: rocky8-w-na.mshome.net,31433 Use Integrated Security: No DataBase: OESQLDemo Language: (Default) Data Encryption: No Trust Server Certificate: No Multiple Active Result Sets(MARS): No Mirror Server: Translate Character Data: Yes Log Long Running Gueries: No Log Driver Statistics: No Use Regional Settings: No Use ANSI Quoted Identifiers: Yes Use ANSI Null, Paddings and Warnings: Yes	on 17.10.0006	^
		\sim
Test Data Source	ок	Cancel



SQL Server ODBC Data Source Test				
Test Results				
Microsoft ODBC Driver for SQL Server Version 17.10.0006	^			
Running connectivity tests				
Attempting connection Connection established				
Verifying option settings Disconnecting from server				
TESTS COMPLETED SUCCESSFULLY				
ОК				

⑦ [OK] をクリックしたのち、ODBC データソースアドミニストレーター (32 ビット) 画面をクローズします。

3.5 クエリーの組み立てとテスト

1) OpenESQL アシスタントの起動

Eclipse メニューより、[実行(R)] > [ツール] > [データツール] > [OpenESQL アシスタント] をクリックします。

プロジェク	ト(P)	実行	<mark>テ(R)</mark> ウィンドウ(W) ヘルプ(H)				
4	r i	R	実行点をリセット		1 🖬		
🗆 🖻 (Progra	Q	実行(R)	Ctrl+F11			
00	PI	165	デバック(D)	F11			
	• • •	~	実行履歴(T)	>	···•····5····•6·····7··	I	8
	9	0	実行(S)	>			
			美仃悌成(N)				
			デバッグ履歴(H)	>			
		夺	デバッグ(G)	>			
	Θ		デバッグの構成(B)				
		0,	項目を検査				
		0	プログラム ブレークポイントを追加				
			ツール	>	構成	>	1
		Q	外部ツール(E)	>	データ ツール	>	OpenESQL アシスタント
		_			WSDL からクライアントを生成		クランツク ナータファイル ツール
			end program Program1.		JSON/YAML からクライアントを生成		データファイル ツール

- 2) テーブルの選択
 - ① OpenESQL アシスタントの画面より、表示されている [OESQLDemo] にチェックします。

File			
DESQLDemo	Query	Results	Details

② データソースの選択画面が表示されるので 3.4 で設定した ODBC データソースを選択し、[OK] をクリックします。



データ ソースの選択			×
ファイル データ ソース コンピューター	データソース		
データソース名 OESQLDemo	種類 説明 システム		
		新	規作成(N)
コンピューター データ ソースはこ ソースはこのコンピューターの一 ーターのユーザーとネットワークT	のコンピューターにのみ適序 人のユーザーが使用できま 『接続されているユーザー!	用され、共有できません。ニ す。システム データ ソース が使用できます。	1ーザー データ はこのコンピュ
	OK	キャンセル	ヘルプ

下記ダイアログが表示された場合は、データベースのログイン情報を入力した上で、[OK] をクリックします。

Password:	•••••	
Login ID:	58	Options >>
Server SPN:		Help
Authentication Mode	SQL Server 🗸 🗸	Cancel
Data Source:	OESQLDemo	ОК
SQL Server Login		×

③ テーブルの一覧が表示されるので「EMP」にチェックします。



④ [生成するクエリーのタイプを選択] ダイアログでは、[SELECT (cursor)] を選択し、[OK] をクリックします。



Select Type of Query to Create			>		
SELECT (cursor) SELECT DISTINCT (cursor) SELECT (Singleton) SELECT DISTINCT (Singleton)	Creates the code multiple rows of d	to retrieve ata.			
INSERT DELETE	Generate query as TSQL Stored Procedure				
	SP Name:	SP_CSR2			
	🗌 Generate query	with FOR UPDATE			
	Cursor Name:	CSR2			
OK Cancel					

[Query] タブにサンプルとなるコードが生成されます。

- 3) カラムの選択
 - EMP テーブルツリーから [A.FIRSTNME]、[A.LASTNAME]、[A.SALARY] にチェックします。[Query] タブの サンプルコードに選択したフィールドが付加されます。



カラム名に "A." が付与されていますが、これは上記画像の6行目にて EMP テーブルのエイリアスを "A" としてい るためです。 複数テーブルを JOIN する場合、エイリアスが "B", "C", … と設定することができ、異なるテーブルに同 名の項目が存在した場合でも、エイリアスを利用することで正しく取得先を識別することができます。

- 4) クエリーのテスト
 - ① [クエリーの実行] アイコン をクリックすると、[Results] タブに実行結果が表示されます。





Q	å L C E IC							
	EACT(dbo)	^	Query	Results	Details	Search Criteria	Sort Au	xiliary Code
	EDEPT(dbo)							
	EEMP(dbo)			FIRS	TNME1	LASTNAME2	SALA	ARY3
	EMP(dbo)		+	CHRIS	STINE	HAAS	5275	50.00
				MICH	AEL	THOMPSON	4125	50.00
	A.FIRSTNME			SALLY	(KWAN	3825	50.00
				JOHN		GEYER	4017	75.00
	A.WORKDEPT			IRVIN	G	STERN	3225	50.00
				EVA		PULASKI	3617	70.00
				EILEE	N	HENDERSON	2975	50.00
	A.EDLEVEL			THEO	DORE	SPENSER	2615	50.00
				VINCE	ENZO	LUCCHESI	4650	00.00
				SEAN		O'CONNELL	2925	50.00
	A.BONUS			DOLO	RES	QUINTANA	2380	00.00
	A.COMM	~		HEAT	HER	NICHOLLS	2842	20.00
	<	>						

3.6 SQL プログラムの生成と埋め込み

- 1) SQL プログラムの生成
 - ① OpenESQL アシスタントにて [Auxiliary Code] タブをクリックします。

Qu	Query Results De		Details	Se	earch Cri	iteria	Sort		Auxiliary (Code			
			FIRSTNME1			LASTN	IAME2	!	SALARY3				
	•		CHRISTINE			HAAS				52750.00			
			MICHAEL			THOMPSON			41250.00				
	SALLY				KWAN			8	38250.00				

 [Statement Type] にて「Generic SQL Program」を選択します。生成された補助コード欄に生成されたソー スコードが表示されます。

Query Results Details Search C	eria Sort Auxiliary Code	
Generate EXEC SQL Statement Type: Generic SQL Program	Generated Auxiliary Code: \$SET SQL(DBMAN=ODBC) WORKING-STORAGE SEC EXEC SQL INCLUDE S *> after an sql error this h 01 MFSQLMESSAGETEXT 01 IDX PIC X(04 EXEC SQL BEGIN DEC *> Put your host variables *> to other COBOL compile EXEC SQL END DECLA Host Variable Array Size:) sourceformat(variable) TION. QLCA END-EXEC. as the full message text PIC X(250). I) COMP-5. CLARE SECTION END-EXEC. here if you need to port ers ARE SECTION END-EXEC.

- 2) OpenESQL アシスタントが生成するプログラムコードを利用したデータベース接続プログラムの作成
 - ① [Program1.cbl] をエディターで開き、現在のコードをすべて削除したうえで保存します。
 - ② OpenESQL アシスタントにて [Auxiliary Code] タブをクリック選択し、[Add query to temporary file] アイ
 コン をクリックし、クリップボードにソースコードをコピーします。



⊈ & C C C C C		
EACT(dbo)	Query Results Details Search Criteria Sort Au	uxiliary Code
EDEPT(dbo)	Generate	
EEPA(dbo)	EXEC SQL SET	ed Auxiliary Code: SOL(DBMAN=ODBC) sourceformat(variable)
	WORKI	ING-STORAGE SECTION.
A.FIRSTNME	Statement Type: EX	EC SQL INCLUDE SQLCA END-EXEC.
	Generic SQL Program *> after	ar an sql error this has the full message text
	01 IDX	(PIC X(04) COMP-5.
	EX *> Put	(EC SQL BEGIN DECLARE SECTION END-EXEC.
	*> to o	other COBOL compilers
	FY	
ABIRTHDATE	Host Va	ariable Array Size: 1

③ 「Program1.cbl」をエディター上で開き、Ctrl+V で生成された補助コードを貼りつけます。

```
*Program1.cbl ×
   Program1.cbl >
   •••••*A•1•B••••••2•••••3•••••4•••••5•••••6
         $SET SQL(DBMAN=ODBC) sourceformat(variable)
          WORKING-STORAGE SECTION.
               EXEC SQL INCLUDE SQLCA END-EXEC.
          *> after an sql error this has the full message text
          01 MFSQLMESSAGETEXT PIC X(250).
          01 <del>IDX</del>
                               PIC X(04) COMP-5.
               EXEC SQL BEGIN DECLARE SECTION END-EXEC.
         *> Put your host variables here if you need to port
         *> to other COBOL compilers
               EXEC SQL END DECLARE SECTION END-EXEC.
          PROCEDURE DIVISION.
               EXEC SQL
                   WHENEVER SQLERROR perform OpenESQL-Error
               END-EXEC
```

④ "CONNECT TO 'OESQLDemo' の末尾に "USER 'データベースユーザーID.パスワード' "句を記述します。以下の例では、データベースユーザーID に "sa", パスワードに "HogeHoge" を指定しています。

EXEC SQL CONNECT TO 'OESQLDemo' USER 'sa.HogeHoge' END-EXEC

補足)

上記例は、SQL Server 認証が有効な場合の記述例です。環境に合わせ、データベースユーザーID にはデータベース にアクセス可能なログインユーザー、パスワードにはデータベースユーザーID のパスワードを設定してください。なお、認証方 式や情報については、自社のデータベース管理者にお問い合わせください。

⑤ OpenESQL アシスタントにて [Query] タブをクリックします。[Add query to temporary file] アイコンをクリック し、クリップボードにソースコードをコピーします。





- ⑥ Eclipseの COBOL エディターにて下記のコメント部分の下部に CTRL+V でコードを張り付けます。
 - "*> Put your program logic/SQL statements here"



エラーが報告されますが、ここでは無視してください。

⑦ ファイルの変更を保存します。



3.7 追加のコードを記述

- 1) コピーブックの生成
 - OpenESQL アシスタントの「EMP」テーブルツリー内で選択した上で右クリックし、コンテクストメニューから [Generate DCLGEN] を選択します。



 [Generate Host Variable Declarations] ダイアログが表示されるので OESQLAssistantTutorial プロジェ クトフォルダ配下まで移動した上で、"EMP.cpy" という名称でコピーブックを保存します。

🚯 Generate Host Variable D	Declarations				×
← → × ↑ 📙 « wa	orkspa > OESQLAssistantTutorial >	× ت	OESQLAssistan	tTutorialの検索	Q
整理 ▼ 新しいフォルダー					?
🏪 ローカル ディスク (C ^	名前 ^	更	新日時	種類	
PC	.settings	20	24/08/08 13:47	ファイル フォルダー	-
🧊 3D オブジェクト	New_Configuration.bin	20	24/08/08 16:24	ファイル フォルダー	-
🕹 ダウンロード					
🔜 デスクトップ					
🔮 ドキュメント					
📰 ピクチャ					
📑 ビデオ					
🎝 ミュージック					
🟪 ローカル ディスク (C	4				>
	•				-
ファイル名(N): EMRc	ру				~
ファイルの種類(T): Copyt	book files(*.cpy)				\sim
▲ フォルダーの非表示			保存(S)	キャンセル	

- 2) INCLUDE 命令の追加
 - OpenESQL アシスタントにて [Auxiliary Code] タブをクリックします。[EXEC SQLINCLUDE] 文が生成されて いるので [Add file to temporary file] アイコン をクリックし、クリップボードにソースコードをコピーします。



Fi	le								
0	Č C C C C C C C C C C C C C C C C C C C	^	Query	Results	Details	Search Criteria	Sort	Auxiliary Code	
			Generate				Generated Auxiliary Code: EXEC SQL INCLUDE EMP END-EXEC.		
ELLO L(dbo) 		Sta	State Gen	ement Typ eric SQL I	e: Program	~			

- ② Eclipse の COBOL エディターにて EXEC SQL INCLUDE SQLCA に下に CTRL+V で貼り付けます。
- ③ Eclipse メニューより [ファイル(F)] > [保存(S)] を選択し、プログラムを保存します。



- 3) 表示フォーマットの追加
 - Eclipse の COBOL エディターにて「EXEC SQL INCLUDE EMP END-EXEC」の下に下記の命令を追加し

```
ます。
```

01 EDIT-PAY PIC Z,ZZ99.99.

② コードを下の方にスクロールして「DISPLAY 'ROW FOUND'」命令を以下の2行で置き換えます。

```
MOVE EMP-SALARY TO EDIT-PAY
             DISPLAY 'NAME: ' EMP-LASTNAME ', ' EMP-FIRSTNME ' SALARY: ' EDIT-PAY
UPON CONSOLE
  修正前)
  PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
     EXEC SQL
       FETCH CSR2 INTO
          : EMP-FIRSTNME
         ,:EMP-LASTNAME
          ; : EMP-SALARY : EMP-SALARY-NULL
     END-EXEC
     *> Process data from the Fetch
     IF SQLCODE = 0
         DISPLAY 'ROW FOUND'
     *> for array fetches, field sqlerrd(3) contains
     *> the number of rows returned
     *>
        PERFORM VARYING IDX FROM 1 BY 1
     *>
           UNTIL IDX > SQLERRD(3)
   修正後)
```



```
PERFORM UNTIL SQLCODE < 0 OR SQLCODE = +100
  EXEC SQL
    FETCH CSR2 INTO
       :EMP-FIRSTNME
       ,: EMP-LASTNAME
       ,: EMP-SALARY: EMP-SALARY-NULL
  END-EXEC
   *> Process data from the Fetch
  IF SQLCODE = 0
       MOVE EMP-SALARY TO EDIT-PAY
       DISPLAY 'NAME: ' EMP-LASTNAME ', ' EMP-FIRSTNME ' SALARY: ' EDIT-PAY UPON CONSOLE
   *> for array fetches, field sqlerrd(3) contains
   *> the number of rows returned
  *> PERFORM VARYING IDX FROM 1 BY 1
         UNTIL IDX > SQLERRD(3)
  *>
```

3.8 プログラムの実行

- 1) プログラムの実行
 - Eclipse メニューより [実行(R)] > [実行構成(W)] をクリックします。
 - [COBOL アプリケーション] を選択し、マウスの右クリックでコンテクストメニューを開き、 [新規構成(W)] を選択します。

構成の作成、管理、および実行

COBOL プログラムを実行します

	このダイアログから起動設
フィルタ入力	📑 - 選択した種類の構成
🗄 Apache Tomcat 🔨	- 選択した種類の起動構
AspectJ/Java Application	💫 - 選択した構成をエク
	📄 - 選択した構成をコピ
페 COBOL JVM アプリケーション	🛛 🗙 - 選択した構成を削
〒 COBOL JVM ユニット テスト 同 COBOL JVM リモート アプリケーショ	🍸 - フィルタ・オプションを
■ COBOL アプリケーション	ゴロレクノゴナリンクナス
G COBOL ユニット テスト C 新規構	\$成(W)
🐵 Debug Adapter Launch 🖉 新規ノ	ロトダイノ(P)
● Eclipse アプリケーション 🍀 エクスオ	ĩ−ト(X)

③ [名前] に "OESQLDemo" を入力し、[実行(R)] をクリックします。

名前(N) OESQLDemo						
🗟 一般 🦻 ソース 写 環境 🔲 共通 🔎 実行時 💱 🕫	デバッグシンボル	3 動的分析	所 @î CTF	🥒 コンテナー		
▼ COBOL プロジェクト(P)						^
OESQLAssistantTutorial	参照					
▶ 接続プロパティ						
▼ 主ブログラム						
✓ プログラムはプロジェクトビルド構成の一部: New C	Configuration	~				
New_Configuration.bin/OESQLAssistantTutorial	.e 参照					
▼ 開始オプション						
コマンド行引数:						
		$\hat{}$				
作業ディレクトリ:						~
						_
		前回	保管した状	態に戻す(V)	適用(Y)	
				実行(R)	閉じる	



④ プログラムが実行され EMP テーブルの内容が出力されることを確認します。

C:¥Windows¥SYSTEM32¥cmd.exe

NAME:	PIANKA ,	ELIZABETH	SALARY:	2,2250.00
NAME:	YOSHIMURA ,	MASATOSHI	SALARY:	2,4680.00
NAME:	SCOUTTEN ,	MARILYN	SALARY:	2,1340.00
NAME:	WALKER ,	JAMES	SALARY:	2,0450.00
NAME:	BROWN ,	DAVID	SALARY:	2,7740.00
NAME:	JONES ,	WILLIAM	SALARY:	1,8270.00
NAME:	LUTZ .	JENNIFER	SALARY:	2,9840.00
NAME:	JEFFERSON .	JAMES	SALARY:	2,2180.00
NAME:	MARINO .	SALVATORE	SALARY:	2.8760.00
NAME:	SMITH .	DANIEL	SALARY:	1,9180.00
NAME:	JOHNSON .	SYBIL	SALARY:	1,7250.00
NAME:	PEREZ .	MARIA	SALARY:	2,7380.00
NAME:	SCHNEIDER .	ETHEL	SALARY:	2,6250.00
NAME:	PARKER .	JOHN	SALARY:	1,5340.00
NAME:	SMITH .	PHILIP	SALARY:	1,7750.00
NAME:	SETRIGHT .	MAUDE	SALARY:	1,5900.00
NAME:	MEHTA .	RAMLAL	SALARY:	1,9950.00
NAME:	LEE .	WING	SALARY:	2.5370.00
NAME:	GOUNOT .	JASON	SALARY:	2.3840.00
NAME:	HEMMINGER .	DIAN	SALARY:	4.6500.00
NAME:	ORLANDO ,	GREG	SALARY:	2,9250.00
NAME:	NATZ .	KIM	SALARY:	2,8420.00
NAME:	YAMAMOTO .	KIYOSHI	SALARY:	2,4680.00
NAME:	JOHN .	REBA	SALARY:	2,9840.00
NAME:	MONTEVERDE .	ROBERT	SALARY:	2,8760.00
NAME:	SCHWARTZ .	EILEEN	SALARY:	2,6250.00
NAME:	SPRINGER .	MICHELLE	SALARY:	1,5900.00
NAME:	WONG .	HELENA	SALARY:	2,5370.00
NAME:	ALONZO .	ROY	SALARY:	2,3840.00
続行す	「るには何かキーを	押してください		

何かキーをタイプすると、実行画面がクローズします。

免責事項

ここで紹介したソースコードは、機能説明のためのサンプルであり、製品の一部ではございません。ソースコードが実際に動作するか、御社業務に適合するかなどに関しまして、一切の保証はございません。 ソースコード、説明、その他すべてについて、無謬性は保障されません。 ここで紹介するソースコードの一部、もしくは全部について、弊社に断りなく、御社の内部に組み込み、そのままご利用頂いても構いません。 本ソースコードの一部もしくは全部を二次的著作物に対して引用する場合、著作権法の精神に基づき、適切な扱いを行ってください。