

1.1.サンプル 11

JDBC-BC を使用してデータベースの参照および更新を行う手順を、サンプルアプリケーションを例にして説明します。

このサンプルは、WebOTX ESB をマルチプロセスモードで動作させることを前提としています。

この項で説明する手順は次のとおりです。

- サンプルのインストール
- ドメインの起動
- アプリケーショングループとプロセスグループの作成と起動
- JDBC データソースの登録
- サービスアセンブリの配備と開始
- クライアントアプリケーションのコンパイル実行

1.1.1.サンプルのインストール

JDBC-BC のサンプルアプリケーションは、WebOTX マニュアルのインストールに伴って、次の場所に配置されますので、zip ファイルを展開してください。

<WebOTX マニュアルインストールディレクトリ>/samples/esb/esb.zip

ファイル構成

- 11/config/jdbcdbc_sample.properties dsadm コマンドで JDBC データソースを登録するためのプロパティファイルです。
- 11/artifacts/jdbcdbc_sample_SA.zip JDBC-BC のサービスアセンブリです。
- 11/clients/jdbc_client.zip データメンテナンス用のアプリケーションです。

1.1.2.ドメインの起動

ドメインの起動

ドメインが起動されていなければ、起動してください。

```
otxadmin> start-domain domain1
```

1.1.3.アプリケーショングループとプロセスグループの作成と起動

WebOTX AS の Standard Edition または Enterprise Edition で、マルチプロセスモードにて ESB

をご利用になる場合、アプリケーショングループとプロセスグループを作成します。

アプリケーショングループの作成

アプリケーショングループを作成します。

```
otxadmin> login --user admin --password adminadmin
otxadmin> create-apg apg1
Command create-apg executed successfully.
```

プロセスグループの作成

プロセスグループ (JBI コンテナ用) を作成します。

```
otxadmin> create-pg --version 7 --kind j2ee --apgroup apg1 pg1
Command create-pg executed successfully.
```

JBI コンテナの有効化

プロセスグループ起動時に JBI コンテナを起動するように設定します。

```
otxadmin> set tpsystem.applicationGroups.apg1.processGroups.pg1.enabledJBIContainer=true
```

(※) 改行していますが実行する際は、1行で記述して実行してください。

アプリケーショングループ プロセスグループの起動

作成したアプリケーショングループを起動します。

```
otxadmin> start-apg apg1
Command start-apg executed successfully.
```

1.1.4.JDBC データソースの登録

JDBC データソースの定義変更

実際の環境に合わせて、`jdbcbc_sample.properties` の次の内容を修正してください。

```
dataSourceName=
userName=
password=
```

JDBC データソースの登録

JDBC データソースを登録します。

```
> dsadm bind jdbcbc_sample.properties
```

(*) ポート番号 6212 の domain1 以外に接続する場合は、dsadm コマンドの`-url`、`-user`、`-password` オプションで適切な値を指定してください。

1.1.5.サービスアセンブリの配備と開始

サービスアセンブリの配備

`jdbcbc_sample_SA.zip` があるディレクトリで、以下のコマンドを実行します。

```
>jbiadmin.bat --user admin --password adminadmin --port 6212 deploy-service-assembly
jdbcbc_sample_SA.zip apg1 pg1
Deployed Service Assembly jdbcbc_sample_SA
```

(*) `--port` には該当ドメインのポート番号を指定してください。アプリケーショングループ名とプロセスグループ名は、WebOTX AS Standard Edition または Enterprise Edition で、マルチプロセスマードにて ESB をご利用になる場合にだけ指定してください。

サービスアセンブリの開始

配備したサービスアセンブリを開始します。

```
> jbiadmin.bat --user admin --password adminadmin --port 6212 start-service-assembly
jdbcbc_sample_SA
Started Service Assembly jdbcbc_sample_SA
```

(*) `--port` には該当ドメインのポート番号を指定してください。

1.1.6. クライアントアプリケーションのコンパイルと実行

クライアントアプリケーション

clients\jdbc_client.zip

を展開してください。

ファイル構成

- compile.bat コンパイル実行バッチです。
- run.bat クライアント実行バッチです。
- src\SampleClient.java クライアントのソースです。
- class\SampleClient.class クライアントのクラスです。
- SampleTable.sql データベースのテーブル作成用ストアドプロシージャ(参考)です。

コンパイル

コンパイル、および、実行時に使用する JDK のコマンドが環境変数パスに追加されていない場合は、追加してください。

compile.bat を実行し、コンパイルを行ってください。

```
>compile.bat
```

JDBC ドライバへのクラスパス修正

実際の環境に合わせて、run.bat の次の内容を修正してください。

```
@set WEBOTX_HOME=
@set JDBC_DRIVER=
```

実行

クラスパスに class ディレクトリを追加した後、run.bat を実行し、プログラムを実行します。

```
>run.bat
```

起動後、次のメニューが表示されます。

```
-----
Menu) 1. Create tables. (In/Outbound tables)
      2. Insert a record. (Inbound table)
      3. Select records. (Inbound table)
      4. Select records. (Outbound table)
      5. Delete records. (Inbound table)
      6. Delete records. (Outbound table)
      7. Drop tables. (In/Outbound tables)
      9. End.

Please select number?
```

1. Inbound/OutBound 用のテーブルを作成します。

```
Please select number? 1
```

```
-----
Created a table for Inboud successfully
Created a table for Outboud successfully
```

2. Inbound で読み込まれるレコードを登録します。

```
Please select number? 2
```

```
-----
Inserted into a sample record successfully
```

3. Inbound 用のテーブルの内容を確認します。処理が完了したレコードは削除されます。

Please select number? 3

12345 , Tokyo , suzuki taro , suzuki_taro@sample.co.jp , 11223344556

Please select number? 3

Not Found Records

4. Outbound 用のテーブルの内容を確認します。処理が完了したレコードが追加されます。

Please select number? 4

Not Found Records

Please select number? 4

12345 , Tokyo , suzuki taro , suzuki_taro@sample.co.jp , 11223344556