

JNDI データソースの WebSphere Liberty 設定

WebSphere Liberty で EAR Java プログラムが `server.xml` ファイルで定義されたデータソースを使用しているかどうかを確認するには、アプリケーションが JNDI (Java Naming and Directory Interface) 名を介してデータソースにアクセスしているかどうかを確認する必要があります。以下はその手順です。

ステップ 1: アプリケーションが使用する JNDI 名を特定する

まず、EAR アプリケーションがデータソースを検索するために使用する JNDI 名を確認します。Java EE アプリケーションが EAR ファイルとしてデプロイされる場合、この設定は以下の場所に指定されることがあります。

- **デプロイメント記述子:** アプリケーションのデプロイメント記述子 (例: `web.xml` (Web モジュール) または `ejb-jar.xml` (EJB モジュール)) を確認し、`<resource-ref>` 要素を探します。この要素は、アプリケーションがデータソースにアクセスするために使用するリソース参照を宣言します。例えば:

```
<resource-ref>
  <res-ref-name>jdbc/myDataSource</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

この場合、アプリケーションは JNDI 名 `java:comp/env/jdbc/myDataSource` を使用してデータソースを検索します。

- **バインドファイル:** WebSphere Liberty では、デプロイメント記述子からのリソース参照がサーバーの JNDI 名にバインドされることがあります。例えば、`ibm-web-bnd.xml` (Web モジュール) または `ibm-ejb-jar-bnd.xml` (EJB) のようなバインドファイルを確認します。以下のように `<resource-ref>` バインドを探します:

```
<resource-ref name="jdbc/myDataSource" binding-name="jdbc/actualDataSource"/>
```

この場合、アプリケーションの参照 `jdbc/myDataSource` はサーバーの JNDI 名 `jdbc/actualDataSource` にマッピングされます。

- **アプリケーションコード:** ソースコードにアクセスできる場合は、JNDI 検索やアノテーションを検索します。

- **JNDI 検索:** 以下のようなコードを探します:

```
Context ctx = new InitialContext();
DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/myDataSource");
```

これは JNDI 名 `java:comp/env/jdbc/myDataSource` を示しています。

- **アノテーション:** 現代の Java EE アプリケーションでは、`@Resource` アノテーションが使用されることがあります。例えば:

```
@Resource(name = "jdbc/myDataSource")
private DataSource ds;
```

これは `java:comp/env/jdbc/myDataSource` を指しています。

バインドファイルが存在しない場合、コードまたはデプロイメント記述子（例：`jdbc/myDataSource`）の JNDI 名がサーバー設定で期待される名前と直接対応することがあります。

ステップ 2: `server.xml` 設定を確認する

アプリケーションが使用する JNDI 名（直接またはバインドを介して）を特定したら、`server.xml` ファイル（および `<include>` 要素を介して含まれる設定ファイル）で一致するデータソース定義を確認します。`server.xml` のデータソースは通常 `<dataSource>` 要素で定義されます。例えば：

```
<dataSource id="myDataSource" jndiName="jdbc/myDataSource">
  <jdbcDriver libraryRef="myDBLib"/>
  <properties url="jdbc:mysql://localhost:3306/mydb" user="user" password="pass"/>
</dataSource>
```

- `jndiName` 属性（例：`jdbc/myDataSource`）を確認します。
- アプリケーションが使用する JNDI 名（例：`jdbc/myDataSource` またはバインド名 `jdbc/actualDataSource`）と比較します。

JNDI 名が一致する場合、アプリケーションは `server.xml` で定義されたデータソースを使用しています。

ステップ 3: 結果を解釈する

- **一致が見つかった場合:** アプリケーションが使用する JNDI 名が `server.xml`（または含まれるファイル）の `<dataSource>` 要素と一致する場合、EAR Java プログラムは `server.xml` データソースを使用しています。
- **一致が見つからない場合:** `server.xml` に一致する JNDI 名が見つからない場合、アプリケーションはサーバー定義のデータソースを使用していない可能性があります。プログラマ的にデータソースを作成しているかもしれません（例：JDBC ドライバーを直接 URL、ユーザー名、パスワードで使用する）。ただし、これは管理環境のような WebSphere Liberty では一般的ではなく、推奨されません。

追加の考慮事項

- **含まれる設定ファイル:** `server.xml` が `<include>` 要素を使用している場合（例：`<include location="datasources.xml"/>`）、そのファイル内のデータソース定義を確認します。
- **デフォルト動作:** バインドファイルが JNDI 名をオーバーライドしない場合、アプリケーションはリソース参照名（例：`jdbc/myDataSource`）が `server.xml` のデータソースと一致することを前提とすることがあります。

- ・ **サーバーログやツール:** 設定の確認には直接関係ありませんが、サーバーログを確認するか、WebSphere Liberty Admin Center を使用してアプリケーションに関連付けられているアクティブなデータソースを確認することもできます。これは実行時の検証です。

結論

WebSphere Liberty で EAR Java プログラムが `server.xml` データソースを使用しているかどうかを確認するには：1. アプリケーションが使用する JNDI 名を特定します（デプロイメント記述子、バインドファイル、コードを介して）。2. その JNDI 名が `server.xml` またはその含まれるファイルで定義されたデータソースと一致するかどうかを確認します。

一致がある場合、アプリケーションは `server.xml` データソースを使用しています。そうでない場合、別の方法を使用している可能性があります。