ORACLE

OIDおよびOUDのディレクトリ・ネーミングのためのOracle Databaseクライアントの構成

Oracle Internet Directory（OID）およびOracle Unified Directory（OUD）におけるネットワークの名前およびアドレスの一元化

2023年6月

目次

1. はじめに

アプリケーションをOracle Databaseに接続する際は、データベースのホスト名やサービス名などの情報を含む接続記述子を使用することが必要です。接続記述子は、アプリケーションによってさまざまな方法で指定できます。たとえば、アプリケーション接続リクエストでハードコードしたり、tnsnames.ora構成ファイルに保存された接続記述子にマッピングされる識別子をアプリケーションが渡したりすることができます。tnsnames.oraファイルを使用する代わりに、外部のマッピング・サービスを使用して接続記述子を検索することもできます。提供されるサービスの1つがディレクトリ・ネーミングです。

ディレクトリ・ネーミングによって、ネットワークの名前とアドレスが1つの場所で一元化され、名前の変更と更新が簡単に管理できるようになります。これにより、tnsnames.oraファイルに保存された接続記述子を管理者が変更する必要がなくなります。大規模な組織には、何百、何千ものデータベース・アプリケーションやtnsnames.oraファイルが存在する可能性があります。

オラクルには、Oracle Internet Directory（OID）とOracle Unified Directory（OUD）という2つのディレクトリ・サーバー製品があります。この技術概要では、これらのインストール後に名前解決を構成する方法について説明します。これらの製品は、Identity & Access Managementのダウンロードからダウンロードできます。OID 12.2.1.4をインストールするには、このインストール・ガイドに従ってください。OUD 12.2.1.4.0をインストールするには、このインストール・ガイドに従ってください。

2. OIDおよびOUDのOracleディレクトリ・ネーミング用クライアントのセットアップ

インストールが完了すると、OIDとOUDにはディレクトリ・ネーミング・オブジェクトで使用されるスキーマができます。ディレクトリ・ネーミングを使用するためのクライアント構成ファイルを作成するには、Oracle Net Configuration Assistantツール（"Oracle NetCA"）を使用します。このツールを使用すると、OracleContextオブジェクトを含むネーミング・コンテキストを選択できます。すべてのディレクトリ・ネーミング・オブジェクトは、OracleContextの下に作成されます。

構成を開始するには以下を実行します。

1. $ORACLE_HOME/bin/netcaを実行
2. 「Directory Usage Configuration」オプションを選択
3. Directory Typeドロップダウンで、OIDディレクトリ・サーバーとOUDディレクトリ・サーバーのどちらの場合も「Oracle Internet Directory」オプションを選択

4. OIDディレクトリ・サーバーまたはOUDディレクトリ・サーバーの詳細を入力
5. ネーミング・コンテキストを選択

完了すると、Oracle NetCAによってファイルldap.oraが作成されます。環境変数$TNS_ADMINが設定されている場合、ファイルは$TNS_ADMINディレクトリに存在します。それ以外の場合、ファイルは$ORACLE_HOME/network/adminまたは$ORACLE_BASE_HOME/network/admin（インストールが読取り専用ORACLE_HOMEの場合）に存在します。

以下はldap.oraファイルの例です。

```
DIRECTORY_SERVERS = (oidserver.example.com:389:636)
DEFAULT_ADMIN_CONTEXT = "DC=example,DC=com"
DIRECTORY_SERVER_TYPE = OID
```

ディレクトリ・ネーミングを使用するクライアント・マシンすべてにldap.oraファイルをコピーするか、またはそれらすべてのマシンでOracle NetCAを再度実行します。

ご参考までに、Oracle Net Configuration Assistantドキュメントはこちらにあります。

3. ディレクトリ・サーバーでのエントリの作成

ディレクトリ・サーバーでネーミング・エントリを作成して管理するには、最初に1つのディレクトリ・ユーザーを作成する必要があります。このユーザーが、上記でセットアップしたOracleContextに保存されるネーミング・エントリを管理します。ユーザーを作成するには、ディレクトリ・サーバー・ドキュメントに従ってください。たとえば、OID 12.2.1.4の場合はこちらに従ってください。OUDの場合はこちらに従ってください。以下の例では、名前を"mynaminguser"と仮定します。

そして、Oracle Net Managerツール（“Oracle NetManager”）、コマンドライン・ツール、またはREST APIを使用することにより、以降のセクションに示すネーミング・エントリを作成して管理できます。REST APIの使用については、ドキュメントREST API for Oracle Unified Directory Data Managementで説明しています。

3.1 Oracle Net Managerによるネーミング・エントリの作成

次の手順を実行してください。

1. $ORACLE_HOME/bin/netmgrを実行します。

   セクション2で作成されたldap.oraファイルが正しい場所にある場合は、Oracle NetManagerによってディレクトリ・アイコンがツリー・ナビゲータに表示されます。アイコンが表示されない場合は、ldap.oraが$ORACLE_HOME/network/adminまたは$ORACLE_BASE_HOME/network/adminにあることを確認してください（いずれかの場所が存在する場合）。または、TNS_ADMIN環境変数がldap.oraを含むディレクトリに設定されていることを確認してください。

   ディレクトリ・サブツリーがない場合は、セクション3.2のコマンドライン・ツールを使用してください。

2. ディレクトリ・サブツリーをクリックして展開します。

   ディレクトリ・サーバー・ユーザーの資格証明を入力するように求められます。セクション3の開始時に作成された資格証明を入力します。

3. サービス・ネーミング・サブツリーを選択します。
4. 左側のメニューにある緑色のプラス記号アイコンをクリックします。
5. ネット・サービス名ウィザード・プロンプトに従って、ネット・サービス名および記述子をディレクトリ・サーバーに作成します。

詳しくは、Oracle Net Managerドキュメントを参照してください。

3.2 コマンドライン・ツールを使用したネーミング・エントリの作成

Oracle Net Managerに代わる方法として、ldapaddやldapdeleteなどのLDAPコマンドライン・ツールを使用する方法があります。

たとえば、エイリアス"sales"を追加するには、以下に示すようにファイルsales.ldifを作成します。

```
dn: cn=sales,cn=oraclecontext,dc=example,dc=com
objectclass: top
objectclass: orclNetService
orclnetdescstring:
(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=databasehost)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=mydbservice.us.example.com)))
cn: sales
```

この例では、OracleContextが"dc=example,dc=com"の下にあることを前提としています。Oracle Databaseのホスト、ポート、サービス名を使用するようにorclnetdescstringを調整します。

そして、以下の構文を使用してldapaddを実行します。

```
ldapadd -h <ldap host> -p <ssl port> -D <Bind DN> -U 2 -q \
        -Q -W file:<wallet directory> -P <wallet password> \
        -f <ldif file>
```

たとえば、次のとおりです。

```
ldapadd -h mydirectoryserverhost -p 636 \
        -D "cn=mynaminguser,dc=example,dc=com" -U 2 -q \
        -Q -W file:/opt/oracle/wallets -P Mysecret -f sales.ldif
```

双方向のTLS 1.2認証を使用している場合は、代わりにオプション-U 3を使用する必要があります。

エントリを削除するには、ldapdeleteコマンドを使用します。構文はldapaddに似ていますが、削除するエイリアスを以下のように指定します。

```
ldapdelete -h <ldap host> -p <ssl port> -D <Bind DN> -U 2 -q \
           -Q -W file:<wallet directory> -P <wallet password> \
           <DN of entry to be deleted>
```

または、削除する1つまたは複数のDNを含むファイルを以下のように指定できます。

```
ldapdelete -h <ldap host> -p <ssl port> -D <Bind DN> -U 2 -q \
           -Q -W file:<wallet directory> -P <wallet password> \
           -f <ldif file with DNs to be deleted>
```

たとえば、次のとおりです。

```
ldapdelete -h mydirectoryserverhost -p 636 \
           -D "cn=mynaminguser,dc=example,dc=com" -U 2 -q \
           -Q -W file:/opt/oracle/wallets -P Mysecret \
           "cn=sales,cn=oraclecontext,dc=example,dc=com"
```

または

```
ldapdelete -h mydirectoryserverhost -p 636 \
           -D "cn=mynaminguser,dc=example,dc=com" -U 2 -q \
           -Q -W file:/opt/oracle/wallets -P MySecret \
        -f sales-delete.ldif
```

この例では、sales-delete.ldifに以下のような行が含まれます。
cn=sales,cn=oraclecontext,dc=example,dc=com

ldapaddのように、双方向のTLS 1.2認証にはオプション-U 3を使用できます。

4. ディレクトリ・ネーミングのためのクライアント側の構成

クライアント・マシンの構成を継続します。

4.1 ネーミング・メソッド参照の有効化

ディレクトリ・ネーミングを使用するすべてのマシン上で、sqlnet.ora構成ファイルを作成または編集します。このファイルは、セクション2のldap.oraファイルと同じディレクトリにある必要があります。

sqlnet.oraファイルで、LDAPをNAMES.DIRECTORY_PATHパラメータに追加します。アプリケーション接続記述子は、そのパラメータで指定したネーミング・メソッドの順に評価されます。たとえば、最初にLDAPを試行してからEasy Connect構文を使用してフォールバックし、最後にtnsnames.oraファイルで接続文字列を参照する場合の例は次のとおりです。

```
NAMES.DIRECTORY_PATH = (LDAP, EZCONNECT, TNSNAMES)
```

NAMES.DIRECTORY_PATHエントリが存在しない場合、LDAPは使用されますが最初には考慮されません。

4.2 ディレクトリ・サーバーの認証

デフォルトでは、ディレクトリ・ネーミングで匿名バインディングが実行されます。ただし、ディレクトリ・サーバーが匿名バインディングを無効化している場合は、以下に示す2つのうちいずれかの方法で、ディレクトリ・サーバーに対して認証を構成する必要があります。

4.2.2 OIDのクライアント証明書ベースの認証

OIDでは、ディレクトリ・サーバーのアクセスに証明書ベースの認証を使用できます。この方法はOUDでは使用できません。

おのおののsqlnet.oraファイルは、認証されたバインディングを有効化し、X.509クライアント証明書を含むウォレットを指している必要があります。たとえば、以下の行をsqlnet.oraに追加します。

```
NAMES.LDAP_AUTHENTICATE_BIND = TRUE
WALLET_LOCATION =
                (SOURCE = (METHOD = FILE)
                    (METHOD_DATA =
                    (DIRECTORY = <wallet directory>)
                )
```

OIDディレクトリ・サーバーは、証明書のDNをそのユーザー名として受け入れ、ディレクトリ・エントリにマッピングされるように構成する必要があります。

4.2.3 ユーザー名およびパスワードベースの認証

または、証明書認証を使用する代わりに、ユーザー名およびパスワードを使用してOIDまたはOUDディレクトリ・サーバーにアクセスできます。この認証方式は、Active Directoryを使用するMicrosoft Windows以外のアプリケーションでも使用できます。アプリケーションは、Oracle Database 21c（または

それ以降）のクライアント・ライブラリを使用する必要があります。

この方式では、ディレクトリ・サーバーのユーザー名およびパスワードはウォレットに保存されます。クライアントは、これらの資格証明を一方向のTLS接続経由でディレクトリ・サーバーへ渡すことによって認証を行います。

sqlnet.oraファイルを編集し、認証されたバインディングを有効化してバインド方式を設定します。たとえば、以下の行をファイルに追加します。

```
NAMES.LDAP_AUTHENTICATE_BIND = TRUE
NAMES.LDAP_AUTHENTICATE_BIND_METHOD = LDAPS_SIMPLE_AUTH
WALLET_LOCATION =
                (SOURCE = (METHOD = FILE)
                    (METHOD_DATA =
                    (DIRECTORY = <wallet directory>)
                )
```

ディレクトリ・サーバー証明書のルートCA証明書を取得し、以下のようにウォレットに保存します。
ユーザー名とパスワードも追加する必要があります。

1. ウォレットがまだ存在しない場合は作成します。

   ```
   orapki wallet create \
         -wallet <directory to create the wallet in>
   ```

2. 証明書をウォレットに追加します。

   ```
   orapki wallet add -wallet <wallet directory> \
         -trusted_cert -cert <root CA certificate>
   ```

3. ユーザー名を追加します。

   ```
   mkstore -wrl <wallet directory> -createEntry \
         oracle.ldap.client.dn <DN of the user>
   ```

   たとえば、次のとおりです。

   ```
   mkstore -wrl /opt/oracle/wallets -createEntry \
         oracle.ldap.client.dn "cn=user1,dc=acme,dc=com"
   ```

   Active Directoryでは、DNユーザー名は、User Principal NameまたはDown Level Logon Name（別名SAMAccountName）となることもあります。

   たとえば、fooがドメイン名でuser1がユーザー名の場合は次のとおりです。

   ```
   mkstore -wrl /opt/oracle/wallets -createEntry \
         oracle.ldap.client.dn "foo\user1"
   ```

   または

   ```
   mkstore -wrl /opt/oracle/wallets -createEntry \
         oracle.ldap.client.dn "user1@foo"
   ```

4. ユーザーのパスワードを追加します。

   ```
   mkstore -wrl <wallet directory> -createEntry \
         oracle.ldap.client.password <password>
   ```

5. ウォレットを自動ログオン・ウォレットにします。

   ```
   orapki wallet create -wallet <wallet directory> \
         -auto_login
   ```

5. 接続のテスト

ネーミング・サーバーの構成が完了しました。構成したクライアント・マシンのいずれかからSQL*Plusを実行することによって接続を検証できます。ネットワーク構成ファイルがデフォルトの場所にあること、または環境変数

TNS_ADMINをそれらのファイルが含まれるディレクトリに設定していることを確認してください。データベース・ユーザー“scott”が存在し、エイリアス“sales”がsales.ldifファイルにある場合は、次のようにして接続します。

```
sqlplus scott@sales
```

6. まとめ

この技術概要では、OIDまたはOUDを使用してディレクトリ・ネーミング用にOracle Databaseクライアントを構成する方法について説明しました。