

Oracle Databaseの Easy Connect Plus

Oracle Databaseを使用したアプリケーションの構成可能なデータベース接続構文

技術概要 / 2024年8月8日

免責事項

本文書には、ソフトウェアや印刷物など、いかなる形式のものも含め、オラクルの独占的な所有物である占有情報が含まれます。この機密文書へのアクセスと使用は、締結および遵守に同意したOracle Software License and Service Agreementの諸条件に従うものとします。本文書と本文書に含まれる情報は、オラクルの事前の書面による同意なしに、公開、複製、再作成、またはオラクルの外部に配布することはできません。本文書は、ライセンス契約の一部ではありません。また、オラクル、オラクルの子会社または関連会社との契約に組み込むことはできません。

本書は情報提供のみを目的としており、記載した製品機能の実装およびアップグレードの計画を支援することのみを意図しています。マテリアルやコード、機能の提供をコミットメント（確約）するものではなく、購買を決定する際の判断材料になさらないでください。本書に記載されている機能の開発、リリース、および時期については、弊社の裁量により決定されます。

製品アーキテクチャの性質上、コードが大幅に不安定化するリスクなしに、本書に記載されているすべての機能を安全に含めることができない場合があります。

目次

Easy Connect Plusの概要.....	4
Easy Connectの背景	4
Easy Connect Plusの構文.....	4
プロトコル	5
ホスト.....	5
ポート.....	5
サービス名.....	6
サーバー・タイプ	6
インスタンス名	6
パラメータ.....	6
パラメータ.....	6
セキュリティ・パラメータ.....	6
プロキシ・パラメータ	7
説明パラメータ.....	7
DRCPパラメータ.....	9
暗黙的接続プーリング・パラメータ.....	10
例.....	10
まとめ.....	15

Easy Connect Plusの概要

Oracle Database 19cでは、Easy Connect Plusと呼ばれる大幅に強化された接続文字列構文が導入されました。Oracle Database 21cでは、Easy Connect Plus構文にDRCP接続パラメータを追加できます。

Easy Connect Plusでは、TLS接続、ウォレット、ロードバランシング、接続のタイムアウトなどの機能の使用やネットワーク・バッファ・サイズの調整を容易に行うことができるようにしています。これにより、*sqlnet.ora*や*tnsnames.ora*などの外部構成ファイルの必要性が減少しています。たとえば、セキュアな通信のために相互TLS (mTLS) 経由でOracle Cloud Databaseサービスにアクセスするには、ウォレット・ロケーションをEasy Connect Plus文字列に含めて、これらの構成ファイルを排除できます。

Easy Connect Plusは、[JDBC](#)¹、CベースのOracle Call Interface (OCI)、Oracle C++ Call Interface (OCCI)、および.NETアプリケーションで使用できます。Python、Node.js、PHP、Ruby、Go向けのOracle Databaseドライバなど、OCIに組み込まれたアプリケーションやドライバでも使用できます。

Easy Connectの背景

Easy Connectは、現在OracleプログラムやOracleアプリケーションがOracle Databaseサービスへの接続時に使用している構文です。もっともシンプルなEasy ConnectのURL文字列には、次のようにOracle Databaseが実行されているホスト名、オプション・ポート、データベース・サービス名が含まれます。

```
host[:port][/[service_name]]
```

たとえば、Oracle Databaseがデフォルトのプラグブル・データベース・サービス“orclpdb1”を使用する *mydbhost.example.com* という名前のホストで実行されている場合、SQL*Plusでの一般的な接続は以下のとおりです。

```
SQL> connect scott/pw@mydbhost.example.com/orclpdb1
```

Pythonでは以下のとおりです。

```
connection = oracledb.connect(user='scott',  
password='password', dsn='mydbhost.example.com/orclpdb1')
```

Easy Connect Plusの構文

Easy Connect Plus構文は、より多くの接続オプションをサポートする、Easy Connectのスーパーセットです。この構文は、Oracle Client 19c (およびそれ以降のバージョン) を使用し、Oracle Database 11.2以降に接続するすべてのOracle Databaseドライバ (*JDBC*、*ODP.NET*²、*OCI*、*C++ベースのOCCI*、*python-oracledb*、*node-oracledb* など) で使用できます。

前のセクションで挙げた通常のEasy Connect機能に加えて、Easy Connect Plusは以下の機能を提供します。

- **TCPS接続**とURLのセキュリティ関連のプロパティを使用可能にします。たとえば、次のとおりです。
`tcps://salesserver1:1521/sales.us.example.com?ssl_server_cert_dn="cn=sales,cn=OracleContext,dc=us,dc=example,dc=com"`

¹ Java Database Connectivity (JDBC) は、オラクルが開発したJavaの標準データベース・アクセスAPIです。

² Oracle Data Provider for .NET – ODP.NET

- URLの**接続プロパティ**が使用可能になり、これによってアプリケーションの高可用性（HA）機能が有効になります。たとえば、次のとおりです。
`tcps://salesserver1:1521/sales.us.example.com?connect_timeout=60&transport_connect_timeout=30&retry_count=3`
- URLで複数のホスト名と複数のポート番号が使用可能になり、Oracle RAC³に対応します。たとえば、次のとおりです。
`tcps://salesserver1:1521,salesserver2,salesserver3:1522/sales.us.example.com`

Oracle DatabaseのEasy Connect Plus構文は次のとおりです。

```
[[protocol://]host1{,host12}[:port1]{,host2:port2}{;host3{,host32}[:port3}]/[service_name][:server_type]
[/instance_name]][?parameter_name1=value1{&parameter_name2= value2}]
```

旧バージョンのクライアント（19c、21cなど）では、Easy Connect Plusでのアドレス・リスト（セミicolon区切り）の使用はサポートされていません。そのため、Oracle 19cおよび21cクライアントでのOracle DatabaseのEasy Connect Plus構文は次のとおりです。

```
[[protocol://]host1{,host2}[:port1]{,host2:port2}]/[service_name][:server_type]
[/instance_name]][?parameter_name1=value1{&parameter_name2= value2}]
```

プロトコル

プロトコルでは、データベース・ホストへの接続中に使用される転送プロトコルを指定します。19c以降では、サポートされる値はTCPおよびTCPSです。デフォルトはTCPです。

ホスト

ホスト・リストは、データベースに接続するために使用されるホスト名またはIPアドレスをカンマで区切ったリストです。

複数のホストが使用される場合、ロードバランシングが自動的に有効化されます。

Easy Connect文字列内に複数のアドレス・リストがある場合、ホスト・リストをセミicolonで区切ることもできます。

ポート

ポート番号は、データベース・サービスがリッスンするポートを示します。デフォルトは1521です。

複数のホストを使用しており、各ホストで同じポートが使用される場合は、以下を使用します。

`host1,host2:port`

または、使用する別々のポートを次のように明示できます。

`host1:port1,host2:port2`

³ Oracle Real Application Clusters

サービス名

サービス名は、データベースが識別される[サービス](#)です。データベース名とドメイン名で構成され、インストールやデータベース作成時に入力される名前です。

サービス名は、データベース・ホストで"`lsnrctl services`"を実行することで表示できます。

データベースの"システム識別子" (SID) は、Easy ConnectおよびEasy Connect Plusの構文では使用できないことに注意してください。

サーバー・タイプ

サーバー・タイプでは、接続を処理するためにデータベース・ホストで使われるサーバーの種類を指定します。

専用、**共有**、または**プール**のいずれかを指定できます。デフォルトでは、指定せずにリスナーがサーバー・タイプを決定します。

一般的な従来型のサーバー・タイプは[専用](#)です。

[共有サーバー](#)を使用すると、サーバー・プロセスは共有されますが、セッション状態は共有されません。セッション状態はSGAに格納されます。

[プール・サーバー](#)は、[データベース常駐接続プーリング](#) (DRCP) が使用するサーバーです。サーバー・プロセスとセッション状態はどちらも共有されます。セッション状態はPGAに格納されます。

共有サーバーとDRCPは、データベース・ホストに多数の専用接続を処理するための十分なメモリがない場合に使用されます。

インスタンス名

サービスに相当する特定のインスタンスに接続する必要がある場合、インスタンス名を使用します。

パラメータ

パラメータは、接続の動作を制御する名前/値ペアです。構文では、パラメータのはじまりを表すために'?'を、各パラメータ間のデリミタとして'&'を使用します。パラメータ値の先頭と末尾の空白は無視されます。値の一部として空白が必要な場合は、二重引用符で囲む必要があります。パラメータについては次のセクションで説明します。

パラメータ

Easy Connect Plusでは、セキュリティ、プロキシ、説明、およびDRCPの各パラメータを指定できます。

JDBC Thinを除くあらゆるドライバでは、ここに記載していないパラメータはすべて説明パラメータとして渡されるため、接続時にさらなる（将来的な）オプションの使用が可能です。JDBC Thinには、許可された説明パラメータの[ホワイトリスト](#)が含まれます。

セキュリティ・パラメータ

セキュリティ・パラメータは、データベースへのTLS接続を構成するために使用されます。このパラメータを使用する場合は、プロトコルがTCPSに設定されていなければなりません。

[SSL_SERVER_CERT_DN](#)

データベース・サーバーの識別名 (DN) を指定します。

SSL_SERVER_DN_MATCHが**ON**の場合、このDNはサーバー証明書のDNと一致します。

SSL_SERVER_DN_MATCH

識別名（DN）の一致によってサーバー証明書をクライアント側で検証するかどうかを指定します。この値は**ON**または**OFF**に設定できます。デフォルト値は**ON**です（Oracle Client 19.2以降）。

このパラメータは、**SSL_SERVER_CERT_DN**とともに使用されます。

SSL_SERVER_CERT_DNパラメータが指定されていない場合は、デフォルトではデータベース・サーバーのホスト名（元の接続文字列の**ホスト**パラメータによって指定）が、サーバー証明書のコモンネーム（CN）およびサーバー別名（SAN）フィールドに対して検証されます。検証が失敗に終わると、接続は成功しません。

WALLET_LOCATION

Oracleウォレットが格納されるディレクトリを設定します。ウォレットとは、SSLによって処理される証明書、鍵、トラストポイント⁴です。

ファイルシステム・ウォレットのみがこの方法でサポートされます。

プロキシ・パラメータ

プロキシ・パラメータは、プロキシ・サーバー経由でルーティングされるトラフィックを構成します。

接続が成功するかどうかは、特定のプロキシ構成に依存します。データ送信のパフォーマンスは、プロキシの能力に依存します。オラクルでは、パフォーマンスが極めて重要な本番環境でこの機能を使用することを推奨していません。

HTTPS_PROXY

TLSクライアント接続をトンネリングするためのHTTPプロキシ・ホスト名またはIPアドレスを設定します。非TLS接続はトンネリングできません。

HTTPS_PROXY_PORT

TLSクライアント接続をトンネリングするためのHTTPプロキシ・ホスト・ポートを設定します。

説明パラメータ

その他のパラメータはすべて、説明パラメータです。これには以下が含まれます。

CONNECT_TIMEOUT

アプリケーションがOracle Net接続を確立するときのタイムアウト期間（秒）を設定します。デフォルトではタイムアウトはありません。

⁴ バージョン19.14および21.5のOracle Clientのライブラリを使用しているドライバの場合は、組込みトラストポイント・サポートを使用できます。この機能を使用すると、（もっとも一般的に使用されるRoot CAが存在する）システム・トラストストアに存在するRoot CAによってサーバー証明書が署名される限り、SSOウォレットは必要ありません。この機能は、ManagedおよびCore ODP.NETドライバ（バージョン19.13および21.4以降）でもサポートされています。すべてのバージョンのJDBC Thinドライバでは、JDKの"トラストストア"メカニズムがデフォルトで使用されています。

EXPIRE_TIME

プローブを送信して接続がアクティブであることを確認する時間隔（分）を設定します。0より大きい値を設定すると、データベース・サーバーの異常終了が原因で接続が永久に開いたままにならないようにすることができます。ファイアウォールによってアイドル接続が終了されないようにするために**EXPIRE_TIME**を使用する場合、この値は、ファイアウォールのタイムアウト期間の半分未満にする必要があります。

FAILOVER

複数ホストにおける接続時フェイルオーバーを有効または無効にします。この値は**ON**または**OFF**に設定できます。デフォルトは**ON**です。

LOAD_BALANCE

複数ホストにおけるOracleクライアントのロードバランシングを有効または無効にします。この値は**ON**または**OFF**に設定できます。デフォルトは**ON**です。

RECV_BUF_SIZE

TCP/TCP socketsの受信バッファ・サイズ（バイト）を設定します。接続の帯域遅延積（BDP）に対応するようにこのパラメータを調整する必要があります。データベース・サーバー構成ファイルの *sqlnet.ora*でも、このパラメータを設定する必要があります。

BDPとは、任意の時点における"有線"（データ・ネットワーク）上のデータ量を意味します。デフォルトのオペレーティング・システム・バッファは、有線を満たすための十分なデータを保持しません。最初は、受信バッファ・サイズをBDPに対応するように設定することが推奨されます。たとえば、エンドポイント間のラウンドトリップ時間が25ミリ秒の40 Mbpsネットワークでは、BDPは、 $(40 * 1000 / 8)$ KB/秒 * (0.025) 秒、すなわち125 KBです。よって、**RECV_BUF_SIZE**パラメータは125000に設定できます。

バッファ・サイズを大きくすると、アプリケーションはより多くのデータをオペレーティング・システムのキューに挿入して、"有線"上で利用可能な帯域幅を効率的に活用できます。

SEND_BUF_SIZE

TCP/TCP socketsの送信バッファ・サイズ（バイト）を設定します。接続の帯域遅延積（BDP）に対応するようにこのパラメータを調整する必要があります。データベース・サーバー構成ファイルの *sqlnet.ora*でも、このパラメータを設定する必要があります。

最初は、送信バッファ・サイズを、先に**RECV_BUF_SIZE**で述べたBDPに対応するように設定することが推奨されます。事例は**RECV_BUF_SIZE**を参照してください。

SDU

Oracle Netセッション・データ・ユニット（SDU）の packetsize（バイト）を設定します。**SDU**の値を大きくすると、メモリを犠牲にしてシステム・コール数とCPU使用率が低減されるため、ネットワークのスループットが向上します。データベース・サーバー構成でも、このパラメータを設定する必要があります。

データの一括送信では、このパラメータを64 Kに設定します。

SOURCE_ROUTE

複数ホスト経由のネットワーク・ルーティングを有効にします。この値は**ON**または**OFF**に設定できます。デフォルトは**OFF**です。

RETRY_COUNT

Oracle Databaseへの接続を試行する際にホスト・リストを反復する回数を設定します。

接続が成功するまで、またはホスト・リスト全体が指定された回数反復されるまで、ホスト・リストの各ホストに対して接続が順に試行されます。1つのホストのみが使用される場合、このホストへの接続が、指定された回数試行されます。

RETRY_DELAY

ホスト・リストの反復が再試行される間隔（秒）を設定します。デフォルト値は0です。このパラメータは、**RETRY_COUNT**パラメータと併せて使用することで機能します。

TRANSPORT_CONNECT_TIMEOUT

クライアントがOracleデータベースへのOracle Net接続を確立するときの転送接続タイムアウト期間（秒）を設定します。このパラメータは**CONNECT_TIMEOUT**のサブセットであり、TCPおよびTCPS接続の確立にのみ適用されます。

その他のパラメータも、JDBC Thinドライバ以外のすべてのドライバに使用できます。その他のパラメータは、*tnsnames.ora*内の同等の接続記述子エントリの**DESCRIPTION**セクションでパラメータとして処理され、接続中に検証されます。

DRCPパラメータ

Oracle Database 21cに接続するバージョン19c以降のOracle Clientを実行するアプリケーションでは、以下のパラメータを通じてアプリケーションのEasy Connect Plus構文で[データベース常駐接続プーリング](#)（DRCP）設定を構成できます。

POOL_CONNECTION_CLASS

Oracle DatabaseへのDRCP接続の接続クラスに明示的に名前を付けます。接続クラスは、アプリケーションによって必要とされる接続のタイプの論理名です。接続クラスを設定すると、（同一のデータベース・ユーザーとしてOracle Databaseに接続する）さまざまなアプリケーションが、DRCPにプールされたセッションを、アプリケーションに対応する論理名を使用して識別できます。そのため、特定の接続クラスに属するセッションは、接続クラス外では共有されません。

POOL_CONNECTION_CLASS値で可能な最大長は1,024文字（NULL終端文字を含む）で、アスタリスク（*）を含めることはできません。

POOL_PURITY

アプリケーションが以前のセッション状態では解決されない新しいセッションを必要としているか、またはOracle DatabaseへのDRCP接続から以前のセッションを再利用する必要があるかを指定します。値は、**NEW**または**SELF**になります。**POOL_PURITY=SELF**を使用することで、最適な再利用とパフォーマンスが実現します。

Easy Connect文字列で**POOL_CONNECTION_CLASS**を設定し、接続を効率的にパーティション化してセッション情報の漏えいを防ぐことを推奨します。**POOL_PURITY=SELF**を設定することで、特にアプリケーションの複数のインスタンスがDRCP経由でOracle Databaseにアクセスしてセッション情報を共有できる場合に、最適なパフォーマンスと最良のリソース（プロセスおよびセッション）利用が実現します。

アプリケーションがEasy Connect文字列で**POOL_PURITY=SELF**を使用する場合、アプリケーションがDRCP接続のドロップ・セッション・コールに対応する保証はありません。

DRCPパラメータは、Oracle Databaseのコンテナ・データベース (CDB) とプラガブル・データベース (PDB) の両方で構成できます。これらのパラメータは、19c以降のバージョンのOracle ClientおよびManaged ODP.NET Thinドライバを使用するあらゆるクライアント・ドライバで使用できます。JDBC Thinドライバではサポートされていません。

暗黙的接続プーリング・パラメータ

Oracle Database 23ai以降、Oracle DatabaseのDRCP機能と、Traffic DirectorモードのOracle Connection Manager (CMAN-TDM) のプロキシ常駐接続プーリング (PRCP) 機能の両方で、暗黙的接続プール (Implicit Connection Pooling) 機能が導入されました。暗黙的接続プーリングを有効にするには、接続文字列でPOOL_BOUNDARYパラメータを設定する必要があります。

POOL_BOUNDARY

DRCPまたはPRCPで暗黙的接続プーリング (Oracle Database 23aiの機能) を有効にします。値は、**STATEMENT**または**TRANSACTION**になります。この設定では、アプリケーション・セッションを解放してDRCPまたはPRCPプールに戻すための境界条件を指定します。

例

このセクションでは、さまざまなEasy Connect Plus構文の例を紹介します。ここでは、各Easy Connect接続文字列が、*tnsnames.ora*や他の構成ファイルで使用される可能性のある接続記述子をどのように置き換えることができるかを示します。

例1 - プロトコル

`tcp://salesserver1:1521/sales.us.example.com`

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

例2 - 複数のホスト

`salesserver1,salesserver2:1782/sales.us.example.com`

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (LOAD_BALANCE=ON)
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1782))
  (ADDRESS=(PROTOCOL=tcp) (HOST=salesserver2) (PORT=1782))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

例3 - 複数のホストとポート

`salesserver1:1521,salesserver2,salesserver3:1522/sales.us.example.com`

以下の記述子を置き換えます。

```
(DESCRIPTION=
```

```
(LOAD_BALANCE=ON)
(ADDRESS=(PROTOCOL=tcp)(HOST=salesserver1)(PORT=1521))
(ADDRESS=(PROTOCOL=tcp)(HOST=salesserver2)(PORT=1522))
(ADDRESS=(PROTOCOL=tcp)(HOST=salesserver3)(PORT=1522))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

例4 – 複数のアドレス・リスト

salesserver1:1521;salesserver2,salesserver3:1522/sales.us.example.com

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp)(HOST=salesserver1)(PORT=1521))
  )
  (ADDRESS_LIST=
    (LOAD_BALANCE=ON)
    (ADDRESS=(PROTOCOL=tcp)(HOST=salesserver2)(PORT=1522))
    (ADDRESS=(PROTOCOL=tcp)(HOST=salesserver3)(PORT=1522))
  )
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
)
```

この例は、旧バージョンのOracleクライアント（21cまたはそれ以前）では動作しません。

例5 – ホストとしてのIPv6アドレス

[2606:b400:41c:8a19:f816:3eff:fe8f:e3ef]:1521/sales.us.example.com

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcp)(HOST=2606:b400:41c:8a19:f816:3eff:fe8f:e3ef)
    (PORT=1521))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
)
```

数値のIPv6アドレスは、角括弧で囲む必要があります。

例6 – TLS接続

tcps://salesserver1:1521/sales.us.example.com

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (ADDRESS=(PROTOCOL=tcps)(HOST=salesserver1)(PORT=1521))
  (SECURITY=(SSL_SERVER_DN_MATCH=ON))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
)
```

SSL_SERVER_DN_MATCH=ONが完全な接続文字列に自動的に追加され、サーバーDNを一致させることに注意してください。接続に使用されるホスト名（この例ではsalesserver1）は、サーバーのSSL証明書のCN⁵およびSAN⁶フィールドと照合されます。クライアント・アプリケーションは、一致が見つかった場合に限りデータベースに接続します。

⁵ CN - Common Name

⁶ SAN - Subject Alternative Name

例7 – TLS接続 (サーバーDN一致なし)

`tcps://salesserver1:1521/sales.us.example.com?ssl_server_dn_match=off`

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521))
  (SECURITY= (SSL_SERVER_DN_MATCH=OFF))
  (CONNECT_DATA= (SERVICE_NAME=sales.us.example.com)) )
```

この場合、サーバーDN一致は実行されません。

例8 – ウォレット・ロケーション

`tcps://salesserver1:1521/sales.us.example.com?wallet_location="/tmp/oracle"`

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521))
  (CONNECT_DATA= (SERVICE_NAME=sales.us.example.com))
  (SECURITY= (WALLET_LOCATION=/tmp/oracle) (SSL_SERVER_DN_MATCH=ON)) )
```

例9 – 識別名の一致

`tcps://salesserver1:1521/sales.us.example.com?ssl_server_cert_dn="cn=sales,cn=OracleContext,dc=us,dc=example,dc=com"`

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521))
  (SECURITY= (SSL_SERVER_DN_MATCH=ON)
  (SSL_SERVER_CERT_DN=cn=sales,cn=OracleContext,dc=us,dc=example,dc=com))
  (CONNECT_DATA= (SERVICE_NAME=sales.us.example.com)) )
```

例10 – HTTPSプロキシ

`tcps://salesserver1:1521/sales.us.example.com?https_proxy=www-proxy.mycompany.com&https_proxy_port=80`

以下の記述子を置き換えます。

```
(DESCRIPTION=
  (ADDRESS= (PROTOCOL=tcps) (HOST=salesserver1) (PORT=1521)
  (https_proxy=www-proxy.mycompany.com) (https_proxy_port=80))
  (CONNECT_DATA= (SERVICE_NAME=sales.us.example.com)) )
```

例11 – セッション・データ・ユニット (SDU)

`salesserver1:1521/sales.us.example.com?sdu=16384`

以下の記述子を置き換えます。

```
(DESCRIPTION=
```

```
(SDU=16384)
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

例12 – 接続のタイムアウト

salesserver1:1521/sales.us.example.com?connect_timeout=60&transport_connect_timeout=30&retry_count=3

以下の記述子を置き換えます。

```
(DESCRIPTION=
(RETRY_COUNT=3) (CONNECT_TIMEOUT=60) (TRANSPORT_CONNECT_TIMEOUT=30)
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

例13 – DRCPプール接続クラス

salesserver1:1521/sales.us.example.com:pooled?pool_connection_class=*value*

以下の記述子を置き換えます。

```
(DESCRIPTION=
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)
(SERVER=POOLED) (POOL_CONNECTION_CLASS=value))
```

例14 – DRCP Pool Purity

salesserver1:1521/sales.us.example.com:pooled?pool_connection_class=*value*&pool_purity=new

以下の記述子を置き換えます。

```
(DESCRIPTION=
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)
(SERVER=POOLED)
```

例15 – DRCPでの暗黙的接続プーリング (STATEMENT境界)

salesserver1:1521/sales.us.example.com:pooled?pool_boundary=statement

以下の記述子を置き換えます。

```
(DESCRIPTION=
(ADDRESS=(PROTOCOL=tcp) (HOST=salesserver1) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)
(SERVER=POOLED) (POOL_BOUNDARY=STATEMENT))
```

例16 – mTLSを使用したOracle Cloud Autonomous Database (ADB) への容易な接続

アプリケーションは、セキュアな通信のために相互TLS (mTLS) を通じてOracle Cloud ADBへ接続できます。mTLSは、アプリケーションが実行されているクライアント側でデータベース・ウォレットを必要とします。このウォレットは、ADBクラウド・コンソールからzipパッケージの一部としてダウンロードされた `sso file` です。mTLS経由でOracle Cloud ADBへ接続するためのEasy Connect Plus構文のサンプルは次のとおりです。

```
tcps://hostname.region.oraclecloud.com:1522/abc_dbsvcname_high.oraclecloud.com?  
wallet_location=/path/to/walletdir&retry_count=20&retry_delay=3
```

ウォレットは、クライアント側の `/path/to/walletdir` ディレクトリに保存されます。

アプリケーションでEasy Connect Plus構文を使用することは、Oracle Cloud ADBの `sqlnet.ora` および `tnsnames.ora` ファイルを必要としないことを意味します。 `sqlnet.ora` および `tnsnames.ora` 構成ファイルでの同等の設定は以下のとおりです。

sqlnet.ora

```
WALLET_LOCATION = (SOURCE = (METHOD = file) (METHOD_DATA =  
(DIRECTORY="/path/to/walletdirectory")))  
SSL_SERVER_DN_MATCH=yes
```

tnsnames.ora

```
(description=  
(retry_count=20) (retry_delay=3) (address=(protocol=tcps) (port=1522) (h  
ost=hostname.region.oraclecloud.com)) (connect_data=(service_name=dbsv  
cname_high.adb.oraclecloud.com)) (security=(ssl_server_cert_dn="CN=co  
mmon_name.oraclecloud.com, OU=Oracle BMCS US, O=Oracle Corporation,  
L=Redwood City, ST=California, C=US")))
```

例17 – 一方向TLSを使用したOracle Cloud Autonomous Database (ADB) への容易な接続

アプリケーションは、セキュアな通信のために一方向TLSを通じてOracle Cloud ADBへ接続できます。最新バージョンのデータベース・クライアントでは組み込みトラスト・ポイントを使用でき (7ページ-注4)、ウォレットのダウンロードは必要ありません。一方向TLS経由でOracle Cloud ADBへ接続するためのEasy Connect Plus構文のサンプルは次のとおりです。

```
tcps://hostname.region.oraclecloud.com:1521/abc_dbsvcname_high.oraclecloud.com?retry  
_count=20&retry_delay=3
```

アプリケーションでEasy Connect Plus構文を使用することは、Oracle Cloud ADBの `sqlnet.ora` および `tnsnames.ora` ファイルを必要としないことを意味します。

`sqlnet.ora` および `tnsnames.ora` 構成ファイルでの同等の設定は以下のとおりです。

sqlnet.ora

```
SSL_SERVER_DN_MATCH=yes
```

tnsnames.ora

```
(description=  
(retry_count=20) (retry_delay=3) (address=(protocol=tcps) (port=1521) (h  
ost=hostname.region.oraclecloud.com)) (connect_data=(service_name=dbsv  
cname_high.adb.oraclecloud.com))
```

例18 – Easy Connect文字列を引用符で囲む

一部のツールや環境では、パラメータを引用符で囲むが必要な場合があります。

以下は、LinuxシェルでSQL*Plusを起動する例です。一重引用符と二重引用符を使用することで、シェルによって特殊文字が解釈されないようにしています。

```
$ sqlplus scott@'tcps://salesserver1:1521/sales.us.example.com?ssl_server_cert  
_dn="cn=sales,cn=Oracle Context  
Server,dc=us,dc=example,dc=com"&sdu=8128&connect_timeout=60'
```

まとめ

Oracle DatabaseのEasy Connect Plus構文を使用すると、データベース接続の管理に外部構成ファイルを使用しなくても、高可用でセキュアなアプリケーションを容易にデプロイできるようになります。

ORACLE CORPORATION

Worldwide Headquarters

500 Oracle Parkway, Redwood Shores, CA 94065 USA

海外からのお問い合わせ窓口

電話 + 1.650.506.7000 + 1.800.ORACLE1

FAX + 1.650.506.7200

oracle.com

CONNECT WITH US

+1.800.ORACLE1までご連絡いただくか、[oracle.com](https://www.oracle.com)をご覧ください。北米以外の地域では、[oracle.com/contact](https://www.oracle.com/contact)で最寄りの営業所をご確認いただけます。

 blogs.oracle.com/oracle

 facebook.com/oracle

 twitter.com/oracle

Integrated Cloud Applications & Platform Services

Copyright © 2024, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載されている内容は予告なく変更されることがあります。本文書は、その内容に誤りがないことを保証するものではなく、また、口頭による明示的保証や法律による黙示的保証を含め、商品性ないし特定目的適合性に関する黙示的保証および条件などのいかなる保証および条件も提供するものではありません。オラクルは本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクルの書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

OracleおよびJavaはOracleおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

IntelおよびIntel XeonはIntel Corporationの商標または登録商標です。すべてのSPARC商標はライセンスに基づいて使用されるSPARC International, Inc.の商標または登録商標です。AMD、Opteron、AMDロゴおよびAMD Opteronロゴは、Advanced Micro Devicesの商標または登録商標です。UNIXは、The Open Groupの登録商標です。0824

技術概要 Oracle DatabaseのEasy Connect Plus構成可能なデータベース接続構文Oracle DatabaseのEasy Connect PlusOracle DatabaseのEasy Connect Plus
2024年8月

 | Oracle is committed to developing practices and products that help protect the environment

ORACLE®