

Oracle Database 19cのEasy Connect Plus

構成可能なデータベース接続構文

ホワイト・ペーパー / 2019年12月5日

ORACLE®

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により、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.....	4
ホスト.....	5
ポート.....	5
サービス名.....	5
サーバー・タイプ	5
インスタンス名.....	5
パラメータ	5
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	6
説明パラメータ	6
例	8
結論	11

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。

Easy Connect Plusでは、外部構成ファイルの必要性を低減することで、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。たとえば、Oracle Database Cloud Serviceにアクセスする際に接続文字列にウォレットのロケーションを含めることで、セキュアな通信でTLSを容易に使用できるようになりました。

Easy Connect Plusは、JDBC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、C言語のOracle Call Interface (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の文字列には、次のようにホスト名、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。

```
database_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 = cx_Oracle.connect('scott', password,  
                               'mydbhost.example.com/orclpdb1')
```

Easy Connect Plusの構文

Oracle Database 19cの強化された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を使用し、Oracle Database 11.2以降に接続するOracle Databaseドライバ (JDBC、ODP.Net、cx_Oracle、node-oracledbなど) で使用できます。

Oracle Database 19c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_name=value{&parameter_name=  
value}]
```

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でサポートされるプロトコルの値は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。

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、次のように別のポートを使用することを明示します。

```
host1:port1,host2:port2
```

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・ホストで使用さ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、パラメータの始まりを表すために?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、説明の各パラメータを指定できます。

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。

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）です。

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側証明書を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。

Oracle Client 19.3以降、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が、サーバー証明書の共通ネーム（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。

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ソケットの受信バッファ・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値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。

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、利用可能な帯域幅を活用できます。

最初は、バッファ・サイズをBDP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、 $(40 * 1000 / 8) \text{ KB/秒} * (0.025) \text{ 秒}$ 、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。

SDU

Oracle Netセッション・データ・ユニットのパケット・サイズ（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。

SEND_BUF_SIZE

TCP/TCPソケットの送信バッファ・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値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。

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、利用可能な帯域幅を活用できます。

最初は、バッファ・サイズをBDPIに設定することが推奨されます。例については、RECV_BUF_SIZEを参照してください。

SOURCE_ROUTE

複数ホスト経由の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。これは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。

例

このセクションでは、Easy Connect Plus構文のさまざまな例を紹介するとともに、各文字列をどのように（tnsnames.ora構成ファイルで使用されるような）接続記述子に変換するかを説明します。

例1 - プロトコル

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

上記は以下の記述子に変換されます。

```
(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
```

上記は以下の記述子に変換されます。

```
(DESCRIPTION=
  (LOAD_BALANCE=ON)
  (ADDRESS=(PROTOCOL=tcp)(HOST=salesserver1)(PORT=1521))
```



```
(ADDRESS=(PROTOCOL=tcp)(HOST=salesserver2)(PORT=1521))  
(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
```

上記は以下の記述子に変換されます。

```
(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 – 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
```

上記は以下の記述子に変換されます。

```
(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5 – TLS接続

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

上記は以下の記述子に変換されます。

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

サーバーDNを一致させるために、SSL_SERVER_DN_MATCH=ONが完全な接続文字列に自動的に追加されます。接続に使用されるホスト名（この例ではsalesserver1）は、サーバー証明書のCNおよびSANフィールドと照合されます。

例6 – TLS接続

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

上記は以下の記述子に変換されます。

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

例7 - ウォレットのロケーション

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

上記は以下の記述子に変換されます。

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

例8 - 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"

上記は以下の記述子に変換されます。

```
(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9 - HTTPSプロキシ

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

上記は以下の記述子に変換されます。

```
(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10 - セッション・データ・ユニット

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

上記は以下の記述子に変換されます。

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

例11 - 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

上記は以下の記述子に変換されます。

```
(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12 – 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ことが必要な場合があります。

以下は、LinuxシェルでSQL*Plus 19c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'
```

結論

Oracle Database 19cのEasy Connect Plus構文を使用すると、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

オラクルの情報を発信しています

+1.800.ORACLE1までご連絡いただくか、oracle.comをご覧ください。北米以外の地域では、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 © 2019, 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。1219

ホワイト・ペーパー Oracle Database 19cのEasy Connect Plus構成可能なデータベース接続構成文Oracle Database 19cのEasy Connect PlusOracle Database 19cのEasy Connect Plus

2019年12月2019年12月 著者：[適宜入力]

力]

共著者：[適宜入力]

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

ORACLE®