特集3 ここがポイント! Oracle DB開発

Oracle meets .NET

第6回

透過的データ暗号化とクライアント識別子

DB側の新機能を極める

大田 浩
OTA, Hiroshi
日本オラクル株式会社
Oracle Direct テクニカルサービス部

はじめに

.NETからOracleデータベースの機能を活用する本連載。今回は、Oracle Database 10g Release 2（以下Oracle 10g R2）より追加された機能を、.NET環境から利用する方法について説明します。

Level

1 2 3 4 5

Technology Tools

- [x] Visual Basic
- [x] Visual C#
- [ ] Visual C++
- [ ] SQL Server
- [x] Oracle
- [ ] Access
- [ ] ASP.NET
- [x] Other:
  - Visual Studio .NET 2003
  - Oracle 10g Release 2 Enterprise Edition
  - Advanced Security Option

Oracle 10g R2ではデータベースそのものの機能はもとより、開発者にもメリットのある新機能が実装されています。Oracle 10g R2のリリースと同時に、.NETからOracleデータベースへ接続するためのミドルウェアであるOracle Data Provider for .NET（以下ODP.NET）も10.2へバージョンアップされており、.NETアプリケーションを開発する開発者はこのODP.NETを利用することにより、Oracle 10g R2の新機能をフルに利用したアプリケーションの開発が可能になります。

Oracle 10g R2による開発者のメリット

Oracle 10g R2ではさまざまな機能が追加されていますが、特に開発者にとってメリットのあるものに以下があげられます。

・Transparent Data Encryptionによる透過的なデータ暗号化

既存のアプリケーションを暗号化する際に改修の必要がない。新規開発でも暗号化を考慮せずに開発可能。また、アプリケーション側での暗号鍵の管理は不要。

・クライアント識別子のサポート

ODP.NETに新たに追加されたOracleConnection.ClientIdプロパティを利用することにより、コネクションを作成する際にクライアントの識別子を簡単にセットできるので、よりセキュアな環境を構築できる。また、Oracleデータベースの機能であるVirtual Private Database（VPD）の機能をより簡単に使用できる。

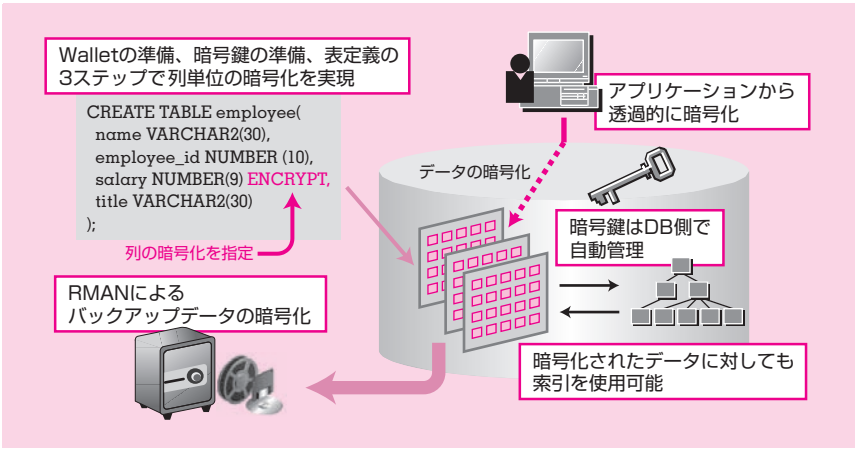
・.NETストアドプロシージャの利用

.NETストアドプロシージャからのデータアクセスの提供（Oracle Database Extensions for .NETで使用可能）。

・コネクション プーリングのReal Application Clusters（RAC）への最適化

ODP.NETでは、Oracleのクラスタ処理であるRACに対するコネクションプーリングの動作が最適化されている。RACへの接続要求があった場合、メトリックやデータベース上で設定されたポリシーに従い、自動的にコネクションを分配する。

図1：TDEを利用した暗号化



・LOBアクセスのパフォーマンスおよび機能の向上

Oracle 10g R2以降でのLOBのアクセスパフォーマンスおよび機能の向上。

・Database Change Notificationサポート

ODP.NETでは、Database Change Notificationをサポートする通知フレームワークが提供されたので、問い合わせの結果セットやスキーマオブジェクト、またはデータベースの状態に変更が生じた場合、その通知をアプリケーションが受け取れるようになった。Database Change Notificationを使用すると、アプリケーションでは、クライアント側のキャッシュ妥当性（ADO.NETのDataSetなど）を簡単に維持できる。

今回はこの中から以下の2つについて説明することにします。

・Transparent Data Encryptionによる透過的なデータ暗号化
・クライアント識別子のサポート

この2つはセキュリティに特化した機能になります。

今までセキュリティを強固にするアプリケーションを開発する際は、既存のアプリケーションの変更が必要となるため、なかなか実装するまでにはいたらないケースが多かったかと思います。今回は、既存のアプリケーションの修正をほとんど行なわずにデータベース側の機能だけでセキュリティを強固にする方法について説明します。

Transparent Data Encryptionによる透過的なデータ暗号化

Transparent Data Encryption（以下TDE）を利用することで、Oracleデータベースに格納されている情報を容易に暗号化することが可能です。本誌2005年10月号の本連載でも暗号化について説明しましたが、この方法でデータベースに格納された情報を暗号化するためにはアプリケーションを修正する必要がありました。

そのためには以下のPL/SQLパッケージを利用して、データの暗号化と復号を行なう必要があります。

・DBMS_OBFUSCATION_TOOLKITパッケージ（8.1.6EE～、9iからSEでも可）
・DBMS_CRYPTOパッケージ（10gの新パッケージ）

そのため暗号化／復号を行なう場合は、SQLの中にあらかじめ作成したプロシージャを埋め込む作業が必要になり、結果既存のアプリケーションを修正しなければなりません。

しかし、TDEを使用することにより、暗号鍵はOracleデータベース側で自動管理されアプリケーションから透過的に暗号化／復号が可能になります。

つまりTDEを利用するメリットとして、

・アプリケーションの変更が不要
・パフォーマンス劣化が最小限
・暗号化の設定が容易
・バックアップの暗号化（Recovery Managerによる暗号化）

があげられます（図1）。

TDEの利用

では、実際にTDEを使用してデータの暗号化を行なってみましょう。

TDEの機能を利用するには以下の設定が必要です。

・暗号鍵を管理するWalletの作成
・SQLNET.ORAファイルの修正
・表の列を暗号化

暗号鍵を管理するWalletの作成

暗号鍵を管理するためのWalletは、SQL*PLUSから直接SQLを発行して作成することも可能ですが、「Wallet Manager」というGUIツールを利用すればよ