

基于 Windows 的 Oracle 数据库 10g: 开发与部署

Oracle 技术白皮书
2004 年 7 月

ORACLE

基于 Windows 的 Oracle 数据库 10g: 开发与部署

执行概要.....	3
基于 WINDOWS 的 ORACLE.....	3
应用程序开发.....	4
Oracle Data Provider for .NET	6
XML 特性	7
性能	7
本地 Oracle 类型.....	8
其他主要特性	8
新特性.....	9
Oracle Objects for OLE.....	10
新特性.....	11
ADO 和 Oracle Provider for OLE DB.....	1
OLE DB .NET	13
新特性.....	13
ODBC.....	14
ODBC .NET.....	14
COM Automation 特性	14
应用程序部署.....	15
Oracle Services for Microsoft Transaction Server.....	16
可串行化事务	17
Oracle Fail Safe.....	17
逻辑/物理备用	18
多集群可管理性.....	18
安全性.....	18
Windows 注册表中的 Oracle Wallets	19
Microsoft Certificate Store Integration.....	19
目录	20
本地认证与 Active Directory.....	20
使用 Active Directory 的 Oracle Net Naming.....	21
结论	21

基于 Windows 的 Oracle 数据库 10g：开发与部署

执行概要

Oracle 数据库 10g 在 Windows 平台上引入了很多新功能，使得基于 Windows 开发和部署 Oracle 产品变得更加简单、快速和成本有效。

由于 Oracle 数据库 10g 在 Windows 数据访问方面引入了新功能，应用程序开发得到了很大改善。开发人员也具有了从多个 Windows 数据访问方法中进行选择，以满足自己需要的灵活性，这些方法包括 Oracle Data Provider for .NET (ODP.NET)、Oracle Objects for OLE (OO4O)、Oracle OLE DB、Oracle ODBC 和 COM Automation 特性。它们与 Microsoft Visual Studio 和 Microsoft Office 产品套件进行了完全集成。这些数据访问驱动程序在具有网络支持、新数据类型支持和增强的 XML 支持的 Oracle 数据库 10g 中性能得到了增强。

.NET 开发人员可以使用任何 .NET 语言，通过 ODP.NET 以最佳性能访问 Oracle 数据库。ODP.NET 是一个本地 Oracle .NET 数据提供程序，它提供了 .NET 环境中可用的最佳性能和大多数 Oracle 数据库功能。

基于 Windows 使用 Oracle 数据库 10g 的应用程序和中间层部署现在功能更加丰富，也更加集成。Oracle Services for Microsoft Transaction Server (MTS) 提供了分布式事务支持，其中新增包括了对于可串行化事务的支持。Oracle Fail Safe 为基于一个或多个 Windows 集群的 Oracle 软件提供了高可用性解决方案，同时提供了更好的集成和可管理性。Oracle 还集成了 Windows 2000 和 Windows Server 2003 本地安全性和 Active Directory，从而使得 Oracle 能够在异构操作系统和目录环境中运行。

基于 WINDOWS 的 ORACLE

Oracle 在 1993 年第一个具有了基于 Windows NT 的关系数据库，从此作出了承诺，能够提供基于 Windows 的最佳信息管理解决方案。从那时起，Oracle 技术针对 Windows 计算中的最新更改也逐步进行了调整，从开始早期的客户/服务器解决方案，发展到 Internet 应用程序，如今又发展为了网络技术。现在，Oracle 通过其世界级的数据库及其与操作系统基础技术的紧密集成，保持着在 Windows 平台上的领先地位。Oracle 具有超过

二十五年的数据管理专业技术，它是一个领先供应商，无论在公司内部网中还是 Internet 中，都具有基于 Windows 部署业务重要应用程序所需的技术、经验和服务。

面向 Windows 的 Oracle 数据库 10g 提供了运行一个企业所需的所有功能，无论用于部门部署还是公司范围的部署都可以。它允许用户利用 Windows 的成本有效和易于使用的优点，同时提供了 Oracle 一直以来所具备的可伸缩性、可靠性和性能。这个新的 Oracle 数据库还包含了 Windows 平台上应用程序集成的进一步增强。

Oracle 集中关注的两个领域为基于 Windows 的应用程序开发和部署。Oracle 数据库 10g 在面向 Windows 开发人员和集成人员的性能改善、使用简便和可管理性方面都有了明显的进步。

应用程序开发

.NET、COM 和 Win32 应用程序开发人员对于 Oracle 数据库具有多种不同的数据访问选择：Oracle Data Provider for .NET (ODP.NET)、Oracle Objects for OLE (OO4O)、OLE DB、ODBC 和 COM Automation 特性。 .NET 开发人员通过使用 ODP.NET 将获得优化的数据访问。 COM 开发人员则会通过使用 OO4O 获得最优化的数据访问。

使用 Oracle 产品的其中一个很大优点是，它支持多个编程标准。Oracle 通过支持 Java、.NET、COM 和 Win32 应用程序，确保了所有开发人员均能使用 Oracle 高级数据特性，从而为开发组织提供了真正的灵活性。Oracle 每个数据访问驱动程序的设计都可使性能最大，并且能够访问最新的数据库特性。

对于 Windows 开发，通常使用 .NET 和 COM 数据访问。Oracle 通过展示新的 Oracle 数据库 10g 特性，如网格支持和新数据类型，改善了这些数据访问方法的特性集，同时提供了更佳的性能和稳定性。

Oracle 通过它丰富的数据访问工具选择，使得组织能够采用最能满足其特性和性能需求的技术。开发人员通过使用他们最熟悉的数据访问方法，而不用学习新方法，因此能够更快地部署其数据库应用程序。

Oracle 数据库 10g 提供了一些 Windows 固有的数据访问方法。对于 .NET 开发环境，Oracle 使用 ODP.NET 提供了数据访问支持。ODP.NET 是一个本地 .NET 数据访问提供程序，可以直接访问 Oracle。从任何 .NET 语言都可使用该数据访问提供程序，其中包括 C# .NET、Visual Basic .NET 和 ASP.NET。另外，开发人员也可以使用 OLE DB .NET 和 ODBC .NET，但 ODP.NET 提供了最佳的性能，以及对于更多 Oracle 数据库特性的访问。ODP.NET 是为了允许 .NET 开发人员充分发挥 Oracle 数据库的功能而专门设计的。而 OLE DB .NET 和 ODBC .NET 则是为了最大化与较旧数据访问编程接口之间的互操作性和向后兼容性而设计的通用接口。它们允许用户分别使用 Oracle OLE DB 提供程序或 Oracle ODBC 驱动程序来访问 Oracle 数据库。这两个 .NET 数据访问方法和它们访问 Oracle 数据库的方式如图 1 所示。

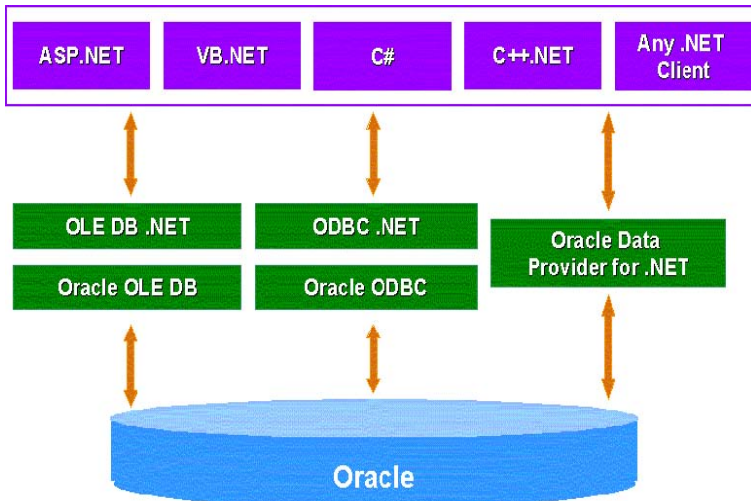


图 1: Oracle .NET 数据访问

ODP.NET 特性通过提供直接数据库访问，而无需附加的互操作性层，加快了数据库的性能。它比任何其他 .NET 提供程序都提供了更多的 Oracle 高级数据库功能。ODP.NET 10g 引入了很多新功能，这些新功能既利用了 Oracle 数据库 10g 中的新功能，也增强了 Oracle 数据库中的现有功能。因此，开发人员无论正在开发新的 10g 数据库应用程序还是正在增强旧式 Oracle 数据库服务器的性能，都将通过使用 ODP.NET 10g 而受益。

COM 编程在 Windows 开发人员中仍然非常受欢迎，尤其当增强旧式应用程序性能时更是如此。Oracle 为 COM 开发人员进行数据访问提供了三个不同的编程接口：Oracle Objects for OLE、Oracle Provider for OLE DB 与 COM Automation 特性。通过 Oracle ODBC 驱动程序还提供了更多的传统 ODBC 数据访问。OLE DB 和 ODBC 通常是通过自动层 Microsoft ActiveX Data Objects (ADO) 使用的。在 Oracle 数据库 10g 中，Oracle 为这些开发人员提供了一些新特性，可增强其基于 COM 的应用程序的性能，以充分利用新的数据库特性。这些 COM 数据访问方法和它们访问 Oracle 数据库的方式如图 2 所示。

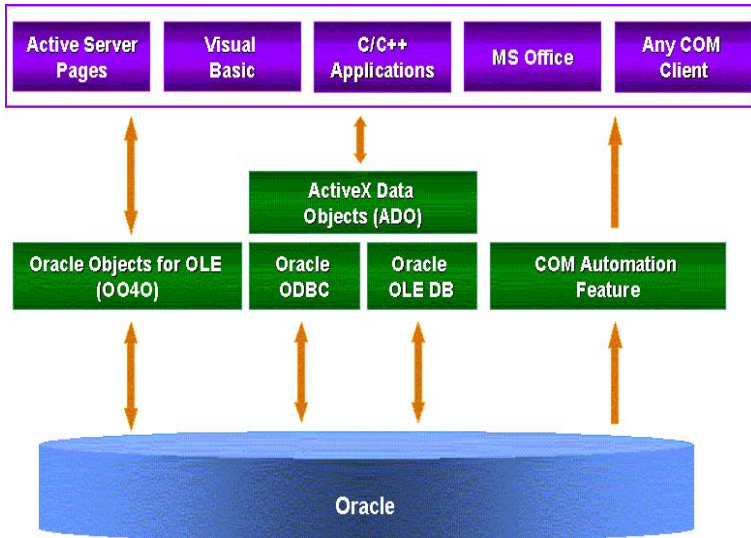


图 2: Oracle COM 数据访问

Oracle Data Provider for .NET

Oracle Data Provider for .NET 为从 .NET 访问 Oracle 提供了最高性能的方法。它支持 Oracle 的最新数据库性能, 包括 XML DB、本地新 Oracle 数据类型和网格功能。

与 OLE DB .NET 和 ODBC .NET 相似, Oracle Data Provider for .NET (ODP.NET) 也遵守 ADO.NET 标准。而与它们不同的是, ODP.NET 是一个本地驱动程序, 它不使用数据访问桥梁, 使用数据访问桥梁会降低性能。ODP.NET 的设计目的是为了提供最佳的性能和访问最多的数据库功能, 因为它是针对 Oracle 特定的数据源而开发的。因此, ODP.NET 与 OLE DB .NET 和 ODBC .NET 相比, 使得 Oracle 的使用变得更加灵活、更加快速、也更加稳定。

OLE DB .NET 和 ODBC .NET 是使用代码可移植性理念设计的通用接口, 而没有考虑程序员灵活性和数据访问性能方面的理念。对于开发灵活性, ODP.NET 的一个优点示例是它对于本地 Oracle 数据类型的支持, 如参考游标和 LOB。使用 ODP.NET 时, 这些 Oracle 类型的行为方式与其数据库相对应内容非常相似。这样就允许开发人员选择最符合其需求的数据类型, 从而提供了更好的灵活性。例如, .NET 中的参考游标会将结果集检索延迟到真正访问数据之前, 这就提供了一个性能优化。应用程序使用 LOB 数据类型可以选择在一个数据库回程中检索一个结果集中的所有 LOB, 一次只是一个结果集块, 还是只是 LOB 定位符, 从而将 LOB 检索延迟到访问 LOB 数据时。ODP.NET 对于性能调整提供了更好的控制, 充分利用了这些数据库特性。

ODP.NET 是通过提供对于 Unicode 以及本地和分布式事务的完全支持, 针对可伸缩的企业 Windows 解决方案而设计的。分布式 .NET 事务是通过 Oracle Services for Microsoft Transaction Server 得到支持的。

XML 特性

随着 XML 逐渐成为用于数据集成和 Web 服务的一种受欢迎语言，很多 .NET 程序员也逐渐在其应用程序使用该语言。ODP.NET 为程序员使用 XML 提供了两种方法：Oracle XML DB 和 .NET System.XML 服务。

XML DB 是 Oracle 的一种高性能、本地 XML 存储和检索技术，可在数据库服务器中使用。它在标准 W3C XML 数据模型下提供了一种独特功能，可同时存储和管理结构化数据和非结构化数据。XML DB 在 XML 和 SQL 术语之间提供了完全的透明性和可互换性。ODP.NET 使得所有 XML DB 功能均可用于 .NET 客户端，从而使得开发人员能够在数据库和 .NET 之间共享 XML 和对 XML 进行更改。这样就支持了到基于方案和不基于方案的 XML 的扩展，提供了用于不同编程需求的灵活性。另外，ODP.NET 包含了两个本地 XML 数据类型，OracleXMLType 和 OracleXMLStream，这样使得操作 XML 数据变得更加简单。XML 可在 XML DB 和 Microsoft 的 System.XML 服务之间轻松共享。

System.XML 是一系列用于从 .NET 数据提供程序操作 XML 数据集的接口。ODP.NET 与 System.XML 编程接口进行互操作，通过 ODP.NET DataAdapter 接口提供数据。XML DB 和 System.XML 之间的一个主要区别是，前者在数据驻留的位置提供 XML 服务，即数据库服务器上。而后者在客户端操作 XML。因此，ODP.NET 为程序员提供了很好的选择，可以选取最适合他们项目需求的 XML 技术。

性能

ODP.NET 与其他提供程序相比，一个主要区别在于它更好的基础性能和更多调整选项的可用性。正如前面讲述的那样，ODP.NET 不使用数据访问桥梁。在这些表面现象之下，还进行了大量优化，以确保对于 Oracle 数据源的快速 .NET 访问。另外，ODP.NET 针对特定的数据检索方案，具有很多可调整参数。开发的很多优化都是用于检索和操作 Oracle 本地类型的，如 LOB 和参考游标。

ODP.NET 具有很多调整选项。其中一个最常用选项为连接池，该选项对于具有大量用户的应用程序来说非常重要。ODP.NET 使用可调整设置创建连接池，其中包括连接生命时间和超时、最小和最大池大小，以及一次从该池增加或减少的连接数量。这些参数使得开发人员能够更好地控制其应用程序处理大量用户的方式，以及处理这些大量用户中更改的方式，因此可为最终用户提供更好的服务质量。

选择行集时，ODP.NET 可以指定每个回程返回的固定数据量。

在很多情况下，开发人员可能不需要马上在一个查询中具有所有数据。实际上，如果马上从大型查询返回所有数据的话，应用程序响应时间可能会减慢，令人不可接受。ODP.NET 使得开发人员能够通过指定每个服务器回程所获取行数量的方式，来设置查询的获取大小。在 LONG 和 LOB 数据类型中，也存在一个与之相当的特性，这些数据类型经常包含大量数据。这些类型的块大小可以进行设置，以指定一个回程中从每个特定 LONG 或 LOB 检索的数据量。这种块大小可以设置为无，一个固定的块大小，也可以设置为所有数据。这些调整选项为 .NET 开发人员提供了调整其数据检索的灵活性，从而优化了性能。

本地 Oracle 类型

Microsoft 在各种 .NET 编程语言之间引入了一个统一的数据类型集。使用 ODP.NET，Oracle 用户能够访问 .NET 数据类型也能访问 Oracle 本地类型。Oracle 类型完全可在 .NET 应用程序中进行操作，并且可与 .NET 数据类型进行互操作。Oracle 本地类型提供了很多用来在数据库中存储和操作数据结构的高级功能，这些数据结构如 XML 或 Microsoft Word 文档。即使对于常见类型，如 OracleDecimal (.NET 小数类型的对等类型)，Oracle 类型仍然在必要时提供了附加功能。在该 OracleDecimal 示例中，这种数据类型具有 38 位数字精确度，与 .NET 小数的 28 位精确度相比，提供了更高级的精确度。

ODP.NET 支持可在 .NET 环境中使用的所有高级 Oracle 类型，其中包括参考游标、LOB (CLOB、BLOB、NCLOB)、BFILE、LONG、RAW、LONG RAW 和 N-数据类型。使用其他提供程序的一个限制为，用户可能受限于数据类型功能。例如，在 ODP.NET 中，从存储过程作为参考游标输出参数返回的多个结果集都可任意访问。您可以在不检索第一个参考游标结果的情况下直接检索第二个参考游标的结果。而使用其他 .NET 提供程序时，数据必须以一种线性方法进行访问。第一个结果集中的数据必须首先检索，然后才能访问第二个结果集。当首先需要第二个结果集中的数据，并且需要较高性能时，这种情况可能就会非常令人讨厌。

其他主要特性

ODP.NET 提供了很多其他 Oracle 数据库特性，包括 PL/SQL、事务和 Unicode 支持。ODP.NET 用户可以在数据库完全执行 PL/SQL 存储过程和函数。PL/SQL 可以是打包的，也可以是不打包的，甚至可以是匿名 PL/SQL。匿名 PL/SQL 是一个有用的性能工具，因为它使得开发人员能够对一系列 SQL 语句进行批处理，并且能够在数据库回程中执行这些语句。

ODP.NET 可以参与 Oracle 数据库作为资源管理器的事务应用

程序。ODP.NET 在 .NET 环境中采用 Microsoft Enterprise Services 作为事务协调器。Oracle Services for Microsoft Transaction Server (OraMTS) 在 ODP.NET、Enterprise Services 和 Oracle 数据库之间充当代理的角色，以便协调这些事务。OraMTS 为 ODP.NET 程序员提供了丰富的体系结构，从而使得他们的事务应用程序能够保持高可用性和高可伸缩性。

ODP.NET 具有完全的 Unicode 支持，因此 .NET 用户可以轻松将其应用程序以多种书写语言进行全球化。这种全球化支持使得开发人员能够为多种文化/语言设置开发一个代码集。ODP.NET 全球化会抽取客户机的本国语言设置，以一种区域设置特定的格式显示信息。例如，设置为日语的浏览器会以日元显示货币。不用附加代码编写，这个应用程序就可以德语进行部署，以显示德国马克。这样就因为无需附加代码编写，使得在多个区域中部署应用程序变得更加简单也更加快速。

新特性

ODP.NET 的新特性包括透明数据库网络支持；对于基于方案和非基于方案 XML 的增强 XML DB 支持；以及改善的数据类型功能，这其中包括对于 PL/SQL 关联数组和 LOB 的 InitialLOBFetchSize 属性的支持。

ODP.NET 包含了一些新特性，可使得 .NET 开发变得更加简单，并且可以最大化利用 10g 功能。这些特性包括对于传递数组参数的支持、LOB 检索性能改善、从连接池中检查有效连接的能力、更多的 XML DB 支持、多个 Oracle 主目录 (MOH) 支持、增加了二进制浮点和双精度数据类型，以及网络支持。

在 ODP.NET 9.2.0.4 版本中，Oracle 引入了在数据库和 .NET 之间传递数组的能力。这样就通过允许大型数据集来回传递，使得 ODP.NET 的使用更加简便。ODP.NET 在数据库层使用 PL/SQL 关联数组向 .NET 数组传递数据。关联数组是对于在 Oracle 和 .NET 之间传递大型数据集的两种方法的补充，这两种方法是匿名 PL/SQL 和参数数组赋值。这些功能为开发人员提供了一些对其应用程序进行性能调整的选择，还减少了所需编写的代码量。

对于 LOB 应用程序，因为 LOB 数据可能非常大，所以性能通常是一个关键问题。如果一次只需一个 LOB 的一部分，Oracle 则可以根据需要检索 LOB 数据，同时创建多个数据库回程。对于具有小型 LOB 的应用程序，则进行多个这样的数据库回程的需要就会少一些。实际上，检索 LOB 的附加回程可能会比在一个回程中检索大量数据的速度慢一些。ODP.NET 在 9.2.0.4 版本中针对 OracleCommand 和 OracleDataReader 对象引入了 InitialLOBFetchSize 属性，以优化小型 LOB 的检索性能。如果 InitialLOBFetchSize 设置为大于零的值，则该 LOB 数据在一次回程中获取的数量为该属性中指定的字符或字节数。这样会明显加快 LOB 应用程序的速度。

ODP.NET 具有一个新的连接字符串属性 Validate Connection，该属性设置为真时，会验证来自连接池的数据库连接。很多情况

下都可能会引起无效连接，如重新启动数据库服务器时，连接池没有清除其预先存在的连接，那么此时这些连接将无效。默认情况下，此属性设置为假。验证对于来自该池的每个连接都需要一个数据库回程，这样会降低应用程序的性能。只有在连接池有效性检查的性能权衡可以接受的情况下，才应使用此属性。

ODP.NET 9.2.0.4 由于集成了 XML 功能和 Microsoft 的 XML 框架以及 Oracle XML DB，所以使得 XML 的使用变得更加简便。Oracle XML DB 是数据库服务器中一个非常强大的 XML 存储库功能。ODP.NET 可以将 XML 数据在数据库服务器中作为数据库本地类型 OracleXMLType 或 OracleXMLStream 进行本地传递。这些数据类型可以使得操作 XML 更加简单、更加快速。关系数据和对象关系数据可以作为 XML 直接从 Oracle 数据库实例访问到 Microsoft .NET 环境中。可以进行 XML 更改，并且这些更改可以作为 XML 数据保存回服务器中。在 ODP.NET 10g 中，除了支持不基于方案的 XML 之外，还包括了对于基于方案的 OracleXMLType 的支持。

ODP.NET 10g 的一个重要附加功能是对数据库网络的客户端支持。使用 Oracle 的网络技术时，无需任何 ODP.NET 更改即可使用。ODP.NET 10g 会自动识别何时连接了网络，并且完全支持数据库网络功能。ODP.NET 开发人员不必进行任何代码更改即可利用网络的功能。

添加到 ODP.NET 的所有这些新特性都关注性能改善和使用简便，从而使得开发人员的生产效率得以提高，还提供了更多用来优化应用程序性能的选项。

Oracle Objects for OLE

Oracle Objects for OLE (OO4O) 是基于 COM 的数据库连接工具，结合了优化的 Oracle 数据库访问以及易于使用的界面。OO4O 可用于从 web 应用程序到客户/服务器应用程序的多种环境。从任何支持 COM 的编程或脚本语言都可以虚拟访问 OO4O，如 Visual Basic、Visual C++、Microsoft Office、Active Server Pages 和 MTS/COM+。

由于 OO4O 是 Oracle 的本地驱动程序，它可为 Windows 客户端到 Oracle 数据库提供快速性能，而不会产生 ODBC 和 OLE DB 驱动程序的系统开销。OO4O 是专为使用 Oracle 数据库服务器而开发和改进的。它提供了轻松访问高级 Oracle 特性的方法，否则如果要从 ODBC 和基于 OLE-DB 的组件（如 ActiveX Data Objects (ADO)）来访问这些特性，将会很麻烦或者是不可访问的。

OO4O 由一个进程内 COM Automation Server、一个 C++ 类库以及 Oracle 数据控件组成。从体系结构上看，OO4O 作为

由于 OO4O 是 Oracle 的本地驱动程序，所以它不会增加 ODBC 和 OLE DB 驱动程序的系统开销。OO4O 提供程序提供了非常好的性能，具有最新的 Oracle 服务器特性。

Oracle Call Interface (OCI) 上面很窄的一层, OCI 则负责提供大量基础数据库访问管道。C/C++ 程序员可以直接使用 OCI。不过,许多 Windows 程序员都使用 OO4O,因为它使用方便,可以从诸如 Visual Basic 等脚本语言访问,而 OCI 则不行。

OO4O 允许访问 Oracle 所有主要的数据库特性,包括:

- 对 PL/SQL 的全面支持,包括无缝访问 PL/SQL 游标
- 可调的客户端、可滚动的以及可更新的游标
- 支持数组获取、更新和插入,从而减少网络回程
- 连接池技术,以增强中间层应用程序可伸缩性
- 多路复用性,允许用户会话共享一个 Oracle 网络连接,例如 web 环境
- 对于 MTS/COM+ 协调事务的完全支持,包括新的 Oracle Services for MTS 高可用性和可伸缩性体系结构
- 无缝访问高级 Oracle 数据类型:
 - 对象引用 (REF)、对象实例 (Object)、嵌套表、VARRAY、BLOB、CLOB、NCLOB 和 BFILE
- 描述模式对象的界面使用简单
- 全面支持对高级队列功能的访问
- 异步处理
- OO4O 的存储过程代码向导,可生成执行 PL/SQL 和 Java 存储过程的 VB 或 ASP 代码
- XML 支持
- 完整的 Unicode 支持使得应用程序得以全球部署
- 发布和订阅数据库事件,使得服务器端事件能够发起客户端操作,客户端事件也能发起服务器端操作。

新特性

OO4O 的新特性包括网格支持、日期时间、间隔、二进制浮点和二进制双精度数据类型,以及多个 Oracle 主目录支持。

Oracle 通过新增对于 Oracle 数据库特性的支持,继续增强了 OO4O 中的 COM 数据访问。OO4O 启用了网格;支持新的日期时间、间隔、二进制浮点和二进制双精度数据类型;支持 MOH。

与 ODP.NET 相似,OO4O 网格支持是透明的,使得 COM 应用程序能够在不进行任何 OO4O 代码更改的情况下使用此特性。数据库管理员可以轻松移植其数据库,以便使用 Oracle 的网格特性,而无需对其 OO4O 应用程序进行任何修改。

OO4O 包括了很多新增的对于 Oracle 最新数据类型的支持。日期时间和间隔数据类型使用所包括的更佳信息和更高精度,在现有日期数据类型基础上提供了改善。这些新的日期时间数据类型可以提供时区信息,精度最高可达十亿分之一秒。两个 Oracle 间隔类型则提供了年到月以及日到十亿分之一秒的信息,通常用于时间戳算术或比较。

二进制浮点和二进制双精度数据类型分别表示单精度和双精度

浮点值，这些值大多遵守用于浮点算术的 IEEE754 标准。在添加这些数据类型之前，Oracle 数据库中的所有数字值均以 Oracle 数字格式存储。这些新的二进制浮点类型不会替代 Oracle 数字。而是 Oracle 数字的备用类型，具有使用的磁盘存储更少的优点。

OO4O 可安装在以 Oracle 数据库 10g 开始的 MOH 中。但是，因为 OO4O 是一个 COM 组件，所以一次只能有一个实例处于活动状态。这就是说，后来的 OO4O 安装不会使得前面一个 OO4O 处于非活动状态。用户可以通过使用 Oracle Home Selector 切换为活动主机的 OO4O 版本，从而在各个 OO4O 客户端版本之间进行切换。

ADO 和 Oracle Provider for OLE DB

Oracle OLE DB provider 是为处理海量数据和高负载的 Internet 应用程序环境而提供的。通常可从任意的 COM 环境（如 Visual Basic）通过 ADO 来使用它。

Oracle Provider for OLE DB 为 ADO 开发人员提供了对于 Oracle 数据库的高性能和高效访问。OLE DB 是一个 Microsoft 数据访问规范，它使用一系列 COM 接口来访问和操作各种不同数据源中的数据。

Oracle OLE DB 提供程序是一个面向 Oracle 数据库的优化驱动程序，提供了高性能以及对于 Oracle 特定数据库特性的访问，如对于 LOB、PL/SQL 和参考游标的支持。所有这些 Oracle 特性和数据访问优化在其他第三方 Oracle OLE DB 提供程序和 OLE DB-ODBC Bridge 中通常是不可用的。由于非常接近 Microsoft OLE DB 规范，该 Oracle provider 使得 ADO 开发人员可以充分利用 OLE DB 的优点而无需牺牲高级 Oracle 功能。

Oracle Provider for OLE DB 是为 ADO 开发人员改进数据访问功能、性能以及可靠性而提供的。该 provider 支持下列通用的数据库特性：

- 可滚动和可更新的行集
 - 使用联合创建的可更新行集（在客户端游标引擎下可用）
- 本地和分布式事务
 - 分布式/MTS 协调事务使用新的 Oracle Services for MTS 高可用性和可伸缩性体系结构
- OS 认证，允许 Oracle 通过操作系统的安全性机制来授权用户
- 数据库链接
- 多模式行集
- ADOX
- 带选择语句的数据整理服务
- 完整的 Unicode 支持使得应用程序得以全球部署
- 该提供程序支持下列 Oracle 特定特性：

- PL/SQL
 - 返回多个行集的过程和函数
 - 打包和非打包的 PL/SQL
- LOB 和 N-数据类型
 - 带有存储过程的 LOB 参数
 - 带有 SQL 语句的赋值 NCHAR 参数
- Oracle 密码过期特性

OLE DB .NET

OLE DB 提供程序通过 OLE DB .NET 完全支持从任何 .NET 编程语言的 .NET 数据访问。

Oracle 数据库 10g OLE DB 提供程序使用 OLE DB .NET 接口支持 ADO .NET。OLE DB .NET 是 Microsoft 提供的、.NET 和 OLE DB 之间的互操作桥梁。它与其他 .NET 数据访问接口共享相似的编程接口，如 ODBC .NET 和 SQL Server .NET 数据提供程序。

Oracle OLE DB 完全遵守 OLE DB .NET 规范。它使得程序员能够执行存储过程和访问 LOB。无论 Windows 开发人员喜欢使用 COM 还是 .NET，Oracle OLE DB 的功能在两个实例中均完全受支持。Oracle OLE DB/OLE DB .NET 支持所有 .NET 语言，从而使得任何 .NET 开发人员都能充分利用 Oracle 数据库。

新特性

Oracle OLE DB 的新特性，包括网格支持、日期时间、间隔、二进制浮点数据类型和二进制双精度数据类型，以及多个 Oracle Home 目录支持。

增加到 Oracle OLE DB 的新数据访问特性与增加到 OO4O 的新特性相似。Oracle OLE DB 启用了网格、支持新的日期时间、间隔、二进制浮点数据类型和二进制双精度数据类型、支持 MOH。

与其他 Windows 数据访问驱动程序相同，OLE DB 网格支持是透明的，使得 COM 应用程序能够在无需任何 ADO/OLE DB 代码更改的情况下使用此特性。数据库可以进行移植，以使用 Oracle 的网格特性，而不必对 ADO 应用程序进行任何更改。

Oracle OLE DB 还新增包括了对于 Oracle 日期时间和间隔数据类型的支持。这些数据类型与 Oracle 的日期数据类型相比，包括更多信息，提供更高的精度。这些新的日期时间数据类型提供的精度可达十亿分之一秒的时区信息。两个 Oracle 间隔类型提供了对时间戳记差异进行比较时的年到月和日到十亿分之一秒的数据。

二进制浮点数据类型和二进制双精度数据类型分别表示单精度和双精度浮点值，这些值大多遵守用于浮点算术的 IEEE754 标准。在添加这些数据类型之前，Oracle 数据库中的所有数字值均以 Oracle 数字格式存储。这些新的二进制浮点类型不会替代 Oracle 数字。而是 Oracle 数字的备用类型，具有使用更少的磁

盘存储的优点。

Oracle OLE DB 可安装在以 Oracle 数据库 10g 开始的 MOH 中。但是，因为 OLE DB 是一个 COM 组件，所以一次只能有一个实例处于活动状态。这就是说，后来的 OLE DB 安装不会使得前面一个 OLE DB 处于非活动状态。用户可以通过使用 Oracle Home Selector 切换为活动主机的 OLE DB 版本，从而在各个 OLE DB 客户端版本之间进行切换。

ODBC

传统的 Windows 应用程序可以使用 Oracle 的最新 ODBC 驱动程序访问 Oracle 数据库服务器，ODBC 驱动程序提供了一些新特性和更好的性能。

Oracle ODBC 驱动程序允许 Windows 应用程序以更传统的方式访问 Oracle 数据库服务器。Oracle ODBC 的版本为 3.51，2 级兼容。它提供了访问 Oracle 的高性能方法。在 Oracle ODBC 中最新引入的新功能包括：

- 扩展的 Unicode 支持
 - Unicode 数据类型
 - 以 Unicode 编码的 SQL 语句
- 多个 Oracle Home 目录支持
- ODBC 目录功能中的性能改善
- 带有 Oracle Services for MTS 的更好可伸缩性和可用性

ODBC .NET

Oracle ODBC 提供程序通过 ODBC .NET 完全支持从任何 .NET 编程语言的 .NET 数据访问。

Oracle ODBC 驱动程序使用 ODBC .NET 接口支持 ADO .NET 数据访问。与 OLE DB .NET 相同，ODBC .NET 是 Microsoft 提供的、.NET 和 ODBC 之间的互操作桥梁。它与其他 .NET 数据访问接口共享相似的编程接口，如 OLE DB .NET 和 SQL Server .NET 数据提供程序。

Oracle ODBC/ODBC .NET 程序员在 C/C++ 中能够与 ODBC 用户访问相同级别的功能，其中包括执行存储过程和访问 LOB。因此，Windows 开发人员可以从 .NET、COM 和 Win32 使用 Oracle ODBC 的完整功能。Oracle ODBC/ODBC .NET 支持所有 .NET 语言，从而使得任何 .NET 开发人员都能使用 Oracle 数据库。

COM Automation 特性

Oracle COM Automation 特性使 PL/SQL 和 Java 存储过程、函数以及触发器可以访问 COM Automation 服务器，例如 Microsoft Office。这个特性使得服务器修改客户端应用程序（如在 Microsoft Excel 中创建图形或者通过 Microsoft Exchange 发送

Windows 平台上的 Oracle 客户通常很熟悉 COM Automation 解决方案。COM Automation 允许通过 IDispatch 接口调用 COM 对象。在其他应用中，它主要在 Visual Basic 编程中使用，提供一种通过 Microsoft Office 的 Automation 接口来访问其应用程序和数据的手段。

COM Automation 特性提供了 PL/SQL 和 Java 程序包，允许存

存储过程开发人员调用 COM Automation 服务器。存储过程程序员现在可以访问并操作任何 COM Automation 服务器(如 Microsoft Office)。例如, COM Automation 特性可设计为通过一个 Microsoft Excel 电子数据表, 自动编辑并图形化显示位于 Oracle 数据库中的销售数据。或者它也可以使用 Microsoft Exchange 自动发送这些数据给相关的销售经理们。事实上, COM Automation 特性并不仅限于调用 Microsoft COM 对象, 它可以用于操作任何 COM Automation 服务器或自定义的 COM Automation 对象。

COM Automation 特性提供了几个主要的优点:

- 简单的 API 使 PL/SQL 和 Java 开发人员易于将 COM Automation 对象合并到他们的程序中
- 通过 COM Automation 特性调用预先存在的 COM Automation, 可以在 Oracle 解决方案中利用这些对象
- Oracle 数据库可推动与其他 Windows 应用程序的集成

这些 PL/SQL 和 Java 程序包展示了实例化 COM 对象, 获取和设置其属性以及调用其方法的 API。开发人员可以从 PL/SQL 和 Java 的子程序、存储过程、存储函数以及触发器调用这些 API。其对 COM 对象所处的位置没有任何限制。它们可以与数据库服务器一起位于本地, 也可以通过分布式组件对象模型 (DCOM) 远程访问。

应用程序部署

Oracle 提供了很多不同的产品, 它们将各种 Oracle 应用程序集成到了 Windows 环境中, 其中包括 Oracle Services for MTS、Oracle Fail Safe、Windows 安全集成和 Active Directory 集成。

在 Windows 上部署应用程序需要与 Windows 自身的服务和中间层服务器密切集成在一起, 以确保应用程序正确的互操作性。Oracle 数据库 10g 提供了这种紧密集成, 使得组织能够充分利用核心操作系统和服务器功能, 以及 Oracle 的高级特性集。Oracle 的集成确保透明的互操作性是可伸缩的、可用的和安全的。通过使用这种方法, 公司可集中资源建立它们应用程序的业务逻辑, 而不是用来纠正产品在 Windows 上的不兼容或低效率。

Oracle 数据库 10g 包括很多工具集, 从而确保了优化的应用程序部署环境。Oracle Services for Microsoft Transaction Server 是一个可使得事务应用程序性能更好、可伸缩性和可用性更好的特性设计。在 Windows 集群上部署任务关键 Oracle 服务器时, Oracle Fail Safe 通过利用 Microsoft Cluster Service 提供丰富的故障切换支持, 从而确保了这些服务器总是保持可用。Oracle 集成了本地 Windows 安全和目录工具, 这两种工具都改善了数据库安全性, 使得用户管理变得更加简单。

Oracle Services for Microsoft Transaction Server

使用 Oracle Services for MTS, 每个 MTS 服务器都可以承载它自己的 Oracle Services。这样就消除了 Windows 应用程序的单点故障, 从而改善了可用性和性能。

很多 Windows 平台上的 Oracle 客户都在中间层中使用 Microsoft Transaction Server (MTS)。MTS 是一个在分布式环境, 用于 COM 对象和事务的应用服务器。MTS 的内核是分布式事务协调器 (DTC), 负责协调分布式资源管理器间的事务。在 Windows 2000 中, MTS 执行已被增强的 COM+ 服务取代。随着 .NET 的引入, 它现在成为 .NET Enterprise Services 的一部分。

Oracle Services for Microsoft Transaction Server 将 DTC 与 Oracle 数据库集成在一起。该服务作为代理, 使客户可以在 DTC 协调的事务中使用 Oracle 数据库作为资源管理器。具体来说, 该服务可提供下述事务操作:

- 获取 (Enlistment) — 对全局事务的上下文维护
- 完成 — DTC 和 Oracle 之间两阶段提交调用转换
- 恢复 — 解决未决 DTC 事务
- 连接池 — 事务数据库连接资源的缓冲

在 Oracle 数据库 10g 中, Oracle Services for MTS 具有一个专门为性能、可伸缩性和可用性而设计的体系结构。每个 MTS 服务器都保存着它自己的资源管理器代理缓存, 因此任何数据库都可能具有与之相关联的很多代理。这种代理缓存由 Oracle Services for MTS 集成层维护, 直接在 MTS 服务器进程内运行。该集成层负责处理应用程序、DTC 和数据库之间事务的获取和完成。

在多个 MTS 部署中, 该模型将带来更好的可用性, 因为不会有代理造成单点故障。而且, 该模型会使得整体性能得以提高, 因为没有任何代理可能会成为数据库的瓶颈。在 MTS 进程中具有 Oracle Services for MTS 集成层, 减少了附加进程通信, 这种通信比进行进程内调用的速度要慢很多。

每个 MTS 服务器都运行一个恢复后台程序。恢复是由数据库驱动的, 数据库通过其重新获取请求可疑事务处理的最终结果。PL/SQL 程序包定期运行一个恢复作业, 该作业与可疑事务相关联的恢复后台程序进行联系。该后台程序从 DTC 请求事务结果并将该结果通知数据库。在安装时将创建一个名为 MTSSYS 的数据库管理帐户, 用于 DTC 恢复。

Oracle Services for MTS 允许使用很多数据访问接口开发应用程序, 其中包括 ODP.NET、OO4O、OCI、ADO/OLE DB 和 ODBC。通常情况下, ODP.NET、OO4O 和 OCI 可以提供更好的性能以及与 Oracle 之间的兼容性。ODP.NET、OO4O 和 OCI 使用 Oracle 资源分散器提供的优化连接池。Oracle Provider for OLE DB 和 Oracle ODBC 驱动程序则使用 Microsoft 资源管理来进

行数据库连接。

尽管 Oracle Services for MTS 自身只能运行在 Windows 上，但却可以操作任何操作系统上的 Oracle 数据库。这种情况对使用 Windows 中间层来访问基于 UNIX 的 Oracle 服务器的用户来说，是再理想不过了。

可串行化事务

Oracle 数据库 10g Services for MTS 完全支持具有可串行化事务。

在 Oracle 数据库 10g 中，OraMTS 增加了对于具有可串行执行事务的支持。OraMTS 用户可以选择使用 Read Committed 和可串行执行隔离级别。具有可串行化事务会模拟串行事务执行，就好像事务在一个接一个串行执行一样，而不是并发执行。在绝大多数情况下，事务在 Read Committed 隔离级别的性能最佳。但是，有些应用程序需求可能需要可串行执行。

Oracle Fail Safe

通过集成 Microsoft Cluster Service, Oracle Fail Safe 确保了 Oracle 软件在 Windows 上保持高可用性。Fail Safe 支持 Oracle 数据库、应用服务器以及电子商务套件。

Oracle Fail Safe 是 Oracle 数据库 10g 的核心特性，它为在 Windows 集群上部署的任何关键解决方案提供了高可用性。集群消除了单个主机系统作为单故障点的情况。Oracle Fail Safe 与 Microsoft Cluster Service (MSCS) 共同运行，确保了在一个集群节点发生故障的情况下，该节点上运行的 Oracle 数据库和应用程序会自动并且快速地故障切换（转移）到仍在运行的节点上。

对可以由单个系统来处理数据库和应用程序负载的 Windows 客户而言，Oracle Fail Safe 是最适合的。Oracle Fail Safe 解决方案可在所有商用 Windows NT、Windows 2000 和 Windows Server 2003 集群上部署。它在 Windows Server 2003 Datacenter 上最多可支持八节点集群，这是如今 Windows 上可用的最大节点配置。支持的产品包括：

- Oracle 数据库 (Oracle10g、Oracle9i、Oracle8i 和 Oracle8)
- Oracle Applications release 11i
- Oracle iAS，但是缓存产品除外，其中包括
 - Oracle Forms Server
 - Oracle Reports Server
 - Oracle HTTP Server
- Oracle Services for Microsoft Transaction Server
- Oracle Intelligent Agent
- 作为 Windows 服务安装的应用程序

SAP、Baan、PeopleSoft、Lawson、J.D. Edwards 以及其他应用程序供应商，也都已使用 Oracle Fail Safe 验证了他们的软件解决方案。

Oracle Fail Safe 包括两个主要组件，服务器和管理器。服务器组

件，即 Oracle Services for MSCS，与集群软件一起确保在计划或意外停机期间进行快速自动的故障切换。管理器，即 Oracle Fail Safe Manager，具有一个使用方便的图形化界面，它与一个或多个集群上的 Oracle Fail Safe Server 一起执行配置、管理、验证和静态负载均衡等工作。同时，这些组件还提供了一组丰富的特性以及集成的问题诊断工具，可以快速部署高可用的数据库和应用程序 — 即完整的电子商务解决方案。

逻辑/物理备用

Oracle Fail Safe 通过实现逻辑/物理备用支持，支持灾难保护。至于改善的可管理性，现在可以从同一个控制台管理多个集群。

使用逻辑/物理备用集成，Oracle Fail Safe 数据库可在一个主站点和任何远程备用站点上运行。Fail Safe 可以监视数据库，可以使其联机，也可以使其脱机。这使得客户可以在集群内在最短的停机时间进行单个实例的故障切换，同时维护备用站点也变得更有利。Oracle Fail Safe 通常用于有计划的故障切换，诸如 Windows 环境全面的软件/硬件的升级。使用逻辑/物理备用，Fail Safe 现在提供了由备用提供的真正灾难保护。总而言之，这两个产品互相补充，与每个单独产品相比，都提供了更高的可用性和灾难保护。

多集群可管理性

对于具有多集群部署的客户，Oracle Fail Safe 管理器允许显示多个集群，从而使得用户能够从一个窗口管理所有集群，而不用从每个集群的单独窗口中进行管理。例如，管理员能够使用一个向导，在所有 Fail Safe 集群中一次性完成所有数据库帐户凭据的修改。管理员将能够避免单独对策略进行比较，或者修改集群显示或警报首选项的烦杂特性。

安全性

Oracle 数据库支持一些 Windows 安全机制，其中包括一次性登录、Windows Kerberos 和 Windows SSL。在 Windows 注册表上存储 Oracle wallets 的功能消除了不安全的操作系统文件存储。Oracle 与 Microsoft Certificate Store 的集成使 Oracle PKI 和 Windows PKI 应用程序可以实现无缝互操作。

Oracle 数据库 10g 与本地 Windows 安全模型进行了紧密集成，可更好地保护应用程序的安全。在客户/服务器环境中，如今对于希望利用 Windows 凭据或认证服务登录 Oracle 数据库的站点来说，有三个选择。

- Oracle 数据库包括 Windows 本地认证适配器，该适配器是自动随 Oracle Net Server 和 Client 一起安装的。此特性通过使用 Windows 用户凭据来支持数据库认证，使得数据库用户具有了一次性登录功能。要使用此特性，Windows 用户必须定义为数据库外部用户。在该数据库可以使用的 Microsoft Active Directory Services 中，可以为这些用户分配外部用户或角色。
- Oracle Advanced Security，这是一个数据库选项，使用 Microsoft Key Distribution Center (MSKDC) 颁发的 Kerberos 票证支持数据库认证。此功能使得已经在 Windows 环境中颁发了有效 Kerberos 票证的用户能够登

录他们的数据库帐户，而不必提供用户名/密码。此功能可存在于多平台环境中，因此数据库服务器可以基于 Windows、Linux 或 Unix 服务器（举例）。

- Oracle Advanced Security 选项还使用 X509v3 证书支持基于安全套接字层 (SSL) 的数据库认证。Microsoft Certificate Store (MCS) 可以颁发此证书。要使用此特性，该证书必须包含在客户端上配置的 Oracle Wallet 中。Oracle Wallet 可能存储于 Windows 注册表的用户概要文件区，也可能存储于客户端上的任意文件位置中。此功能也可存在于多平台环境中，因此数据库服务器可以基于 Windows、Linux 或 Unix 服务器（举例）。

对于 Web 环境，与 MCS 的集成可以通过 Oracle 一次性登录实现。除了支持基于密码的认证机制之外，Oracle 应用服务器通过 Oracle 一次性登录服务器还支持基于证书的认证。此证书可由 MCS 生成。要使用基于证书的认证，必须将用户证书填充到浏览器钱夹或 Oracle Internet Directory 中相应的用户条目中。

Windows 注册表中的 Oracle Wallets

Oracle wallets 可以存储在 Windows 注册表中，为 Windows 客户端上的 Oracle wallets 提供了增强的安全性。如果没有这个特性，Oracle wallets 通常存放在 Windows 文件系统中。如果操作系统的文件权限是不安全的，那 Oracle wallets 也就不安全了。不安全的文件权限可能是由于管理不正确或者缺少操作系统文件安全性引起的。因此在 Windows 系统上，通过在 Windows 注册表中用户配置文件区中存储 Oracle wallets，改善了公钥基础设施 (PKI) 安全性。只有正确登录的用户才能访问该注册表区域。可以使用多个 Oracle wallets，并存储在该配置文件区。

通过 Oracle Wallet Manager 和 Oracle Enterprise Login Assistant 工具，可启用 Oracle PKI 应用程序的一次性登录。Wallet Manager 创建加密 Oracle wallet，而 Enterprise Login Assistant 则创建解密 wallet。然后 Oracle PKI 应用程序使用该解密 wallet 进行 SSL 认证。这两个工具都经过改进，支持 Oracle wallets 在注册表中的存储和位置，如果需要也支持在默认文件系统的存储和位置。

Microsoft Certificate Store Integration

Oracle 与 Microsoft Certificate Store 的集成使 Oracle PKI 应用程序可以使用 Windows PKI 的产品进行互操作。当该特性被启用时，Oracle PKI 安全性就使用 Microsoft CryptoAPIs 来访问 Microsoft Certificate Store。CryptoAPI 用于操作，诸如签名、加密、解密、验证和确认。Wallet Resource Locator (WRL) 确定 PKI 类型以及其他相关信息。

使用此特性，Oracle 和非 Oracle 应用程序就可以利用同一组 PKI 认证（诸如证书、密钥、吊销列表以及信任点）进行公共密钥安全性服务，如验证和加密等。

目录

Oracle 的目录集成和供应平台使得 Oracle Internet Directory (OID) 能够与其他目录一起集中进行同步和提供，其中包括 Active Directory。Oracle 通过允许用户映射存储在 Active Directory 或 OID 中，使得一次性登录成为了可能，用户管理也变得更加简单。而且，Net naming (tnsnames.ora) 现在可以存储在 Active Directory 或 OID 中，以便进行集中管理。

通过采用集中管理，目录服务器通常都可简化用户和资源管理。Oracle Internet Directory (OID) 使得企业能够将其目录服务器进行集中，并且可以利用其现有目录服务器，如 Active Directory。OID 的目录集成和供应平台由一系列构建到 OID 的服务和接口组成。这些都简化了该目录和其他存储库之间同步和提供解决方案的开发。这些其他存储库可能包括其他目录，如 Microsoft Active Directory Services、应用程序用户存储库或包含人力资源信息的数据库表。

此平台包括两个服务：供应服务和同步服务。如果目录中的用户或组条目被更新或删除，该供应服务则简化了应用程序特定用户供应活动的自动执行。这些活动是作为 PL/SQL 过程执行的，可在连接系统中添加、删除或挂起用户的权限。不同的 Oracle 产品组件使用该供应引擎可以自动为该目录中管理的用户创建“帐户覆盖区”。

该同步服务则会将 Oracle Internet Directory 所包含数据的更改发布到订阅了特定目录数据集的连接代理中。它还会读取并应用连接系统中所作的更改。这些内容一起简化了 Oracle Internet Directory 和连接系统中所包含数据的“同步”，这样的连接系统如 Microsoft Active Directory Services。

Oracle 还包括了两个附加的本地集成特性，这些特性也会同时在 Windows 上利用 OID 和 Active Directory (AD)：

- 本地认证和 AD
- 带有 AD 的 Oracle Net naming

本地认证与 Active Directory

Oracle 客户通常要求企业级的安全性和模式管理。对具有大量用户的业务来说尤其如此。OID 的设计目的在于处理这些企业环境并与部门目录服务器进行互操作，如 Active Directory。Oracle 安全和管理与这两个目录服务器都进行了集成，以提供无缝访问和管理。但是，还有一些方式可用来在没有 OID 的情况下从数据库直接集成到 Active Directory。

通过 Windows 的认证机制，Oracle 提供本地认证，操作系统可以对用户执行 Oracle 数据库识别。Oracle 在 Windows 上支持 Kerberos 和 SSL 安全协议。启用了本地认证的情况下，用户通过登录 Windows 即可利用一次性登录访问 Oracle，从而简化了最终用户管理，消除了冗余的安全凭据。

Oracle 外部用户映射允许 Windows 用户作为 Active Directory 用户访问 Oracle 数据库。这些外部用户和角色映射可以存储在一个用作 Oracle 中心存储库的 Active Directory 中。

使用 Active Directory 的 Oracle Net Naming

Oracle 通过 OID 和 Active Directory 利用 LDAP 技术来改善数据库连接信息管理。一般来讲，最终用户使用 Net-style(样式)名称来引用数据库，该名称通过 TNSNAMES.ORA 配置文件进行解析。此文件必须在访问 Oracle 数据库的每个客户机上都进行管理。使用 Oracle, Net 名称可以通过 Active Directory 进行存储和解析，而无需 OID。将这些信息集中在目录中既消除了管理负担，又免去了用户配置各自的客户机的工作。

而且，Windows 2000 和 Server 2003 工具(如 Windows Explorer 和 Active Directory 用户和计算机)也可能解析数据库连接和测试连接。Oracle 工具也得到了增强。Oracle 数据库 Configuration Assistant 可以在 Active Directory 中自动注册数据库对象。同时 Oracle Net Configuration Assistant 会在该目录中注册网络服务对象。这些增强技术进一步简化了管理。

结论

Windows 应用程序开发与部署的技术已经从客户/服务器模式发展到 Internet 模式，Oracle 也是如此。Oracle 数据库 10g 数据库为客户提供了前所未有的灵活性，他们可以选择符合其需求的应用程序工具和体系结构，而不用牺牲性能、可伸缩性、使用简便性和安全性。Oracle 对于 Windows 平台的承诺包括与 .NET 基础架构的新增集成。通过带有 Windows 理念设计 Oracle 数据库 10g，客户软件应用程序可以获得操作系统和数据库之间的无缝互操作性。

有关基于 Windows 的 Oracle 详细信息，请访问 <http://otn.oracle.com/windows>, <http://www.oracle.com/windows>



基于 Windows 的 Oracle 数据库 10g：开发和部署
2004 年 7 月
作者：Alex Keh

Oracle Corporation
全球总部，
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

全球咨询热线：
电话：+1.650.506.7000
传真：+1.650.506.7200
www.oracle.com

Oracle Corporation 提供推动互联网发展的软件。

Oracle 是甲骨文公司的注册商标。本文中提及的各种产品和服务的名称可能是甲骨文公司的商标。其他所有提及的产品和服务名称可能是各自所有者的商标。

版权所有 © 2004 甲骨文公司
保留所有权利。