



Programming with Oracle Developer

Oracle Developer 编程指南

(美) Matthew Bennett 著 武欣 罗云峰 刘侃 等译

TP311.138

59

Oracle 技术系列丛书

Oracle Developer 编程指南

(美) Matthew Bennett 著

武欣 罗云峰 刘侃 等译



机械工业出版社
China Machine Press

本书是一本利用 Oracle Developer 创建高性能应用程序的完全手册。它是 Oracle 开发人员必备参考书。本书介绍了 Oracle Developer Forms 组件和 Reports 组件的特性和体系结构、Developer 的安装问题、Developer 套件的基本组成模块、Developer 的编程元素、Developer 应用程序的创建以及如何将 Developer 应用程序移植到 Web 环境中。最后,介绍了项目管理和源代码控制的内容以及 Developer 应用程序开发的准则。

Matthew Bennett: Programming with Oracle Developer.

Authorized translation from the English language edition published by Sams, an imprint of Macmillan Computer Publishing U.S.A.

Copyright © 2001 by Sams.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2002 by China Machine Press.

本书中文简体字版由美国麦克米兰公司授权机械工业出版社独家出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

版权所有,侵权必究。

本书版权登记号:图字:01-2001-4761

图书在版编目(CIP)数据

Oracle Developer 编程指南/(美)本奈特(Bennett, M.)著;武欣等译.-北京:机械工业出版社,2002.4

(Oracle 技术系列丛书)

书名原文:Programming with Oracle Developer

ISBN 7-111-09982-6

I. O... II. ①本...②武... III. 关系数据库-数据库管理系统, Oracle-程序设计 IV. TP311.138

中国版本图书馆 CIP 数据核字(2002)第 014182 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:武恩玉

北京忠信诚胶印厂印刷·新华书店北京发行所发行

2002 年 4 月第 1 版第 1 次印刷

787mm×1092mm 1/16·15.25 印张

印数:0 001-4 000 册

定价:30.00 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

译者序

Oracle 是世界著名的数据库软件供应商，也是世界上最大的电子商务软件供应商之一。Oracle 数据库在许多国家和地区得到广泛的应用。Oracle 在世界进入互联网时代后，推出了用于开发基于 Web 的应用程序的 Oracle Developer 套件，该套件包括 Graphics Builder（图形构造器）、Procedure Builder（过程构造器）、Project Builder（项目构造器）、Query Builder（查询构造器）、Schema Builder（模式构造器）、Forms Builder（表单构造器）和 Reports Builder（报表构造器）等组件。

本书以介绍 Oracle Developer Forms 组件和 Reports 组件的特性和体系结构为开始。依次介绍了 Developer 的安装问题、Developer 套件的基本组成模块、Developer 的编程元素以及 Developer 应用程序的创建、如何将 Developer 应用程序移植到 Web 环境中。最后，介绍了项目管理和源代码控制的内容，并以介绍 Developer 应用程序开发的准则为结束。

在学习了本书后，本书的读者将：

- 掌握 Web 应用程序的设计和原型设计，这包括 Developer 的表单、报表和图形所提供的新向导和 GUI 设计特性。
- 扩展应用程序的可重用性并优化应用程序的安全性。
- 部署 Internet 应用程序并将它们集成到其他工具中。
- 创建健壮、基于 Web 的 Oracle 应用程序。

本书是一本利用 Oracle Developer 创建高性能应用程序的完全手册。它是 Oracle 开发人员必备参考书。本书内容深入浅出、分析透彻、概念清楚、实例丰富，实用性强，适用于 Oracle 数据库应用程序开发的中、高级人员。

本书主要由武欣、罗云峰、刘侃翻译。参与翻译工作的还有武斌、余勇、邓一凡、宁靖、廖小锋、敖晓芳等同志。本书由武欣同志统一审校。在此一并表示感谢。

2002 年元旦

前 言

本书是我在 Oracle 工作经验的结果。Oracle Developer 已经发行了许多不同版本，但是其基本原理还是相同的。在解释所有新特性时陷入迷茫与忘记基础知识一样很容易发生。但愿本书所包含的内容能够满足你的需求和兴趣。

第 1 章“Oracle Forms 和 Reports 概述”简单地介绍 Oracle 公司提供的各种不同的开发工具的历史，还介绍了包含在 Oracle Developer 中的不同组件。

第 2 章“Developer 基础”介绍了安装。虽然应该包括每一个平台的完整指南，但是限于本书空间和时间，并没有给出详细指南。因此，我精简了安装的重要部分，这样你将知道如何避免缺陷和问题。

第 3 章“Oracle Forms Developer”，第 4 章“Oracle Reports Developer 和 Graphics”以及第 5 章“查询构造器和模式构造器”介绍了 Oracle Developer 的基本组件。

第 6 章“过程构造器和 PL/SQL”与第 7 章“在 Oracle Developer 和 Oracle 数据库中使用 Java 语言”介绍了 Oracle Developer 的编程元素。

第 8 章“在表单中使用图形元素”和第 9 章“报表的详细介绍”，以及第 10 章“Oracle 图形构造器开发”详细介绍了 Forms、Reports 和 Graphics Developer。这些内容是基于前面章节的例子之上的，但是它们详细地介绍了如何使用高级特性自定义应用程序。

第 11 章“Developer 应用程序设计”介绍了一个示例应用程序的基本设计方法和步骤。Big Jim 是一个需要简单的数据库应用程序的假想人物。你将了解这个应用程序的设计和开发。

第 12 章“使设计成为现实”继续介绍 Big Jim 的示例应用程序；在客户/服务器部署环境中创建这个应用程序。

第 13 章“在 Web 上部署 Developer 应用程序”介绍了如何使用表单服务器和报表服务器将你所设计的 Big Jim 示例应用程序在 Web 环境中部署。

第 14 章“项目管理和源代码控制”介绍了使用项目构造器进行的项目管理和源代码控制。

第 15 章“创建大型 Developer 应用程序的准则”介绍了创建应用程序的有用准则。它介绍了创建可移植应用程序的外观、性能调整、多语种应用程序以及提示的内容。

我的目的是为读者提供创建 Oracle Developer 应用程序所需的信息。在本书中，还有许多其他特性没有介绍，这是为了避免出现混淆。如果你觉得我遗漏了一些重要的内容，请立即给我发电子邮件：mbennett@utah-inter.net。所遗漏的内容可能会出现在本书的第二版中。

本书英文版书名：Programming with Oracle Developer。

本书英文版书号：ISBN 0-672-32110-6

本书原出版社邮件地址：feedback@sampublishing.com

目 录

译者序	
前言	
第 1 章 Oracle Forms 和 Reports 概述	1
1.1 Oracle 开发工具的历史	1
1.1.1 在图形化用户界面之前的日子	1
1.1.2 客户/服务器计算	3
1.1.3 基于 Web 的应用程序的诞生	4
1.1.4 基于 Web 的客户/服务器应用程序的优点	4
1.2 Oracle Internet Developer Suite 组成	5
1.2.1 Oracle Designer 和 Repository	5
1.2.2 Oracle Forms Developer	5
1.2.3 Oracle Reports Developer	5
1.2.4 Oracle JDeveloper	5
1.2.5 Oracle Discoverer	6
1.2.6 Oracle Portal	6
1.3 Oracle Forms Developer	6
1.3.1 Forms Compiler 和 Forms Runtime	7
1.3.2 Oracle Forms Server	7
1.4 Oracle Reports Developer	8
1.4.1 Oracle 报表编译器和报表运行时	8
1.4.2 Oracle Reports Server	9
1.5 表单和报表共有的工具	9
1.5.1 图形构造器和运行时	9
1.5.2 查询构造器	9
1.5.3 模式构造器	10
1.5.4 过程构造器	10
1.5.5 项目构造器	10
1.5.6 翻译构造器	11
1.6 Oracle 数据库服务器在表单和报表开发中的角色	11
1.6.1 PL/SQL	11
1.6.2 Java	11
1.6.3 对象	12
1.7 表单和报表的部署	12
1.8 小结	13
第 2 章 Developer 基础	14
2.1 安装	14
2.1.1 操作系统要求	14
2.1.2 硬件要求	15
2.1.3 数据库服务器要求	15
2.1.4 在 Microsoft Windows 平台下的安装帮助	16
2.1.5 在 UNIX 上安装	16
2.1.6 安装示例表	19
2.2 使用 Forms Developer	20
2.2.1 创建你的第一个表单	21
2.2.2 运行你的表单	23
2.2.3 创建复杂的表单	24
2.3 使用 Reports Developer	25
2.3.1 创建你的第一个报表	25
2.3.2 创建复杂的报表	28
2.4 使用 Oracle Graphics	29
2.4.1 你的第一个图形	29
2.4.2 复杂的图表和图形	31
2.5 小结	32
第 3 章 Oracle Forms Developer	33
3.1 使用表单构造器创建一个主细节表单	33
3.1.1 主细节关系	33
3.1.2 开始创建一个主细节表单	34
3.1.3 对象浏览器和属性选项板	35
3.1.4 添加细节块	37
3.1.5 使用可重入的布局向导	39
3.2 运行时环境下的主细节表单	41
3.3 使用布局编辑器	41
3.3.1 移动对象	41
3.3.2 布局编辑器工具栏	42
3.3.3 布局编辑器的可视化工具	47
3.3.4 布局编辑器的标尺和网格	49
3.4 对象浏览器的详细介绍	49

3.5 小结	52	6.3.5 包	96
第 4 章 Oracle Reports Developer 和 Graphics	53	6.3.6 数据库触发器	97
4.1 使用报表构造器创建一个主细节报表 ...	53	6.3.7 处理异常	98
4.1.1 使用查询构造器创建数据库查询 ...	53	6.3.8 Developer 中的 PL/SQL	99
4.1.2 使用可重入的布局向导	55	6.4 小结	100
4.1.3 使用报表编辑器	56	第 7 章 在 Oracle Developer 和 Oracle 数据库中使用 Java 语言	101
4.1.4 使用对象浏览器	62	7.1 在 Oracle 中使用 Java	101
4.2 在报表中集成图形	63	7.2 使用 Java 语言创建存储过程	102
4.3 小结	65	7.2.1 一个简单的存储过程	103
第 5 章 查询构造器和模式构造器	66	7.2.2 在 Java 存储过程中使用 SQL	105
5.1 查询构造器的详细介绍	66	7.2.3 删除一个用 Java 语言编写的 存储过程	106
5.1.1 运行查询构造器	66	7.3 在 Oracle Developer 中使用 Java 语言	106
5.1.2 创建查找查询	68	7.4 小结	108
5.1.3 创建表关系	69	第 8 章 在表单中使用图形元素	109
5.1.4 保存查询	70	8.1 表单运行时环境	109
5.1.5 查询窗口工具栏	70	8.1.1 可视化运行时环境	110
5.1.6 更多的例子	72	8.1.2 运行时环境的隐藏特性	112
5.1.7 结果窗口工具栏	73	8.1.3 运行时环境的小结	114
5.1.8 查询构造器小结	76	8.2 手动地创建一个表单	115
5.2 模式构造器	76	8.2.1 计算列	117
5.2.1 创建表	77	8.2.2 向导使操作更简单	118
5.2.2 创建关系	79	8.3 不同的可视化元素	118
5.2.3 创建索引	80	8.3.1 静态图像	118
5.2.4 模式构造器工具栏	80	8.3.2 按钮	119
5.2.5 创建完整性约束条件	81	8.3.3 列表项	121
5.2.6 创建视图	83	8.3.4 复选框	125
5.3 小结	83	8.3.5 单选按钮和成组单选按钮	125
第 6 章 过程构造器和 PL/SQL	84	8.3.6 图形向导	126
6.1 PL/SQL 语言在 Developer 中的角色	84	8.4 在调试器中运行表单	126
6.2 过程构造器	85	8.5 背景图	128
6.2.1 程序块编辑器	86	8.6 值列表	129
6.2.2 PL/SQL 解释器	87	8.7 对话框	130
6.2.3 对象浏览器	89	8.8 小结	132
6.2.4 语法选项板	89	第 9 章 报表的详细介绍	133
6.3 PL/SQL 语言	90	9.1 报表示例	133
6.3.1 声明并使用变量	90	9.1.1 Tabular 报表	134
6.3.2 集成 SQL	92	9.1.2 Form-Like 报表	135
6.3.3 常用的语言结构	94	9.1.3 Mailing Label 报表	136
6.3.4 函数	96		

9.1.4 Form Letter 报表	138	11.5 小结	183
9.1.5 Group Left 报表	140	第 12 章 使设计成为现实	184
9.1.6 Group Above 报表	141	12.1 表单的创建	184
9.1.7 Matrix 报表	142	12.1.1 三个维护表单	184
9.1.8 Matrix with Group 报表	144	12.1.2 Events 表单	186
9.2 报表视图	145	12.1.3 签名表单	188
9.2.1 即时预览器	145	12.1.4 结果项表单	196
9.2.2 数据模型	147	12.2 报表的创建	197
9.2.3 布局模型	148	12.2.1 确认信	197
9.3 创建并修改报表模板	150	12.2.2 结果报表	198
9.3.1 修改一个已有的模板	150	12.3 图形的创建	199
9.3.2 创建一个新模板	151	12.4 创建菜单	199
9.3.3 模板小结	153	12.5 可能的改进	201
9.4 参数表	153	12.6 小结	202
9.5 小结	155	第 13 章 在 Web 上部署 Developer	
第 10 章 Oracle 图形构造器开发	156	应用程序	203
10.1 图表类型	156	13.1 理解 Developer 应用程序的	
10.1.1 条形图	156	Web 部署	203
10.1.2 线状图	158	13.1.1 表单服务器体系结构	203
10.1.3 饼图	159	13.1.2 报表服务器体系结构	204
10.1.4 表图	161	13.2 在 Web 上部署表单	204
10.1.5 条形和线状混合图	161	13.3 在 Web 上部署报表和图形	207
10.1.6 双 Y 轴图表	163	13.4 在你的 Developer 应用程序中	
10.1.7 散点图	163	使用 Java	208
10.1.8 高低图表	164	13.4.1 使用 Java 存储过程和函数	208
10.1.9 Gantt 图表	165	13.4.2 在表单中使用 Java Bean	208
10.1.10 图表类型小结	166	13.5 小结	208
10.2 Drill-down 图表	167	第 14 章 项目管理和源代码控制	209
10.3 小结	170	14.1 项目构造器	209
第 11 章 Developer 应用程序设计	171	14.1.1 在项目构造器中创建并	
11.1 应用程序背景和需求	171	添加文件	209
11.2 应用程序界面	172	14.1.2 使用项目构造器构建文件	214
11.2.1 表单	173	14.1.3 使用分发向导	215
11.2.2 报表	175	14.1.4 使用模块类型向导自定义项	
11.2.3 图形	176	目构造器	218
11.2.4 Big Jim 的确认	176	14.2 源代码控制	220
11.3 要创建的数据库对象	176	14.2.1 源代码控制系统	221
11.3.1 表和数列	177	14.2.2 在项目构造器中使用源	
11.3.2 索引	179	代码控制	221
11.4 创建数据库模式	181	14.3 小结	223

第 15 章 创建大型 Developer 应用程序

的准则	224	15.2.6 关于性能的最后一点提示	229
15.1 调整外观	224	15.3 开发多语种应用程序	230
15.1.1 外观方面	224	15.3.1 考虑因素	231
15.1.2 实用性	226	15.3.2 多语种应用程序应该 注意的问题	232
15.2 性能调整	226	15.4 编写可移植的应用程序	232
15.2.1 所有应用程序	227	15.4.1 颜色	232
15.2.2 表单	228	15.4.2 字体	232
15.2.3 报表	229	15.4.3 特定键消息	232
15.2.4 图形	229	15.5 小结	232
15.2.5 Web 部署	229		

第 1 章 Oracle Forms 和 Reports 概述

本章的基本要点：

- Oracle Developer 代表了 20 世纪 80 年代初期以来数据库开发工具的发展。
- Oracle Developer 能够开发基于客户/服务器的应用程序。这些应用程序可以不经修改就在 Web 上进行部署。
- Oracle Developer 6i 中提供了两个主要的开发工具：Oracle Forms（表单）和 Oracle Reports（报表）。
- Oracle Developer 还包括其他工具：用于构造 SQL SELECT 语句的查询构造器（Query Builder）；用于创建数据库对象的模式构造器（Schema Builder）以及用于创建数据库存储过程的过程构造器（Procedure Builder）。
- Oracle Developer 还包括了用于维护应用程序代码的项目构造器（Project Builder）。
- 可以使用 PL/SQL 和/或 Java 语言创建应用程序。

当前，Oracle 公司所推出的开发工具套件代表了意义重大的产品发展。Oracle 数据库的第一个版本还依赖高级编程语言（例如，COBOL、FORTRAN、C）来创建界面。虽然有些开发人员还喜欢使用高级编程语言，但是大多数开发人员最终还是放弃了高级编程语言所提供的一些灵活性，而选择了可以进行快速应用程序开发的开发工具，例如 Oracle Developer。

1.1 Oracle 开发工具的历史

要了解 Oracle Developer 的最新版本，最好是先了解其开发产品的历史。虽然你可能希望跳过本节内容，但是本节提供了大量关于这些产品和工作原理以及工作模式的详细信息。

1.1.1 在图形化用户界面之前的日子

Oracle 公司所推出的第一个被业内广泛接受的表单开发工具是 Interactive Application Facility（交互式应用程序工具，IAF）。这个工具由两个基本模块组成：Interactive Application Generator（交互式应用程序生成器，IAG）和 Interactive Application Processor（交互式应用程序处理器，IAP）。IAG 可以询问开发人员一系列关于应用程序外观的问题。它可以将开发人员的回答保存在一个文本文件中。IAP 可以解释这个文本文件中的回答并向开发人员显示表单界面。这样，表单就可以用来查询、插入、更新或者删除数据库中的信息。

在 Oracle 的表单开发工具中保持了这个基本的体系结构。虽然表单布局的数据文件不再是以问题的回答保存在文本文件中，但是该工具中还是有一个应用程序生成器和一个应用程序运行时（Runtime）。

编辑文本文件

我在 Oracle 公司的第一个工作就是为电信销售部门创建基于 Oracle 的应用程序。在那时候, Oracle 公司刚刚发布了 Oracle 数据库版本 4, 因此我就非常熟悉 IAF、IAG 以及 IAP 工具。

在第一次创建表单时需要大量的工作。首先, 使用图纸来设计表单的布局并确定屏幕的几何图形。接下来, 我就将回答 IAG 所提出的问题。当完成表单时, 我可以运行该表单并查看最后的结果。对于某些问题的回答, 出现错误的情况是非常少见的。因此, 表单看上去就与当初设计的没有什么差别。这样做的优点就是 IAG 工具将所有的问题和回答都保存在一个文本文件中, 而我可以对其进行编辑并且修改文本文件所保存的问题和回答。

我认为我已经考虑到了所有的事情, 并且非常高兴地去创建和修改应用程序。可是, Oracle Developer 改进了其 IAF 工具, 从而使我的整个系统变得非常糟糕。因为在这些新增加的特性中, IAG 工具将询问更多的问题, 这意味着我正在使用的原来的开发文件不再有效了。要解决这个问题, 我不得不重新创建至少一个表单并且还要非常重视由新特性带来的新问题。于是, 我又回到了文本文件并且在其他的表单中增加新的问题和回答。

这样, 我又回到了以前的快乐中。可是, 这个快乐又是短暂的。电信销售商将使用不联网的 PC 机向一个本地数据库输入新的客户, 而且在一天工作结束时才将这些数据上载到中央客户数据库。中央客户数据库运行在一台 Digital VAX 上, 而且该机器对于处理实时数据项来说是非常慢的。基于 DOS 的 PC 机处理速度也较慢, 但是它适合处理这些数据。IAF 可以在两个系统中运行, 因此我只要维护一种数据源——至少这就是我所希望的。

在 20 世纪 80 年代中期, Oracle 的主要开发平台是 Digital VAX。最初, 所有的代码都是在这个平台中开发后, 再移植到其他平台上。当它们发布 VAX 平台的新版本 IAF 时, 我的问题就出现了, 但是如果是 DOS 平台的话, 就不会出现问题。我记得我曾经花费了许多个夜晚坐在计算机面前试图使一个最初在 VAX 平台下开发的表单能在 PC 机上运行。

如今, 我非常高兴地看到 PC 版本的表单可以与 VAX 平台的表单相兼容。当然, 这是在前两个版本没有同步不久之前的事情, 我又不得不回到了为每一个表单保持两个版本的情况。

Oracle 第一个被广泛使用的报表工具的工作原理非常类似于 IAF 工具的工作原理。这个工具就是 RPT/RPF, 它由两个模块组成: Report Generator (报表生成器, RPT) 和 Report Formatter (报表格式化器, RPF)。首先, 应该使用一个非常简单的报表语言编写一个能够从数据库中提取信息的报表。接着, 应该在 RPT 中运行这个程序, 这样可以使用格式化命令将信息从数据库中取出并保存在一个文本文件中。再者, 也可以在 RPF 中运行输出文本文件, 这样可以格式化结果。

RPT/RPF 是一个非常流行的报表工具, 这使 Oracle 公司非常不愿意放弃这个工具。在数据库版本 6 中, Oracle 公司开发了一个更加健壮的报表工具——ReportWriter。但是, 由于大量的

报表是使用 RPT 和 RPF 开发的, 所以 Oracle 在数据库版本 7 中继续支持这两个工具。即使是现在, 仍然还有一些数据库产品继续支持这个旧的报表工具。

Oracle 是第一批接受结构化查询语言 (或者称为 SQL, 其发音为 “se-quel”) 的数据库开发商。Oracle 决定通过其产品的命名来改进这个语言。IAF 工具最终就演变成 SQL * Forms。这个报表工具就成为了 SQL * Reports 和 SQL * ReportWriter。

在 20 世纪 80 年代中期, Oracle 所具有的可以在所有主要的计算机平台上运行的特性成为其销售的主要卖点之一。回想那时候, 市场上还没有出现广泛使用图形化用户接口 (或者 GUI) 的数据库工具, 因此表单和报表也是以基于文本的终端来运行的。虽然其显示效果并没有如今所使用的 GUI 的效果那样好, 但是它们却是非常实用而且为用户提供了获得信息的有效方法。

基于文本的终端所提供的有限界面还需要一些额外的产品。在 Oracle 数据库版本 5 中, 就引入了 SQL * Menu 工具。这使得开发人员可以创建菜单, 通过这些菜单, 开发人员可以使用最少的代码就运行或调用表单和报表。当发布了基于 GUI 的工具后, SQL * Menu 的功能就已经被集成到表单构造产品中了。

随着 Microsoft Windows、Apple Macintosh 界面和 X Window 系统的不断流行, Oracle 的开发工具也在不断地发展并且包括了对更多健壮的 GUI 环境的支持。Oracle 还保留了对基于文本的应用程序的支持。用户喜欢能够更新运行在基于文本终端上的已有应用程序, 以及能够尝试新的 GUI 界面的灵活性。

从基于文本的界面到基于 GUI 的界面所发生的转换是在 Oracle Forms 版本 3 和版本 4 之间实现的。为了帮助实现这个转换, Oracle 提供了能够帮助开发人员转换非 GUI 表单的转换工具。Oracle 还保留了对旧版本触发器的支持——这消除了对所有表单进行重新编码的需要。

最后, 对基于文本的应用程序开发工具的需求逐渐减少, 而 GUI 界面则成为了标准。正是 GUI 界面成为标准导致了客户/服务器计算模式时代的到来。客户/服务器计算允许数据库在一个中央服务器上运行而应用程序界面则在另一个单独的客户端运行。客户端是通过网络连接服务器的。

1.1.2 客户/服务器计算

在客户/服务器计算模式中, 也出现了新的问题。当使用原来的基于文本的终端时, 中央计算机负责维护客户与服务器的连接以及数据的完整性。如果有两个用户尝试更新相同的信息块时, 服务器将协调这两个事务并将问题降低到最小程度。使用客户/服务器网络连接同样也需要进行相似的协调, 但是对连接的协调相对要复杂一些。Oracle 提供了 SQL * Net 工具, 使客户端可以无缝地保持在以前的环境中所实现的相同的数据具有完整性。

客户/服务器计算模式的副作用是用户可能会继续使用 Oracle 数据库, 但是却又选择不同的前台应用程序工具。这就需要由其他公司来创建能够模拟 SQL * Net 工作模式的连接协议。大多数软件提供商并不希望将他们的前台应用程序工具锁定在一个数据库提供商上, 因此就开发了类似于 ODBC (开放式数据库连接) 的协议。但是, 惟一的问题是如果这个应用程序不包括在其他数据库提供商的产品之内, 那么应用程序可能不再根据 Oracle 数据库所提供的功能来调整其功能。

对 SQL * Net 工具来说,使用这个替代协议的另一个问题是服务器上到底有多少数据将传递到客户端。有些协议可能会将服务器上的一个查询所产生的所有数据发送给客户端。这样,当一个用户错误地查询整个数据库时,将导致一个巨大的网络瓶颈。在客户端计算机使用完所有的内存或整个结果集被返回之前,客户端计算机看上去可能会处于死机状态。

随着 Oracle 不断地推出所有新开发的产品,SQL * Net 工具的名称也不再具有 SQL * 前缀并且具有了版本号。这就是我们现在所知道的 Net8 工具,而且该工具仍然是无缝地集成在 Oracle Developer 和 Oracle 数据库服务器中的。

1.1.3 基于 Web 的应用程序的诞生

随着互联网的不断发展,客户/服务器计算模式的缺陷日益明显。这种计算模式最主要的问题是维护客户端软件的开销。这个开销不仅包括了运行客户端数据库应用程序的必要软件,还包括应用程序本身的代码。

当在客户/服务器计算模式中运行 Oracle Developer 时,需要安装你所使用的所有组件(例如,表单和报表)的运行时(runtime)。还需要安装 Net8 来管理到服务器的连接。不幸的是,它并不能简单地通过一个远程系统管理员来完成,而通常是在客户端计算机的物理位置中完成的。

当对实际的应用程序进行更新时,问题会变得更加复杂。随着这些更新的完成和问题的修改,将由特定的人员负责对每一个单独的客户端机器进行这些更新。

虽然这对只有五台或六台客户端机器的小型办公环境来说还不是非常困难的,但是对于具有上千台客户端机器的大型办公环境来说却是一件非常恐怖的事情。这就是为什么许多信息技术(IT)组织非常迅速地接纳了 Web 技术的原因。

在基于 Web 的应用程序中,应用程序的分布就被大大地简化了。应用程序客户端惟一所需要的就是满足特定需求的浏览器。在大多数情况下(并不是所有的情况),计算机现在都安装了功能强大的浏览器,这个浏览器是作为基本系统的一部分。应用程序的一个拷贝就可以放置在中央服务器中运行并可供每一个用户访问。当需要对应用程序进行修改时,只要在中央服务器进行一次安装就可以完成。这种体系结构消除了安装运行时软件的需要以及对应用程序进行多次更新的可能。

1.1.4 基于 Web 的客户/服务器应用程序的优点

虽然基于 Web 的应用程序可能解决客户/服务器应用程序的一些问题,但是这种应用程序还是存在一些缺点。最主要的缺点是基于 Web 的应用程序中事务的维护非常困难。Web 连接是由超文本传输协议(HTTP)来维护的并且是无状态的。使用一个无状态的连接,数据库事务不能被正确地维护。用 Java 语言编写的应用程序或者使用 Java 数据库连接(JDBC)来连接数据库可以正确地维护数据库事务,但是在 ODBC 中存在的问题同样会出现在 JDBC 中。虽然 Oracle 已经开发了 SQLJ 来解决 JDBC 的缺陷,但是它还是值得一提的。

为了解决与基于 Web 应用程序相关的所有问题,Oracle 公司推出了 Oracle Internet Developer Suite。除了是一个功能完善的基于 Web 应用程序开发环境外,这个工具还允许通过修改应用程序的部署方法开发客户/服务器应用程序和基于文本的应用程序。这样的操作不需要修改任

何代码就可以完成。

1.2 Oracle Internet Developer Suite 组成

Oracle Internet Developer Suite 组件包括如下产品：

- Oracle Designer 和 Repository
- Oracle Forms Developer
- Oracle Reports Developer
- Oracle JDeveloper
- Oracle Discoverer
- Oracle Portal

虽然本书主要介绍 Forms Developer 和 Reports Developer，但是对于开发人员来说，了解每一个产品的功能是非常重要的。它将有助于数据库应用程序的开发。

1.2.1 Oracle Designer 和 Repository

Oracle Designer 是一个可视化建模工具，它将使开发人员掌握业务需求和规则。它支持不同的建模方法，包括 ER 图表、信息工程以及对象分析和设计。这个工具是用来支持大型项目的开发并支持一个团队工作环境。

Oracle Designer 是集成在 6i 正式发行版的 Oracle Repository 工具中。Repository 工具是用来提供对 Oracle 数据库对象、表单应用程序、Java 类、XML 以及其他类型文件的配置管理。

任何参与一个大型开发项目中的开发人员都希望花费一定时间来学习如何使用 Oracle Designer 和 Repository 工具。除了可以帮助你和你的团队记录数据库表关系之外，它还可以用来创建表（完成数据完整性约束条件）以及开发一些你所需要的应用程序表单和报表。

1.2.2 Oracle Forms Developer

Oracle Forms Developer 可以用在数据库中创建高效的、可扩展的以及可移植的应用程序（或窗口）。这些应用程序允许查询、更新、插入以及删除数据库中的信息。你可以创建一个单独的应用程序并使用 Web、客户/服务器模型或者基于文本的终端来配置这个应用程序。

由于 Oracle Forms Developer 是本书的重点之一，因此在本书中将详细介绍这个工具。

1.2.3 Oracle Reports Developer

Oracle Reports Developer 可以创建报表，这些报表可以根据数据库信息创建并分布报表。根据你的需求，可以通过 Web 方式或打印形式分布这些报表。

与 Oracle Forms Developer 一样，Oracle Reports Developer 也是本书的一个重点，并且将在本书中详细介绍。

1.2.4 Oracle JDeveloper

Oracle JDeveloper 是一个用于创建和分布 Java 组件和应用程序的功能强大的 Java 开发环境。

与大多数集成的开发环境一样，这个工具包括一个综合的并且可以减少错误和问题的调试器，以及常规的语法标识和颜色标识的功能。

JDeveloper 完全支持 JDK 1.1、1.2 和 1.3，以及 J2EE 企业版编程模型。这对于希望拥有一个健壮的 Java 环境的开发人员来说是非常重要的。

Oracle JDeveloper 集成在 Oracle Designer 和 Repository 中，可以用于团队项目开发。

1.2.5 Oracle Discoverer

Oracle Discoverer 可以用来帮助非技术用户从数据库中检索信息，它可以不要求用户了解复杂的 SQL 语言。一般地说，这应该是由某些技术人员设置的预定义查询或窗口信息。非技术用户可以简单地修改查询条件并检索重要的信息。

1.2.6 Oracle Portal

Oracle Portal 也就是以前用户所知的 WebDB，它可以用来创建企业 Web Portal（互联网门户网站）。与其他 Web 开发工具不同，你没有必要在你的计算机中安装任何软件，只要将浏览器的地址栏指向运行 Web Portal 的机器，这样就可以开始发布信息。

使用 Oracle Portal 工具，可以不需要 Web 站点管理员或应用程序工程师的帮助就能完成结构化数据或非结构化数据的发布。这样，就可以使员工之间实现针对特定项目的轻松交流和协作，从而避免了给内部 IT 员工带来的大量工作。

1.3 Oracle Forms Developer

Oracle Forms Developer 是一个非常复杂和健壮的产品，该产品由几个组件组成。最常用的组件是 Forms Builder（表单构造器）。表单构造器是一个用来在数据库中创建窗口的 IDE（集成式开发环境）。这些窗口可以用来查询、更新、插入或者删除信息。

在 Oracle Forms Developer 中，有一些内置的向导，这些向导可以帮助快速而高效地开发常规表单。它们包括：

- Data Block Wizard（数据块向导）——一个数据块对应于从数据库的特定表中所选择的数据。虽然数据库块只能映射到一个表，但是一个表单可以包含多个数据块。
- Layout Wizard（布局向导）——布局向导将负责组织表单的外观。它将使用数据块向导中的信息并且可以操作表单的背景显示。
- LOV Wizard（值列表向导）——LOV（List of Values，值列表）可以为你的表单创建一个值的列表。一个 LOV 可以用于查找查询表的值，例如州或国家。
- Chart Wizard（图表向导）——图表向导将帮助创建你的数据的图形化显示效果。这个向导包括基于百分比的饼图或销售数量的条形图。

如图 1-1 所示的是使用表单构造器创建一个简单的表单。

在开发你的表单过程中，经常要使用对象浏览器窗口、属性选项板窗口和布局编辑器窗口。它们将在第 3 章中详细介绍。

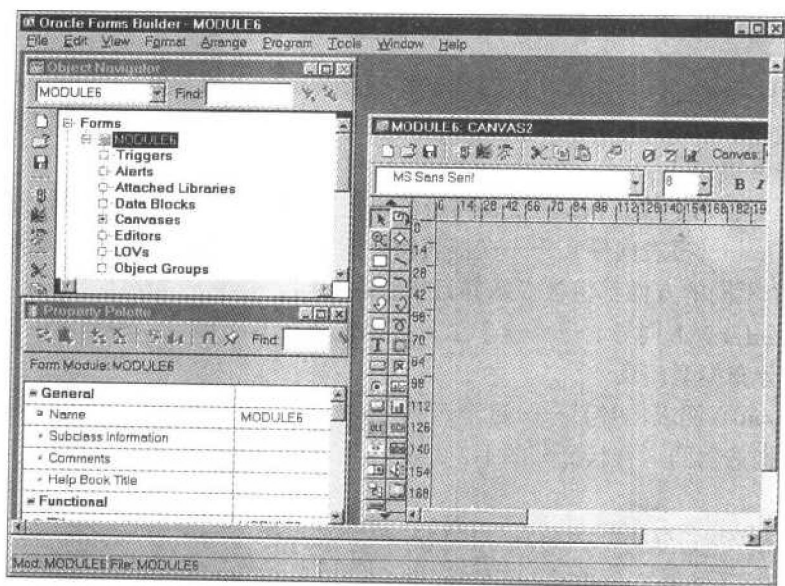


图 1-1 在表单构造器中显示了 Object Navigator (对象浏览器) 窗口、Property Palette (属性选项板) 窗口和 Layout Editor (布局编辑器) 窗口

1.3.1 Forms Compiler 和 Forms Runtime

如果你决定使用客户/服务器模型来部署你的应用程序，可以使用 Forms Compiler (表单编译器) 和 Forms Runtime (表单运行时) 工具。当在表单构造器环境中测试表单时，表单的编译和运行都是自动完成的。但是，如果你计划使用客户/服务器方法部署表单，你需要以下两个工具：

- 表单编译器——这个工具将读入一个以 .FMB 为扩展名的文件并将该文件转换成一个具有 .FMX 扩展名的可执行数据文件。在 SQL * Forms 的早期版本中，类似于 .FMB 文件功能的文件是以文本文件的形式保存的，因此你可以手工地打开并修改文件信息。但是现在，这两个文件都以特定的数据格式进行保存。Forms Developer 的以前版本将表单编译器当成是“表单生成器”。
- 表单运行时——表单运行时将处理 .FMX 文件并在客户端机器上运行这个文件。你需要在使用你的应用程序来访问数据库的每一个计算机上安装表单运行时。

1.3.2 Oracle Forms Server

如果你决定使用基于 Web 的模型来运行你的应用程序，你就需要使用 Oracle Forms Server (表单服务器)。它是一个专门为事务管理、记录缓存、记录锁定、异常处理以及工作量的平衡而设计的应用程序服务器。

在使用客户/服务器方法来部署应用程序时，你可以在基于 Web 的环境中系统地测试表单。

1.4 Oracle Reports Developer

Oracle Reports Developer 是用来在数据库中创建查询或报表的工具。与 Oracle Forms Developer 类似，Oracle Reports Developer 的主要模块是报表构造器。它可以用来定义在报表中出现的信息以及这些信息的格式。

为了使报表更容易生成，Oracle 报表构造器也包括了几个向导。这些向导分别是：

- Report Wizard (报表向导) —— 在报表构造器中，这是创建报表的最简单的方法。报表的每一个不同的方面都将被表示并且很容易访问。
- Web Wizard (Web 向导) —— 这个工具可以创建在 Web 浏览器中显示的报表，它与客户/服务器分布模型相反。
- Chart Wizard (图表向导) —— 这个工具可以很容易地创建表示报表中数据的图表。

Oracle 报表构造器的运行状况如图 1-2 所示。

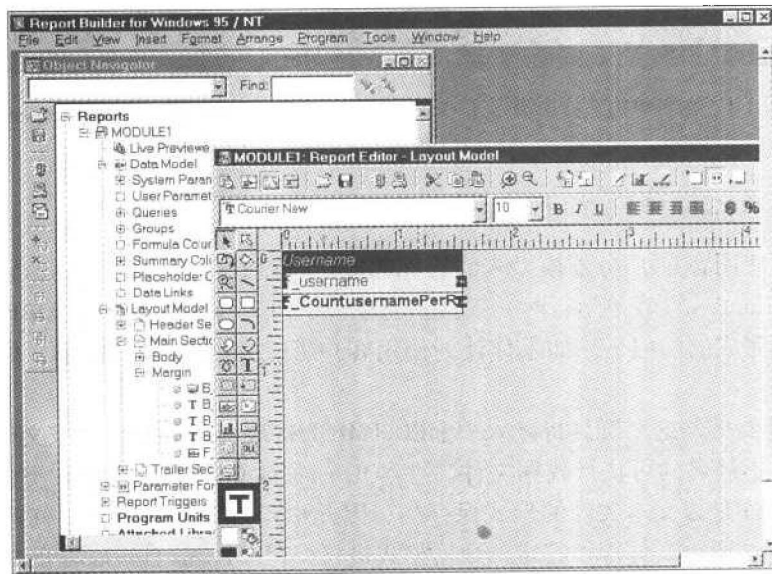


图 1-2 显示了对象浏览器窗口和布局编辑器窗口的报表构造器

1.4.1 Oracle 报表编译器和报表运行时

你可以发现，Oracle Reports Developer 的体系结构非常类似于 Oracle Forms Developer 的体系结构。它们都可以使用表单编译器和表单运行时工具来编译客户/服务器模型的报表。当使用报表构造器时，该工具还提供了一些菜单选项，这些选项能让你直接运行报表，而不需要手工地编译这些报表。当你准备将报表在客户/服务器环境中使用时，你需要确认表单编译后的版本和 Oracle 报表运行时同样被放置在客户端机器中。