



ORACLE®

Oracle 技术系列 丛书

Oracle

Forms 与 Reports

高级开发

Oracle
Developer Advanced
Forms & Reports

Oracle Press™ 授权出版

(美) Peter Koletzke 著
Dr. Paul Dorsey

李逸波 黄海帆 王华驹 马赛红 等译



机械工业出版社
China Machine Press

OSBORNE Education

Oracle技术系列丛书

Oracle Forms与Reports

高级开发

(美) Peter Koletzke 著
Dr. Paul Dorsey

李逸波 黄海帆 王华驹 马赛红 等译



机械工业出版社
China Machine Press

本书全面地介绍Oracle Developer Forms组件和Reports组件的特性、标准以及使用技术。全书内容包括：Oracle Developer 6的新特性，Oracle Developer的基本标准，Forms的开发方法和有关技术，Reports的开发方法和有关技术，表单和报表开发的技巧和技术。本书附录还给出包括许多示例、问题、帮助、向导的具体内容。

本书内容覆盖了在使用Forms组件和Reports组件时各个方面的技术，讲解内容极为丰富和详尽，为Oracle Developer的应用开发者提供了许多方法、经验和途径，是一本具有很高实用价值的开发指导书。

Peter Koletzke and Dr. Paul Dorsey: Oracle Developer Advanced Forms & Reports (ISBN 0-07-212048-7).

Copyright © 2000 by the McGraw-Hill Companies, Inc.

Authorized translation from the English language edition published by McGraw-Hill, Inc.

All rights reserved. For sale in the People's Republic of China.

本书中文简体字版由机械工业出版社和美国麦格劳-希尔国际公司合作出版。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：01-2000-1315

图书在版编目（CIP）数据

Oracle Forms与Reports高级开发/（美）考洛克（Koletzke, P.），（美）道斯（Dorsey, D.P.）著；李逸波等译。—北京：机械工业出版社，2001.9

（Oracle技术系列丛书）

书名原文：Oracle Developer Advanced Forms & Reports

ISBN 7-111-09289-9

I.O… II.①考… ②道… ③李… III.①关系数据库—数据库管理系统，Oracle Developer Forms ②关系数据库—数据库管理系统，Oracle Developer Reports IV.TP311.138

中国版本图书馆CIP数据核字（2001）第055713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：张文敏

北京市密云县印刷厂印刷·新华书店北京发行所发行

2001年9月第1版第1次印刷

787mm×1092mm 1/16·32.75印张

印数：0 001—4 000册

定价：56.00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。Oracle数据库在许多国家和地区得到广泛的应用。Oracle在世界进入网络时代时，推出了Oracle Developer的各个组件，包括Graphics Builder（图形生成器）、Procedure Builder（过程生成器）、Project Builder（工程生成器）、Query Builder（查询生成器）、Schema Builder（模式生成器）、Translation Builder（转换生成器）、Forms Builder（表单生成器）和Reports Builder（报表生成器）。其中的Forms组件和Reports组件是使用最多的产品。本书着重介绍了这两种产品的特性、标准和使用技术。

Forms是所有开发者都要使用的组件，而Reports组件常常是有些开发者容易误解和不理解的组件。书中不但讨论在Reports组件和Forms组件中相同的标准和技术，而且分别介绍这两个组件各自的特性。阅读本书的读者最好已经使用过Oracle Developer Forms，对Forms组件的基本知识有所了解而对Oracle Developer的Reports组件可以不太了解。因为本书对Reports组件的介绍要比Forms组件稍微详细一些。

这本书的作者具有十分丰富的使用Oracle Developer进行开发应用程序的经验。书中介绍的许多涉及操作和技巧的内容非常具体、周到和细致，对于需要了解Oracle Developer Forms组件和Reports组件的使用方法的读者来说，这是一本十分难得的参考书。希望这本书的翻译和出版，能够给Oracle Developer 软件的开发者 and 应用者以较大的启发和帮助。

本书由李逸波、黄海帆、王华驹、马赛红、张景生、齐玉东、田妍、于鸽、陈培军、郭冬梅等翻译。本书内容丰富、涉及面广、篇幅较多，译者的时间和水平有限，难免有疏漏和错误，敬请读者批评指正。

李逸波
2001年2月

序 言

因特网改变了一切事物，它改变了我们的工作方式、生活方式、甚至我们的思维方式。各个公司都在重新考虑和重新调整自己的核心业务，并根据这种新的思维方式研究自己的商业模式和竞争对手。总之，他们正在重新定义该如何运作。通过因特网，一个公司可以开展自己的电子商务。在电子商业中，准确适时的信息是至关重要的。通过使用因特网标准来开发和使用数据库应用程序，会给电子商务带来巨大的动力，这也要求公司重新界定与供应商、合伙人及客户的关系。利用因特网，过去只能在企业内访问的数据库应用程序如今在企业外也可以使用。如今这些实际企业、综合供应商、合伙人和客户对新一代数据库应用程序已经有所了解。这样要求访问存储在数据库中重要商业信息的用户数就显著地增加了。因此，访问这些数据库的应用程序必须性能高、容易使用而且随时能使用。因特网带来更多的机会和竞争，因此也大大缩短了产品的生命周期。要保持竞争力，就必须以最快的速度开发和利用这些应用程序。

Oracle是世界上最大的电子商务软件供应商。《财富》100强的公司中有65%使用了Oracle的电子商务软件。这是因为Oracle提供了完整的因特网平台，包括Oracle8i数据库、应用程序服务器和工具。

Oracle工具包括一组综合的Business Intelligence Tools（商业智能工具），可以方便地回答有关商业的任何问题，Oracle还提供用于开发事务处理应用程序的有效的开发工具。Oracle Business Intelligence Tools包括获奖的适用于特别查询和分析的Oracle Discoverer，用于高保真数据库出版的Oracle Reports，以及用于开发分析应用程序的Express工具。Oracle Development Tools（开发工具）包括富有成效并基于模型的开发工具Designer和Developer，还有基于部件灵活性的Java开发工具JDeveloper。Oracle工具使应用程序的开发自动化，而且减少了开发应用程序所需的劳动和时间。更主要的是，通过减少易于出错的人工编码过程，Oracle工具大大改善了应用程序的质量。

Oracle为在因特网上使用这些应用程序提供了最优化的通用的应用程序服务器。JDeveloper使得能够利用Oracle Application Server和Oracle8i JServer灵活地开发多用户的Java商业逻辑。Developer Server用于协调事务处理和企业报表应用程序。Discoverer Server用于优化高性能的查询和分析。因此，数据库应用程序可以在集中管理的服务器上高效地运行。只要使用标准的Web浏览器，用户就能访问这些应用程序。

用户经常要花大量的时间去寻找他们需要的应用程序和信息。Oracle开发了入口框架（portal framework）Oracle WebDB，为便于用户访问必需的数据库应用程序和信息提供了统一的桌面。现在用户可以容易地从单个公共入口访问他们所有的数据库应用程序以及任何Web站点。用户还可以根据自己的任务和个人爱好使入口的内容和布局个性化。通过入口可以立即访问到当时最新的信息。

Oracle Developer是市场上建立数据库应用程序的第一流的开发工具。它提供了建立世界级数据库应用程序的有效工具和条件，使企业内外的商务功能和处理自动化。现在Oracle

Developer是世界范围内开发定制的企业级应用程序的公司所选择的工具。此外，应用程序的卖主，如Oracle Applications，使用Oracle Developer为Enterprise Resource Planning（ERP）和Customer Relationship Management（CRM）提供全球的打包应用程序。事实上，由于Developer Server的强大功能，这些应用程序甚至可以作为预约服务从因特网上的头号应用程序服务提供商Oracle Business Online得到。简言之，Oracle Developer是在市场上能提供整套电子商业应用程序的惟一工具。

本书介绍如何使用Oracle Developer Release 6.0迅速地产生可装入Oracle入口框架的数据库应用程序。作者Peter和Paul为Oracle用户提供了有价值的资源：如何充分使用Oracle Developer面向对象特性，并研究Oracle Developer的关键技术，以确保你的项目能够成功。在开发应用程序时，这些技术将帮助你节省时间和减少工作量。有Oracle Developer和这本书的指导，用户可以迅速地建立使用方便的、功能强大的、作为电子商务基础的数据库应用程序。

Oracle公司工具产品部副总经理

Sohaib Abbasi

目 录

译者序

序言

第一部分 概 述

第1章 绪论	1
1.1 为何有些人不用Oracle Developer	1
1.2 本书所涉及的范围	2
1.2.1 对Web的覆盖面	3
1.2.2 本书的覆盖面	3
1.2.3 为什么讨论标准	4
1.2.4 版本号策略	4
1.3 本书的读者对象	5
1.4 本书是如何组织的	5
1.5 怎样获得更多信息	6
1.5.1 获得更多信息的策略	6
1.5.2 Oracle 用户信息	7
1.6 关于Developer Forms数据体系结构的回顾	8
1.7 示例文件	9
第2章 Oracle Developer 版本6的新特性	10
2.1 如果跳过版本2怎么办	10
2.1.1 Forms从版本1到版本2的变化	10
2.1.2 Reports从版本1 到版本2 的改变	12
2.2 Developer版本6对Oracle8的支持	13
2.2.1 对象数据类型	13
2.2.2 PL/SQL 8	15
2.3 Forms Builder 的新特性	16
2.3.1 使用方便的特性	16
2.3.2 Web特性	28
2.4 Report Builder的新特性	31
2.4.1 灵活使用	32

2.4.2 Web特性

35

第二部分 Developer的基本标准

第3章 SQL和PL/SQL标准	37
3.1 为什么制定代码标准	37
3.1.1 易读性	38
3.1.2 文档	38
3.2 强制执行标准	38
3.2.1 不使用代码标准的10个重要理由	38
3.2.2 答案是什么	39
3.2.3 帮助工具	39
3.3 通用代码标准	40
3.3.1 最长30个字符	40
3.3.2 缩进使用3个空格字符	40
3.3.3 关键字大写	41
3.3.4 小写用户定义的元素	42
3.3.5 一行不超过80个字符	42
3.3.6 等宽字体	44
3.3.7 每行做一件事	44
3.3.8 文档注释	45
3.3.9 将数据库对象的名字记到账户上	46
3.4 SQL代码标准	47
3.4.1 大写和小写	47
3.4.2 空格和缩进	47
3.4.3 代码注释	48
3.4.4 命名约定	49
3.4.5 代码的组织	50
3.4.6 样本代码	50
3.5 PL/SQL代码标准	51
3.5.1 大小写	51
3.5.2 空格和缩进	51
3.5.3 注释代码	52

3.5.4 命名规则	54	第6章 表单命名约定	98
3.5.5 代码的用法	57	6.1 为什么要为表单对象命名	99
3.5.6 样本代码	63	6.2 命名约定考虑	99
第4章 建立块的基础	66	6.2.1 块命名考虑	99
4.1 表单查询文本	66	6.2.2 其他考虑	100
4.2 不根据任何东西建立块	67	6.3 特殊对象的命名约定	101
4.3 根据数据表建立块	68	6.3.1 应用程序系统	101
4.4 根据视图建立块	69	6.3.2 表单和对象库	102
4.4.1 可更改视图	70	6.3.3 程序库	109
4.4.2 带有INSTEAD OF触发器的视图	71	6.3.4 菜单	110
4.4.3 不可更改视图	73	第7章 表单中的代码放在哪里	112
4.4.4 实现基于视图的块	74	7.1 代码放置的标准	112
4.5 创建基于过程的块	75	7.2 程序包与独立的程序	112
4.5.1 PL/SQL表过程	75	7.2.1 程序包的好处	113
4.5.2 REF CURSOR过程	76	7.2.2 程序包的缺点	114
4.5.3 实现基于过程的块	76	7.3 数据库服务器与表单	115
4.6 根据FROM子句查询建立块	82	7.3.1 如何分离表单代码	116
4.7 根据事务触发器创建块	84	7.3.2 其他数据库代码源	117
第5章 Forms 的GUI 标准	86	7.4 处理Forms功能	118
5.1 表单的开发过程	86	7.4.1 如何处理Forms功能	119
5.1.1 第一阶段——非功能故事板	87	7.4.2 表单级触发器	120
5.1.2 第二阶段——半功能原型	88	7.5 触发器与程序单元	120
5.1.3 第三阶段——产品表单	89	7.6 表单与程序库	121
5.2 表单开发指南	89	7.7 程序库	122
5.2.1 表单的大小	89	7.7.1 通用程序库	123
5.2.2 屏幕的分辨率和监视器的大小	89	7.7.2 应用程序专用的程序库	123
5.2.3 屏幕实际状况	90	7.7.3 表单专用的程序库	123
5.2.4 项: 字体、颜色和属性	93	7.8 子类化与从对象库中拷贝	124
5.2.5 文本、标签、线条和矩形	94	7.9 哪一级的触发器	125
5.3 表单布局编辑器的技巧和技术	95	7.9.1 执行层次	127
5.4 各种表单标准技术	95	7.9.2 另外的例子	127
5.4.1 延迟要求的强制性属性	95	7.9.3 选项	127
5.4.2 建立目录结构	96	7.10 开发过程	128
5.4.3 在Form Builder中排列窗口	96	7.11 通用准则	129
5.4.4 排列块和项的次序	97	第8章 Reports开发标准	130
5.4.5 表单对象的注释属性	97	8.1 使报表通用化	130
5.4.6 开发环境中的快捷键	97	8.1.1 通用化报表的好处	130

8.1.2 通用化报表的缺点	131
8.1.3 什么时候使报表通用化	131
8.2 设置报表标准	131
8.2.1 打印硬件	131
8.2.2 把报表的代码放在哪里	133
第9章 报表命名约定	137
9.1 命名报表	137
9.2 为查询命名	137
9.3 组的命名	138
9.4 命名框架	139
9.4.1 封闭框架	139
9.4.2 重复框架	141
9.5 命名显示域	141
9.6 命名程序单元	141
9.7 命名模板项	141

第三部分 Forms 模板

第10章 Forms模板基础	143
10.1 Forms和VRAD	143
10.2 什么是模板表单	144
10.2.1 使用模板的优点	144
10.2.2 使用模板的缺点	145
10.2.3 在模板中放什么	146
10.3 在模板中使用的Form Builder特性	154
10.3.1 模板支持	154
10.3.2 子类	155
10.3.3 对象组	157
10.3.4 对象库	159
10.4 模板系统结构	163
10.5 怎样使用模板	166
第11章 构造Forms模板系统	167
11.1 过程	167
11.1.1 生命周期	167
11.1.2 从哪里开始	168
11.1.3 RAD入门	169
11.1.4 步骤总结	169
11.2 代码和对象的通用化	170

11.2.1 代码的通用化	170
11.2.2 编写通用对象	174
11.3 对象库元素	177
11.3.1 标准对象	178
11.3.2 可选对象	181
11.3.3 容器	182
11.3.4 项	182
11.3.5 其他的SmartClasses	183
11.4 其他模板任务	185
11.4.1 参考表单	185
11.4.2 模板表单	186
11.4.3 为模板文件编写文档	186
11.4.4 菜单模板	187
11.4.5 修改模板组件	188
11.5 特殊类型的模板	189
11.5.1 选项卡界面	190
11.5.2 导航模板	190
11.5.3 管理模板	190
11.5.4 定位器模板	191
11.5.5 层次树模板	191
11.5.6 特定工程模板	191

第四部分 高级Forms主题

第12章 以模板为基础建立表单	193
12.1 安装模板	193
12.1.1 放置模板文件	193
12.1.2 设置开发环境	194
12.1.3 使用模板版本	194
12.1.4 设置数据访问	195
12.2 创建新表单	195
12.3 添加数据对象	196
12.3.1 使用块向导	196
12.3.2 手工建立块	202
12.4 应用SmartClasses	202
12.4.1 怎样进行SmartClass	203
12.4.2 选择哪个SmartClass	204
12.4.3 对其他对象进行SmartClass	204

12.4.4 移去SmartClass	205	14.1.2 需要多少内容	239
12.5 修改模板代码	205	14.2 内部Forms帮助	240
12.5.1 已子类化的触发器	205	14.3 表格式帮助	241
12.5.2 为什么拷贝程序单元	207	14.3.1 帮助系统的组件	241
12.5.3 STARTUP程序包	207	14.3.2 查询机制	245
12.5.4 UTIL程序包	212	14.3.3 增强系统	248
12.5.5 添加可重用对象和代码	214	14.3.4 使用帮助窗口	249
12.6 增加其他对象	215	14.3.5 其他表格式的帮助系统	249
12.6.1 增加可重用组件	215	14.4 WinHelp帮助系统	250
12.6.2 增加其他对象	215	14.5 这是什么	251
12.7 增加其他代码	216	14.5.1 应用WinHelp	251
第13章 在Web上发布Forms	218	14.5.2 怎样结合WinHelp和Forms	252
13.1 Web表单环境	219	14.6 其他的帮助选择对象	255
13.1.1 Java虚拟机	220	14.6.1 HTML形式的帮助	255
13.1.2 中间层	220	14.6.2 用Designer生成帮助系统	256
13.1.3 Web表单的运行时期	221	第15章 定位器表单	257
13.1.4 优点	224	15.1 设计定位器项	258
13.1.5 开发环境	225	15.1.1 弹出列表和LOV	259
13.2 Oracle信息导航文档	225	15.1.2 复选框	259
13.3 一般建议	226	15.2 实现定位器查询准则	260
13.3.1 为Web进行设计	226	15.2.1 实现定位器	260
13.3.2 测试Web	227	15.2.2 实现简单的数字项和 简单的字符项	260
13.3.3 分割代码	227	15.2.3 查询其他表	262
13.3.4 了解“瘦客户机”的含义	227	15.3 通用项查询	263
13.3.5 建立用户期望	228	15.4 定位器结果块	265
13.3.6 重新编译表单	228	第16章 多值选择对象	266
13.3.7 简化表单	228	16.1 复选框	267
13.3.8 使用认证的客户JVM	229	16.2 单选组	269
13.3.9 设置基础结构	229	16.3 列表项	272
13.4 技术和考虑	229	16.3.1 实现弹出列表	272
13.4.1 操作系统	230	16.3.2 弹出列表的限制	273
13.4.2 降低网络流量	231	16.4 值列表	273
13.4.3 设计差别	234	第17章 集成Forms和Reports	275
13.4.4 限制和变通办法	236	17.1 从Forms调用Reports	275
第14章 Forms帮助系统	238	17.1.1 向报表传递信息	276
14.1 需要多少帮助	238	17.1.2 处理参数列表	276
14.1.1 哪一种层次的帮助	238		

17.2 绑定变量和词汇参数	277	19.1.2 多重查询	300
17.2.1 绑定变量	277	19.1.3 使用视图	303
17.2.2 词汇参数	278	19.1.4 给列取别名	304
第五部分 Reports的应用		19.1.5 在查询中选择PL/SQL函数	304
第18章 Reports综述	281	19.1.6 外部连接	305
18.1 Reports Builder的体系结构	282	19.1.7 报表的临时数据表	305
18.1.1 Data model数据模型	282	19.2 非查询字段	306
18.1.2 Layout Model布局模型	284	19.2.1 公式列	306
18.1.3 Parameter Form参数表单	284	19.2.2 占位符列	306
18.1.4 Live Previewer实况预览	284	19.2.3 报表层对象	307
18.2 高级特性和尚未使用的特性	284	19.3 组	308
18.2.1 词汇参数	285	19.3.1 Break列	309
18.2.2 内置的PL/SQL函数	285	19.3.2 指定组	310
18.3 报表向导	285	19.3.3 组过滤器	311
18.4 数据模型	286	第20章 报表布局模型	312
18.4.1 建立查询	287	20.1 使用布局模型	313
18.4.2 组	288	20.1.1 使用框架	314
18.4.3 汇总列和公式列	288	20.1.2 打印方向	317
18.5 布局模型	292	20.1.3 垂直和水平弹性	317
18.6 对象导航器	295	20.1.4 折行模式	320
18.6.1 Live Previewer	295	20.1.5 限制模式	321
18.6.2 数据模型	295	20.1.6 Additional default layout工具	321
18.6.3 布局模型	296	20.1.7 增加字段	322
18.6.4 参数表单	296	20.1.8 移动对象	322
18.6.5 报表触发器	297	20.2 锚点：让它呆在属于它自己的地方	323
18.6.6 程序单元	297	20.3 格式触发器	324
18.6.7 附加库	297	第21章 使用和建立Reports模板	325
18.6.8 模板	297	21.1 综述	325
18.6.9 外部SQL库	298	21.2 模板结构	326
18.6.10 PL/SQL库	298	21.2.1 模板编辑器——数据模型	326
18.6.11 Debug Action and Stack	298	21.2.2 模板编辑器——布局模型	327
18.6.12 内置程序包	298	21.2.3 报表触发器	330
18.6.13 数据库对象	298	21.2.4 程序单元	330
第19章 报表数据模型	299	21.2.5 附属库	330
19.1 精心设计查询——让它简单	299	21.3 创建样本模板	330
19.1.1 查询生成器	299	21.3.1 数据模型	331
		21.3.2 布局模型	331

21.3.3 程序单元和附属库	334
21.3.4 改变模板的缺省设置	334
21.4 设置报表模板标准	336
第22章 建立报表	337
22.1 准备建立报表	337
22.2 建立报表步骤	339

第六部分 Developer的应用

第23章 测试问题与解答	347
23.1 Oracle Developer 的专业知识	347
23.2 关于测试问题	347
23.3 Oracle Forms测试问题	347
23.3.1 Forms初级问题	348
23.3.2 Forms中级问题	353
23.3.3 Forms高级问题	356
23.4 Oracle Reports测试问题	360
23.4.1 Reports 初级问题	360
23.4.2 Reports 中级问题	364
23.4.3 Reports 高级问题	367
23.5 计算答案得分	372
第24章 Reports技巧与技术	373
24.1 页面中断	373
24.2 表单类型的报表中页面 布局的控制	376
24.3 复杂的报表中心	378
24.4 使用圆点引导的策略	378
24.5 管理带有表单的报表	379
24.6 更换页标题	380
24.7 记录子集的计数	380
24.8 以函数为基础的查询	381
24.8.1 程序包	381
24.8.2 其他步骤	382
24.9 报表中的性能优化	383
24.9.1 使用小数量的查询	383
24.9.2 避免使用要求数据库 访问的公式列	383
24.9.3 不要从数据库中提取不显示	

的从属记录	383
24.9.4 优化 SQL	383
24.9.5 注意绑定变量	384
24.9.6 避免不必要使用图形	384
24.9.7 打印机性能	384
第25章 Forms的项的开发技巧 与技术	385
25.1 层次树控制	385
25.1.1 准备	386
25.1.2 组成部分	387
25.1.3 步骤	387
25.1.4 在查询时	388
25.1.5 附加步骤	388
25.1.6 内置程序和触发器	390
25.2 动态弹出列表	391
25.3 系统分配的ID值	393
25.3.1 表单和序列	393
25.3.2 利用数据库触发器	393
25.3.3 利用PRE-INSERT触发器	394
25.4 加载图像项	395
25.5 文本项的访问键	397
25.5.1 定义访问键	397
25.5.2 文本项访问键的制法	398
25.5.3 其他思路	400
25.6 作为按钮图标的字符	400
第26章 Forms的块的开发技巧与技术	402
26.1 命名块	402
26.2 动态块的筛选	402
26.2.1 PRE-QUERY 触发器中的 查询条件	403
26.2.2 WHERE子句块属性	403
26.2.3 如何以过程为基础建立块	404
26.3 动态块排序	404
26.3.1 实现动态排序	404
26.3.2 列表项Order By	405
26.3.3 列的Order By按钮	406
第27章 Forms模块的开发技巧与技术	409

27.1 处理Forms错误和信息	409	27.9.2 对齐对象	430
27.1.1 触发器	410	27.9.3 浏览窗口和画布	430
27.1.2 过程	410	第28章 Forms菜单技巧与技术	432
27.1.3 使用MESSAGE	412	28.1 建立菜单的技术	432
27.1.4 错误文本和信息文本的标准	413	28.1.1 创建菜单	432
27.1.5 变更	413	28.1.2 菜单属性	434
27.2 警报系统	414	28.1.3 魔术项	435
27.2.1 通用警报	415	28.1.4 复选项和单选项	436
27.2.2 警报函数	415	28.1.5 菜单图标	437
27.2.3 调用示例	416	28.2 菜单工具条	438
27.2.4 标准的按钮标签	417	28.2.1 按钮排序	438
27.2.5 标准的警报信息文本	417	28.2.2 隐藏和显示按钮	439
27.2.6 OK警报过程	417	28.3 实现菜单安全	440
27.3 通过对象循环	418	28.3.1 数据库设置	440
27.3.1 通过项循环	418	28.3.2 菜单设置	440
27.3.2 通过块循环	419	28.4 动态工具条和菜单	441
27.3.3 通过其他对象循环	419	28.4.1 准备	442
27.4 调整表单	419	28.4.2 组成要素	442
27.4.1 从表单外部调整	420	28.4.3 KEY-ENTQRY和KEY-EXEQRY 触发器	443
27.4.2 从表单内调整	421	28.4.4 CHANGE_MODE过程	443
27.5 查找表单中的代码和对象	424	28.4.5 步骤	444
27.5.1 查找代码	424	28.4.6 变化	444
27.5.2 查找对象	425	第29章 Forms系统的技巧与技术	445
27.5.3 生成Forms对象列表报告	425	29.1 共享变量	445
27.5.4 利用Forms Open API	426	29.1.1 数据库程序包说明变量	445
27.6 改变光标的形状	426	29.1.2 表单程序包说明变量	446
27.7 调用其他表单	428	29.1.3 参数	446
27.7.1 CALL_FORM	428	29.1.4 块中的项	448
27.7.2 NEW_FORM	428	29.1.5 全局变量	448
27.7.3 OPEN_FORM	428	29.1.6 使用“不变的”变量	449
27.8 调用其他程序	429	29.2 编译表单文件	449
27.8.1 不在Web上	429	29.2.1 编译表单	449
27.8.2 HOST	429	29.2.2 编译程序库	450
27.8.3 HOST START	429	29.2.3 命令行参数	450
27.8.4 WinExec	429	29.2.4 编译的次序	451
27.9 其他技巧和技术	430	29.2.5 UNIX批编译	452
27.9.1 用户注释系统	430		

29.3 设置表单登记值	452
29.3.1 设置路径	453
29.3.2 从表单内设置注册	453
29.4 外部函数接口	454
29.4.1 FFIGEN.FMB	454
29.4.2 FFIGEN创建什么	455
29.4.3 FFI程序包说明	455
29.4.4 FFI程序包体	455
29.4.5 错误检查	458
29.5 Oracle安装程序安装哪些文件	458

附 录

附录A PL/SQL库软件包和样本模板	461
附录B 面试问题	482
附录C WinHelp 的概念	491
附录D 使用Report Wizard的步骤	494
附录E 进行报表审计	502
附录F 报表说明书	504
附录G 在Web上发布表单 的Oracle白皮书	509

第一部分 概 述

第1章 绪 论

Oracle Developer 是一个功能强大的产品。熟练的开发小组在建立访问Oracle数据库的应用程序时,使用Oracle Developer 的Forms (表单)和Reports (报表)能够很快地建立起一个对用户友好的复杂系统。由于Oracle Developer支持大部分用户界面所需的要求,即便使用户要求非常复杂,所建立的系统也能满足。另外,Oracle Developer还具备足够的灵活性,允许用户对应用系统进行调整,以优化性能。当前的Oracle Developer Server版本功能相当成熟,为开发应用程序提供了稳固的环境,并且具有在Web上开发应用程序的能力。

尽管上述理由促使用户采用Oracle Developer工具来建立应用系统,但是,还须注意下列问题:

- 为什么仍然有许多开发者抱怨Developer的使用如此困难?
- 为什么一些购买Developer的开发者在参加培训班六个月之后却以项目失败而告终?
- 为什么在用户群中对Oracle Developer的标准没有普遍的认识?
- 为什么开发者不了解Oracle Developer提供的模板系统、对象和代码共享特性的强大能力?
- 为什么某些系统专业开发人员放弃Oracle Developer而重新用Visual Basic和C++去编写应用程序?

1.1 为何有些人不用Oracle Developer

这个问题在相当程度上没有意义。一方面,有许多开发者使用Oracle Developer进行产品开发,他们对软件感到满意,并称赞Oracle Developer。另一方面,有许多Forms和Reports的用户却在使用工具时遭到失败。造成这种矛盾现象的因素有很多:

1. 产品的发展

无论是Oracle Developer 的Forms组件还是Reports 组件,都和产品一样是在不断发展的,而每次发展都为满足开发者的进一步要求。如果你使用过SQL *Forms 2.3版或3.0版,那么,你很容易就能理解当前产品的基本概念。然而,如果你使用Oracle Developer是从6.0版开始,则须克服很多困难。在Developer工具程序的整个生命周期中,大部分的开发资源集中于增加和增强产品的功能,另外也注意使产品对开发者友好。要提供对各种层次的用户都容易操作且功能强大的产品在某种意义上是很困难的。

2. 需要经验

对早期的Oracle Developer开发者来说,使用当前的版本不用花多大功夫。然而,要理解Developer工具程序如何运行和交互,要掌握Developer并达到一定水平则是具有挑战性的。当本书一名作者第一次用Developer工作时,花费了四个小时的时间来弄明白如何完成某些新任务。

即使在此之后，他仍不敢确定所提出的方法是否是最好的。随着建立Forms和Reports的实际经验的增多，花费的时间会减少到一小时或更少。如果你想不经过训练或在没有良师益友的帮助下学习使用Developer，则将经历一个非常困难的时期才能学会如何建立各种有用的Forms和Reports。经验就是一切。如果没有经验，则要从经历各种事情中学到经验。

3. 需要一个模板系统和一些标准

Oracle Developer的Forms组件是一个优秀产品。它为建立应用程序提供了一个适当的开发环境。然而，有效地建立和使用Forms要求具有经过周密考虑过的模板，GUI集和代码标准。这一点对于任何的产品开发来说都是正确的。使用没有某些基本的重用组件集的框外产品来建立产品系统将是一个艰苦的过程。但如果在Forms中建立一个固定的模板和一套标准，其效率的改善可能将是非常巨大的。对于Developer的Reports组件，也同样是正确的。

4. 大量的技巧

应该采用“正确的”方法去使用Forms和Reports完成指定的任务。按照作者的估计，一个优秀的Forms开发者最少需要了解上千个技巧，对于Reports的开发者来说，最少也需要上百种技巧。这是非常大的知识面。仅仅只有少数有经验的开发者才了解大多数特性、属性、技巧和技术。而如果不具备这些知识和不了解如何使用这些知识，新建立的应用程序的性能就不好，对用户也不友好，在建立时还会浪费许多时间。

5. 文档不适用和培训不充分

在很多情况下，每次新版本发行时，相应的文档和培训都不是很充分。产品发展得很快，以至于文档和培训常常很难与版本保持一致。随Developer发送的主要文档支持在线使用（随购人员附带的硬拷贝），其内容覆盖的方面涉及语言和属性以及Web开发，给出了一个概述，使得可以开始着手建立应用程序。但却不是最好的关于实践和标准的书。很多培训组织，包括Oracle的培训组织，为开发者讲授如何使用工具完成各种任务，但却不提供足够的从模板系统建立表单产品和报表产品的知识，也不提供有关重用代码和对象库的知识。

解决方法

怎样才能改善上述情形呢？其中的一个办法就是撰写这本书。本书要达到的目标是：

- 文档化和解释有效使用Oracle Developer的Forms和Reports所需要的许多必需的技术。
- 讨论Forms和Reports中的模板结构和建立重用组件的概念。
- 介绍在使用Oracle Developer创建应用程序时可以使用的构造标准的框架。

本书不可能包括所有在用户中存在的Oracle Developer的知识缺口。然而，本书瞄准了我们当前考虑问题的实质。这本书解释了基本原理、核心技术、标准方法以及使用Oracle Developer建立应用程序的最好的体系结构。

1.2 本书所涉及的范围

要达到本书的目的并在所定的篇幅中合理安排内容，必须要确定一个范围，该范围不可能满足所有人的需要。本书的目的不仅仅是为了使读者读到有关技巧，因为我们觉得仅仅讨论这种话题达不到具有高级知识的目的。例如，在本书中，将找不到如何使用Forms APL、Oracle

Terminal和 PL/SQL调试器的解释,因为这些话题在初学者的书中和Oracle Developer的帮助系统中已经进行了充分的讲述。

本书的标题和副标题正好反映了它的内容。本书中讨论的技术是在Forms和Reports组件产品的高级范畴(意味着有较大比例的Oracle Developer用户不知道和不理解这些内容)和标准(在许多Oracle Developer工程中遗漏了的内容)中的技术。

这本关于所有的Oracle Developer组件的高级教科书,在讨论Graphics Builder(图形生成器)、Procedure Builder(过程生成器)、Project Builder(工程生成器)、Query Builder(查询生成器)、Schema Builder(模式生成器)和Translation Builder(转换生成器)的技术时,决定只精选这样两个部分的内容:Forms组件——开发者要使用其大部分特性,Reports组件——开发者非常容易误解和不理解的组件。因为我们认为读者已经使用过Oracle Developer,所以我们假定他们已具备Forms组件的基本知识。如果在基于对象的方法中使用Oracle Developer的Reports组件(也像Oracle Reports一样出售),会得到巨大的生产效益。但是,许多开发者甚至不了解Reports的基本知识,因此,在书中解释Reports组件比Forms组件要稍微详细一些。同时讨论在Reports组件和Forms组件中相同的标准和技术。

1.2.1 对Web的覆盖面

本书不是一本关于在Web上开发应用程序的书。Oracle Developer Server允许在Web上开发应用程序,但本书不解释它在这个领域的全部内容。关于Oracle Developer Server的安装、配置和优化需要一整本书去讲述。这种书的读者往往是数据库管理员和网络管理员,并且一般还负责Web服务器的安装工作。本书不准备涉及这些内容,因为本书的读者是开发者和管理者,他们需要与他们有关的技术知识和有关标准方面的知识。

另一方面,本书在第13章解释了在Web上开发Forms的某些基本知识和细节。同时,本书还介绍了用Oracle Developer生成应用程序的有关技巧和最好的实践内容。所有这些,对于用Oracle Developer来完成的工作都适用,而不管采用的是哪种开发平台。

1.2.2 本书的覆盖面

作为Oracle工具的用户,我们开发那些与工具同时发展的方法。因为有越来越多使用产品的工作,因而我们能够检验出那些使用这些工具去进行开发工作的最好的方法。我们频繁地提到Oracle Developer的面向对象的思想,就像我们生产应用程序的感觉:在成长和(满怀希望地)成熟一样。本书并不介绍与Reports和Forms有关的每一件事情,而是介绍以下内容:

- 说明开发Reports和Forms的基本原理和使用Oracle Developer创建应用程序的重要技术。
- 给出发现更多的关于特定论题的信息的方向。
- 讨论迄今尚未发表的Oracle Developer部分标准。

Oracle在产品中销售标准解决方案已有很多年,例如Custom Development Environment (CDE)、ERP (Enterprise Resource Planning) 和CRM (Custom Relationship Management) 程序组(也称为Oracle 应用程序)。第三方的计算机销售商许多年来同样也销售这些解决方案,但是Oracle Developer的标准仍然未被大众所接受。