



Oracle 技术系列丛书

ORACLE®

AUTHORIZED ORACLE PRESS™—EXCLUSIVELY FROM OSBORNE

# Oracle E-Business Suite Financials 手册

Oracle E-Business Suite Financials Handbook

David James  
(美) Simon Russell 著  
Graham Seibert

王军德 张琳 等译



OFFICIAL • AUTHORIZED

Oracle Press

ONLY FROM OSBORNE



机械工业出版社  
China Machine Press



Education

Oracle技术系列丛书

# Oracle E-Business Suite Financials手册

David James

(美) Simon Russell 著

Graham Seibert

王军德 张琳 等译



A1008598



机械工业出版社  
China Machine Press

Oracle E-Business Suite (电子商务套件)是由150多个模块集成在一起的程序集。本书讲述E-Business Suite, 将重点主要放在三个基本的内容上, 即模块的功能、模块实现的管理和组织以及与其他模块的集成。相关章节的内容也是按照这三部分组织的。虽然本书的重点是Oracle Financials 模块, 但多数的技术和模块管理的内容也适用于其他Oracle Applications 模块。内容包括: Oracle Applications程序集的范围、功能以及基本的概念, 使用Oracle Financials中的应用程序模块, 如Oracle General Ledger、Oracle Payables和Oracle Receivables等, 着重介绍每个模块的功能和实现, 讨论如何管理和定制 Oracle Financials 应用环境, 使其更适合管理人员的操作。第四部分简要概述其他的 Oracle Applications 模块。

本书适合 Oracle E-Business Suite以及其他Oracle财务软件的使用人员、相关平台的财务软件开发人员阅读。

David James, Simon Russell and Graham Seibert: Oracle E-Business Suite Financials Handbook (ISBN: 0-07-213230-2).

Copyright © 2002 by The McGraw-Hill Companies, Inc.

Original language published by the McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字版由美国麦格劳-希尔教育出版公司授权机械工业出版社出版, 未经出版者预先书面许可, 不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有McGraw-Hill公司防伪标签, 无标签者不得销售。

版权所有, 侵权必究。

**本书版权登记号: 图字: 01-2002-0803**

### **图书在版编目 ( CIP ) 数据**

Oracle E-Business Suite Financials手册/ (美) 詹姆斯 (James, D.) 等著; 王军德等译.  
-北京: 机械工业出版社, 2002. 9

(Oracle技术系列丛书)

书名原文: Oracle E-Business Suite Financials Handbook

ISBN 7-111-10539-7

I. O… II. ①詹… ②王… III. 电子商务 - 软件包 - 技术手册 IV. F713.36

中国版本图书馆CIP数据核字 (2002) 第046247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: 李 密 张鸿斌

北京第二外国语学院印刷厂印刷 · 新华书店北京发行所发行

2002年9月第1版第1次印刷

787mm × 1092mm 1/16 · 37印张

印数: 0 001 - 4 000册

定价: 65.00元

凡购本书, 如有倒页、脱页、缺页, 由本社发行部调换

# 前 言

Oracle E-Business Suite (Oracle电子商务套件) 是一个由150多个集成软件模块组成的程序集, 这些软件模块可用于财务管理、供应链管理、制造、工程系统、人力资源以及客户关系管理等。

Oracle Applications引入了许多代表当前技术水平的功能, 并且进行了精心的优化, 从而能快速地实现预定的目标。Oracle公司已经在数据库市场上建立了良好的声誉。虽然近10年来它们大力着手于应用系统的开发, 但其基本宗旨仍然没有变化。世界一流水平的商务管理应该建立在卓越的Oracle数据库之上, 并完全采用Oracle自身的开发工具。应用软件一直是Oracle战略方向的一个重要的基础部分。

Oracle Applications拥有广阔的应用市场, 其范围从基于PC的小型机构到拥有大型机的世界500强中的大公司。应用系统的用户与数据处理支持人员有很多共同之处, 他们都欣赏Oracle强大功能和灵活性所带来的显著的效益, 但却总是对要获得这些效益所需的规划、准备等大量艰苦的工作估计不足。

本书主要讨论用于财务管理的应用模块。全书围绕三个主题进行讨论, 即: 这些模块的商业方面的功能、基本的技术以及项目实施的管理和组织。因此, 本书按这三条主线划分成不同的部分。在功能的介绍中, 把主要精力放在Oracle Financials模块上。虽然我们着重讨论的是Oracle Financials, 但是有关章节中的技术与项目管理的讨论对Oracle Manufacturing、CRM和Oracle Human Resources也是适用的。

## 关于本书

选择Oracle E-Business Suite处理财务事务的用户正在不断增加。许多用户正从原来的管理系统转向UNIX或者基于Windows 2000的开放系统, 并且作为公司的标准会计系统实施Oracle Financials。本书主要针对与实施有关的人员, 针对需要对Oracle Financials有一个大致了解以预先知道成组软件的复杂性与缺陷的人员。财务管理人员和系统人员认识到, 他们能够吸取已经实施了Oracle Applications的很多组织机构的智慧。本书的目的是与大家共享作者多年来实施Oracle Applications项目所积累的经验和教训。

## 本版本中的新增内容

本书的第1版《Oracle Financials Handbook》涵盖了Smart Client版本10.7, 也涵盖了版本11。本书的第2版涵盖了更新的E-Business Suite中的Oracle 11i版本。除了包含原来所有章节的内容外, 本书另外增加了5个完整的新章节。这些新内容包括: Oracle Projects (第14章)、Oracle 工作流和实现向导 (第15章)、11i版本中的新应用 (第16章)、Oracle Applications技术体系结构 (第18章) 和Oracle财务分析器 (第24章) 等。有关11i中的一些重要特性变化, 如订单管理、自主

服务Web应用以及Automatic Account Generator等也进行了更进一步的说明。最后，有关每个应用中功能的增强，都在第二部分中有详细的讲解。

### 本书适合的读者

数据处理人员和用户必须同时安装Applications。本书努力使每一类人员对其他类人员的工作有相当的了解。各类人员之间思想的融合，可以帮助开发人员在定制和修改数据库之前对底层商业功能有一个透彻的理解。Applications强调公司的整体观念。因此，我们尽量突出每个Application的综合方面。

本书很有新意，就其中所介绍的一些内容来说，几乎每个读者都只能算是初学者。虽然只有Oracle自己出版的参考资料能够对每个功能提供专家级的知识，但是本书将为专家们及其工作伙伴提供协同进行系统集成所需的通用参考构架，同时还提供了一种Oracle自身不可能提供的有关转换和集成的外部的观点。

本书讲述了如何成功地组织实施一个项目，介绍Oracle Financials的基本概念以及基本的技术和会计理论。本书读者应是具有会计管理知识的商业人员以及具有一定信息技术经验的技术人员。本书假定商业人员未受过专门的信息技术培训，而信息技术人员没有会计知识。本书的魅力在于它以一种商业人员和技术人员都能读懂的方式讲述了商业决策和信息技术。我们是通过向财会人员详述Oracle Financials技术方面的专门知识，为实施人员和开发者提供商务功能方面的指导来实现这一目的的。

本书并非Oracle的推销手册，也不能取代Oracle的参考手册，而是一种理解Applications的实际途径的实用指南。多读几遍本书将一定会大有收益，它能使你避免犯代价很大的错误。

### 本书内容的组织

- 第一部分内容包括Oracle Applications程序集的范围、功能以及基本概念，它是本书后面内容的基础。如果读者不熟悉Oracle Applications的话，应该首先阅读第一部分。
- 第二部分介绍使用Applications中涉及的功能和实施问题。第二部分的各章可作为工作中的参考，不需要按出现的顺序阅读。
- 第三部分讨论适用于所有Applications的技术和管理问题。其中介绍了一些公用的基础部分，它们将自己的功能和谐地提供给Applications；还介绍了Applications的操作环境以及Oracle编程工具，这些编程工具有助于定制Applications以满足特定的需求。读者将学习Oracle Applications实施项目的组织和管理，以及一旦着手工作就必须考虑的实际操作问题。这部分的知识对于负责某个Oracle Applications实施的管理人员来说是必须了解的。
- 第四部分概述了E-Business Suite产品。这部分的内容对我们的主题来说都是一些外围的话题，包括：Financial Analyzer、CRM、HR和Manufacturing。它们可以单独实施，但更多的情况是与核心的Financial Applications一起实施。

### 没有讲述的一些主题

某些重要的主题并没有在本书中讲述。本书的篇幅问题是其中的一个原因，另一个原因是：

此为试读,需要完整PDF请访问: [www.ertongbook.com](http://www.ertongbook.com)

本书是站在Greenfield站点的初级实施人员的立场上写的。本书没包括Oracle为满足一些特定行业需求而设计的产品（如下所示）：

- Oracle Clinical
- Oracle Public Sector Financials
- Oracle Student Systems
- Oracle Energy
- Oracle Consumer Packaged Goods
- Oracle Grants Management

我希望本书涉及到的一些内容，如系统设置、项目组织和管理以及软件开发等，能够对读者使用Oracle的一些统管生产和销售全过程的产品有所帮助。

## Oracle公司介绍

Oracle的奠基人Larry Ellison是一个有远见的人物。Oracle的创建构想类似于Microsoft。在政府司法部门（Justice Department）要求发展市场的压力下，随着IBM一统天下的崩溃，计算机业界出现了巨大的机会。IBM的一些公开的替代品，如DEC、HP以及非专利UNIX操作系统等，为企业提供了丰富的、便宜的处理能力。Oracle的奠基人和首席执行官——Larry Ellison的构想是创建一个能够利用这种能力的数据库，而且该数据库应该有专利。

Oracle公司得到了快速的发展，在20世纪80年代的大多数年份中销售额成倍增长。它的每一个后继版本都提供了更多的功能和更大的可靠性。Oracle所发布的“第4代”开发工具使编程人员将Oracle数据库用于数据获取、报表和特定的查询更加容易。

“第4代”程序设计语言通过有效提高程序设计者的产出率使其价值大大增加。然而，编程人员产出率提高的原因完全不是出于编程。在20世纪80年代，正当Oracle声称自己是主导性的开放式系统数据库（Open-systems Database）的同时，McCormick和Dodge、MSA以及American Software正在应用软件领域中争夺支配地位。它们主要是在IBM专有的操作系统、数据库以及COBOL程序设计语言的范围内工作。

20世纪80年代末，Oracle的数据库及其工具在支持企业级的系统中已具备了与IBM一争高下的能力。Ellison看到了Oracle利用其强有力的数据库、工具和开发专长开发世界一流的应用所能提供的巨大价值，提出建立一个财务软件程序集。Oracle的快速成长在公司内部引起了对集成软件控件的较大需求，Oracle本身就可作为这种控件的试验场所。

Oracle Applications遵循了与其他竞争产品不同的发展路线。Oracle的竞争者采用客户机/服务器（client/server）计算，这个企业计算的模型是20世纪90年代初与Windows桌面环境一起出现的。PeopleSoft采用该技术开发了他们自己的人力资源、财务和制造业软件程序集。SAP重构了其主机驻留的（mainframe-hosted）R2软件以便以客户机/服务器方式运行。然而，Oracle看到了客户机/服务器方式的局限性，将眼光投向了Internet（因特网）。他们的想法是很正确的；Internet确实是提交企业解决方案的一种更好的体系结构。而采用客户机/服务器已被证明并不能够完全适应进一步的发展。

客户机/服务器的图形用户界面（GUI）表现出了很大的吸引力，它克服了客户机/服务器体

系结构在代价和性能方面的缺陷。在基于GUI的竞争者面前Oracle的字符方式应用不占优势，尽管Oracle具有较高的性能和较完善的功能。而Ellison的Internet设想由于技术的原因被推迟了。虽然他正确地预见到了网络的快速增长，但似乎还是低估了网络浏览器变得智能化所需要的时间。Oracle的Internet Computing（网络计算）模型需要Java程序在桌面一级提供智能。没有Java，Web和数据库服务器的大量交换是不可能的。然而，Java比Oracle预期的来得还要迟。直到20世纪90年代的末期，Java在商业计算中的应用才变得足够成熟，这样便使得Oracle的Internet Computing成为可供多数用户选择的最好体系结构。

如果在产品的演化中，Oracle的目的是绕过客户机/服务器阶段的话，那么是时间捉弄了他们。1996年，他们公布了自己的客户机/服务器版Smart Client，其版本为10.6。10.6版提供了GUI功能以及几个仅在GUI中可用的新产品。版本10.7巩固了这个GUI的产品系列，并能适应2000年问题。由于Oracle的客户机/服务器产品在与其竞争者更为成熟的产品的比较下不占优势，所以Oracle软件产品的增长减慢了。

无中介（不要中间人）是Internet的一个最显著的好处。2000年中，.com公司的失败只是其失败的一面而已——许多公司为客户提供需要中间人方式的服务，而这对于传统的公司在实施电子商务以后来说是不需要的。Oracle E-Business Suite的构想就是为公司提供一种通过Internet扩展与客户、供应商和雇员关系的机会。

让自己的员工制作、处理客户订单要花很多钱，况且客户还宁可自己做订单。所以，最好让客户自己做订单，并授权给客户使他能看到自己的订单，而供应商也能明白下一步的需求。Oracle的Self-Service Web Applications正不断地为公司外部的人员提供查询公司信息和提交像订单这样的数据的能力。Oracle的Web产品、RDBMS和Applications已实现了很高的安全性，允许所有人员访问而不会对机密数据造成危害。

在Smart Client应用系统趋于成熟的同时，Oracle的Web工具也在发展。Oracle内部包含了一整套完整的技术，并使用了一个全新的产品——Oracle Applications Server（9iAS）代替了原有的Web服务器产品。该产品是同类产品性能最好的Apache Web服务器。

Oracle升级了开发人员工具，可以自动生成利用安全HTTP通信协议访问Web的代码。Applications程序集发布Oracle Self-Service Applications作为前端，把各模块固定在一起，将主干产品Purchasing、Order Management、Projects和Human Resources扩展到了能访问Internet的任何人员。iProcedure和CRM的最新版本提供了用户界面的方式，这也是所有的Applications将来会采用的方式（代替Java和Java ServerPages中的表单）。对设计人员来说，这也是采用新技术满足目前的一些功能所面临的主要修改任务和挑战。

Oracle一直是CASE（Computer Aided Systems Engineering，计算机辅助系统工程）技术的先驱。利用Oracle产品系列，能够在设计工具的高层CASE（分析和设计）和低层CASE（自动程序生成）间得到最广泛的集成。虽然CASE是一个非常有力的工具，但成功地利用它需要有较高的水平和较多的约定。它适合于软件机构使用。由设计器生成100%的代码还只是未来的事情，其中的部分原因在于以前Applications版本中的代码中的可重用性。

Internet Computing模块集中了所有内容。它采用了Oracle的Web技术，利用的是Developer 6i，该模块可以完成Oracle Smart Client的大部分功能。Oracle曾经使用CASE技术生成Applications，

并提供CASE元数据给Applications的用户。Internet Computing采用了Oracle8中发布的插件技术 (cartridge technology), 为提高效率和完整性将应用处理置于数据库内部。E-Business Suite现在包含大约150个模块, 在支持Internet的ERP软件包中占据最重要的位置, 并且Oracle拥有完整的技术。

## Oracle RDBMS

巨大的软件帝国通常都是采用“拿来主义”。Microsoft从一个购买来的称为QDOS (Quick and Dirty Operating System) 的产品发展成为桌面系统的垄断者。Lotus主要借用了Electronic Arts的思想, 而Apple则来自Xerox。Oracle也不例外。IBM认为他们的Santa Theresa实验室学术机构提出的SQL语言有一定的价值, 但会威胁到他们自己主导的IMS数据库。但Larry Ellison却从中看到了机会。

关系模型是一种比层次模型和网状模型能更好地查询数据信息的方法。起初它的效率很低, 并且不易编程。Oracle通过添加工具和应用不断加强其基础产品的实用性。通过在产品中加入大量的操作系统功能将用户与操作系统隔离开来, 并扩充了RDBMS的概念, 使其包含诸如分布式处理和对象管理这样一些概念。Oracle还利用其开放式的体系结构培植了大量的第三方商业伙伴。

位于Oracle Applications下的RDBMS管理数据、进行备份和恢复、监控自身的性能并提供充分的编程环境。它使用户不用在主要产品上选择下层的操作系统, 因为Oracle中已包含了大多数识别器。其他还有可靠性和可扩展性等操作系统要求解决的问题。

Applications 11i所需的RDBMS版本8是面向对象的。它在其关系结构中引入了面向对象的数据库管理系统 (OODBMS) 概念。编程人员现在可以定义一些数据类型来描述特定的对象, 如某个地址或某个人。按照这种方式, 这个地址可以嵌入到数据库自身之中, 而不仅仅只是一个编程结构。

版本8i以及更高的9i更关注性能和容错能力。这是针对市场上对更大或更小数据库需求的响应。Oracle现在已经发展为一个全球化的数据库实例。数据库和E-Business Suite支持Unicode字符集, 这样, 一些多字节字符集可以同英文和西欧语言共存于同一个数据库中, 如Cyrillic (西里尔) 脚本和日语。将数据集中在一个数据库中, 比在很多小数据库中存储数据可以节省很多的开销。在进行全局事务处理、报表和合并中具有很好的优点。市场对数据库的可靠性要求很高, 其关键在于全局化过程。

## Oracle Application Server

Application Server位于整个体系结构的中间层。它运行表单, 并处理Internet中的连接和同其他服务器 (大部分情况下为数据库服务器) 之间的通信。该部件的性能和可靠性是Internet计算模型成功的关键 (尽管通常有多个Application Server, 且在某个服务器失效后可以重新进行平衡)。

Application Server已经更新到了9i版本, 甚至已经经过验证可以同E-Business Suite一起工作。它是一个有关通信服务、编程支持和表示工具 (使得可以支持采用不同技术编写的应用和portal)

的巨大集合。它的数据缓冲机制可以大大增强Applications的性能。E-Business Suite的更新版本一定会更好地利用这些特性。

## Oracle编程工具

Oracle RDBMS是存储和检索事务数据的主要设施，但它的服务对象并不是最终的用户。程序必须从用户那里接收数据和查询并将结果返回给他们。编程人员一开始利用Oracle工作的方式与20世纪80年代初期的大多数数据库一样，要通过编译用COBOL和C这样的语言编写的程序。

编程人员需要用关系数据库的工业标准语言SQL和PL / SQL来访问数据库。工业委员会制定的语言规范是不断演变的，其中一般都要纳入一些主要供应商已实现的功能。Oracle的实现方式通常都会被全盘接受。

SQL是Oracle和其他大多数RDBMS系统的基础语言。SQL的创新在于它天生就是非过程化的（non-procedural）。它的设计目的是处理一组记录而不是个别的记录。术语“非过程化”表示编程人员不必告诉计算机如何工作，即不必告诉计算机如何执行查询。编程人员只要用SQL语言说明所需要的结果就行。决定如何生成结果的智能部分已设计在RDBMS内。

PL/SQL是SQL的过程化的或“可编程的”扩展。尽管SQL功能很强，但有时还是需要编程人员告诉计算机怎样每次一个记录地执行某个操作，就像主机内部实际执行的操作一样。下面介绍的许多工具一般都每次处理一个记录。PL/SQL是过程化程序设计的现代语言，是一种有很大改进的COBOL，是大多数Oracle产品共同采用的语言。

Oracle早就意识到，编程人员通过SQL和PL / SQL使用内建的访问数据库的产品应该能提高产出率。他们利用内建的、编程人员和最终用户的数据库接口创建了“第4代”程序设计语言。程序设计工具的名称随功能的发展一直在变动，但其意义用途不变。Oracle Applications中的工具及其用途如下：

- SQL\*Plus是SQL与PL/SQL语言的简单、直接的编程工具。它为编程人员和最终的用户作了扩展。Oracle Applications的安装和批处理用它来为Oracle Applications创建数据库对象。Oracle Applications可以接受用SQL\*Plus编写的用户代码，这是一个开发快速简单报表和批处理的好工具。
- Oracle\*Forms以前称为SQL\*Forms，是Oracle Developer的组成部分。它是编程人员的专用工具。Oracle Applications的核心接口都是用Forms编写的（Self-Service模块不是）。E-Business Suite是使用Forms版本6i编写的，它支持优化用于Internet的HTTP和HTTP-S通信。
- Oracle\*Reports取代了以前提供的SQL\*Reports和SQL\*Reportwriter。它也是Oracle Developer的组成部分。在设计上，Oracle\*Reports既支持字符方式报表，也支持位图（GUI）报表。Oracle Application报表内置于Oracle\*Reports版本6i中。Oracle以字符方式发布多数报表，使得对于所有的用户都是相同的，即使是采用字符方式的用户也可以很容易地将其转换为位图操作。Applications的用户经常要用Oracle\*Reports来定制程序。
- Oracle公司内部使用Designer CASE工具来开发 Applications。它为Oracle\*Reports和Oracle\*Forms生成代码。Designer的用途是改进软件的可靠性和提高生产率，特别是用于

程序维护。在对Oracle Applications进行大的扩充时，它能提供有效的帮助。Oracle提供Applications的CASE元数据，这使得在版本11i中修改代码更为容易，而且也更为可靠。

- Oracle还提供一系列支持RDBMS的实用程序。Applications的程序设计员充分利用了Oracle\*Loader将“无层次（flat）”操作系统文件装载入Oracle的表中以导入到Applications中。其他实用工具，如导出和导入，主要是数据库管理员使用。

Oracle环境包含大量的软件模块，需要具有一定的编程能力才能设置和运行，但它们本身不是编程语言。SQL\*Net是分布式系统中连接服务器到客户机、服务器到服务器的软件层。而备份、恢复和重组软件是相当复杂的。Oracle Enterprise Manager（OEM）是一系列很复杂的模块，用于在实时基础上监控数据库的活动。有一个特定的E-Business Suite工具集支持OEM，从而使这些产品之间的关系更加紧密。

## 数据仓库的发展方向

SQL语言和Oracle RDBMS最初是打算用于决策支持环境的，即从事务数据导出有意义的管理信息。RDBMS在不同方面都取得了进展。客户需求和体系结构本身都会从事务处理的优化中得到好处。通过定义主键和索引，E-Business Suite使用的数据库会对特定的访问进行优化。为支持对数据的组织不做假定的查询，人们精心地构造了不同的技术，如“倒排表”（inverted list）、“多维”（multi-dimensional）和“三维”（cubic）表等。

Oracle以几种不同的方式在其数据仓库产品中采纳了这些概念。Oracle Express是一个多维的数据仓库。Oracle 9i这个最新的RDBMS产品引入了大量的功能，它们增强了其作为关系联机分析处理（ROLAP）引擎的用途。Oracle的数据仓库方向超越了上述两个产品。Express用于通过Oracle Financial Analyzer和Oracle Sales Analyzer的多维建模。Enterprise Data Warehouse（EDW）是用于E-Business Suite的Key Data Warehouse产品，它用于运行同Business Intelligence产品（比如Purchasing Intelligence和Financials Intelligence）集成在一起的数据仓库产品。分散的Collection软件包用于从Application获取数据，并可以通过Oracle Warehouse Builder进行扩展。它可以把来自多个供应商的多个ERP部件中的数据汇集到一起。随着企业使用ERP的成熟，首先它们会通过过程的自动化来降低开销。随后，它们会转向使用数据库中的数据以获取更进一步的信息，从而降低开销和增加收入。

Oracle Express和Oracle Discoverer是设计提取和表示数据仓库信息的最终用户工具。它们具有与其预期用途一致的表格和图形数据表示设施，支持在“假定方案”（what if）中执行决策和可视表示。Balanced Scorecard和Activity-based Management产品为数据库增加了更大的价值。

## Oracle Applications的未来方向

事务处理一直是Oracle Applications的强项。Oracle的最新产品从许多应用中提取事务数据，包括从原有的和第三方的系统中提取事务数据，转换为支持关键事务决策的信息。Applications程序集的当前和未来的一些精彩之处包括：

- **联机服务（Online Services）** Internet工作在公用或专用网络上，这就使公司没有必要有一个本地数据中心。Oracle Online Services程序使客户不需要自己的计算机。Oracle保存

一个Applications的实例并进行合理的配置，让客户感觉就好像运行在公司的站点上一样，以便通过Internet使用。客户也不需要购买硬件、雇佣数据库系统管理员、处理备份有关的问题、灾难恢复、安装补丁和升级等。Oracle公司也节省了开支，它们可以以比同等的本地服务方式更低的成本提供服务。

- **小业务套件 (Small Business Suite)** 结合通过Internet访问远程宿主应用以及重写业务功能的优点，Oracle Small Business Suite主要是满足无法承担运行全部功能的应用软件的公司需要。它的原型是Netledger。Oracle Small Business Suite包含处理Finance、Payroll以及CRM的功能。
- **真正全局购买 (True Global Purchasing)** Supply Chain在逐步发生着变化。iProcurement使得所有的雇员可以通过直接访问提出请求——降低了每次请求的开销。Oracle Exchange连接到某些注册商那里，比如TPN，并将购买需求广播给市场中的合适的供应商。访问多个供应商可以避免出现无货的情况，也可以进行价格竞争。这些在今天已经实现了。Oracle认识到，发展应该是从全局的组织到共享服务中心的方向发展。组织模型需要得到增强以支持集中式的购买，就好像分散式地收款和支付一样。
- **桌面系统集成 (Desktop Integration, DI)** DI用电子表格和其他桌面软件链接Oracle Applications。General Ledger是一个初步的例子：用户无缝地在电子表格中输入预算，并在电子表预测中输出余额。利用工业标准接口，可将DI概念推广到客户方的软件包。在较少结构化的方法中，Oracle Applications可以管理来自语音、字处理、项目管理软件包、图形和表示软件包 (presentation package) 的对象。
- **更自主服务的应用** 通过与Applications接口的前端固定，Oracle的Workflow产品可以提供Oracle的所谓“未经培训用户的普通访问”。Workflow产品向用户提供了它们取得成功所需的指导。它利用安全和加密技术保护了 Applications。当用户进入核心应用时，它通过利用标准的Open Interfaces (开放式接口) 重新编辑事务确保了数据的完整性。
- **大量的可定制功能** 现在的用户在某种程度上都受到软件包设计的制约。应用软件包的长处在于它们包含了几乎是所有用户都需要的功能；其弱点在于它们有时需要所有的用户都来处理只有很少的用户才会涉及的问题。有时特定的结构要求用户用好几个步骤来完成一个简单的任务，因为软件也必须适合那些任务不那么简单的用户。利用CASE技术和面向对象的程序设计技巧对Applications进行实现，在未来Oracle可能会让客户在设计层次上进行自定义，而不是通过参数设置进行自定义。

开放系统和Internet贯穿了Oracle的产品系列，使它们统一起来。Oracle产品一直都能够运行在大多数硬件和操作系统平台上运行。Oracle正在建立在数据库、工具以及应用一级的混合环境中运行的能力。

Oracle的Internet版本使得客户在他们的桌面系统上所需的任何功能都能够从远程获得。它已经使这种版本可用于Applications程序集。许多功能，如E-mail，天生就是面向网络的。有一些公司正在研究字处理和电子表的Java实现，这些公司包括Corel、Yahoo、IBM等。当这些应用实现后，他们将会采纳其中的主要原理，在桌面系统上得到本地存储和很大的可配置的操作系统。桌面系统的用户将可以访问像Internet自身那么大的程序库，但不用承担维护它的职责。

Oracle的数据库和网络产品具有提供集中管理数据和程序的功能，以支持这个版本的开放性、强功能、高速度和可靠性。

### 选用Oracle Applications的理由

Oracle的主要竞争者SAP、PeopleSoft和JDEdwards等公司在他们的产品中已经采用了第三方的开发、网络和数据仓库工具。他们中的每个公司都将第三方的这些工具组合进一个开发平台，客户可以用这个平台来进行系统的扩充。与其他供应商的应用软件的比较，Oracle Applications下的Oracle RDBMS和程序设计层是强有力的基本因素。企业级的软件是一个不寻常的购置品。这种产品较为昂贵、不可捉摸、且不可能完全理解。购买这样的产品意味着购买了一个关于体系结构和供应商的故事。下面是有关Oracle的故事。

- Oracle Applications可能是最为有效的，因为它们与Oracle数据库结合在一起。别的供应商为了完全支持其应用放弃了Oracle以及其他一些供应商的DBMS中的效率特点。在不同的数据库供应商之间进行变换对大多数客户来说没有意义。大多数公司都看准一家DBMS供应商。当这些公司进行变动时，往往都会转向Oracle，因此增加了Oracle在RDBMS领域内的优势。
- 编写Oracle Applications的编程工具也因为专用于Oracle RDBMS的特点而受益。很多人都认为它们效率很高。而且，它们还广泛地用于Oracle Applications之外。几乎用不着像使用诸如SAP或PeopleSoft的专用语言那样需要困难地或高价地寻找编程人员来定制Oracle的应用。
- Oracle Applications的开放结构使它们易于定制和与其他供应商的软件包进行集成。几乎所有应用逻辑都可以得到。编程人员可以分析它们，包括分析许多最好不要更改的模块，以便对系统得到一个完全的理解。表结构是开放的、公开的、甚至通过链接到非Oracle数据库也可以访问。Oracle的体系结构使得较其他供应商的软件包更易于弥补用户需求和所提交的功能间的差距。
- Oracle的开发工具与Designer 2000 CASE工具紧密结合。这种结合使开发Oracle Applications的定制扩充很容易。它表示Oracle可能会在某一时刻将Oracle Applications作为元数据发布，这将使它们变得容易定制。
- Oracle的AIM产品（它为设置Oracle Applications获取参数）现在可以通过将参数插入数据库自动进行设置。用户可以指望Oracle或第三方将会编写输入AIM或从AIM输出的翻译程序，以方便在主要的应用供应商如SAP、PeopleSoft、Baan之间进行转换。
- Internet是Oracle所有产品的策略中心。Oracle正在开发很强的工具（如Oracle Web Server）支持自己。这些工具与Developer、Oracle 2000、SQL\*Net以及其他Oracle产品程序集具有紧密的结合。在将应用程序集移至Web上这一方面，Oracle比其他集成性不高的供应商具有更大的优越性。
- 经济规模构成了软件工业中的一个固定的驱动因素。一个公司为产品开发所花费的钱的数目相对于其用户数目成正比地增加。Oracle已经在RDBMS市场取得了主导性的地位，而且似乎在Web和应用市场中将自己推向主导地位方面也处于一个绝佳的位置。在写本书的时

候，Oracle Applications正从业内领先者SAP那里得到市场共享。

- Oracle的方向表示Oracle Applications将能够与其他供应商的应用一起使用。Oracle的主要竞争者已经使用了Oracle RDBMS。Oracle是通过它的EDMS和OADW产品进行数据集成的。Oracle是惟一一个有能力对其技术进行控制，以保护客户在应用软件（不管是什么样的供应商）上的投资的供应商。

上述理由都令人信服地说明，为了满足事务需求，选择Oracle是一个可靠的长期决定。它的体系结构为长期增长提供了一个可靠的基础，它的开放性表示它能与其他系统协同工作。或者说，最后这种情况将使它成为一个可移植、可再利用的系统。

## 将本书用于成功的财务实施

虽然财务实施中的步骤重复了传统开发的那些步骤，但是其过程是完全不同的。支配一个客户实施的开发阶段在Applications中减少成为几个转换和过渡程序。每样事情都由外到内：分析、编写过程、设置、培训和测试。所有这些都得到了相当的重视。

几乎Applications实施的每一步都比传统的开发在更大程度上涉及用户。资源计划者一般不太关心用户的参与，这主要是因为用户在过去一直不占主导地位。第21章“项目组织和管理”中对Applications实施给予了特殊关注。

### 建立计划

一个财务项目一般要涉及大量的软件包、人员和任务。每个任务都不大，但是，许多任务不好理解，即使确实理解也难于评估。

即使对需要做什么不完全了解，也应该做一个计划。从Oracle、顾问和自己的员工那里听取最有用的建议。不过，比建立计划还重要的是随着你经验的增加修改该计划，即使这意味着要推迟期限也在所不惜。

### 理解公司的实际业务

在设置Oracle软件包时，必须对业务如何处理有一定的概念。虽然最大的效益是通过更改业务过程实现的，但是有许多公司决定先转到Oracle，然后再重组业务。

许多用户发现他们目前的业务定义得并不好，特别是在处理一些例外事情上。处理的定义可能很具个性，如“玛丽处理客户回执，去问她”。默认是另一种可能出现例外的大场所。可能所有的客户都在COD基础上接受，有某些例外；或者所有客户都有Net30的付款项，也有例外。

Oracle的全面设计为处理和例外提供了自动支持。即使根本不打算更改业务规则，Oracle的设置也将要求在每个应用范围内定义自己的当前操作模式。对于那些处理过程并未被记录下来的公司，这个似乎很简单的工作几乎总是要比预算花更多的时间。因此，在进度表中一定要留有余地。在建立项目进度表和为Oracle定义业务处理过程两个方面，Oracle或过去曾经实现过Oracle Applications的外部顾问将会很有帮助。第21章介绍了如何编写业务处理、它们适合于放在项目计划的什么位置以及在测试场景和测试脚本中如何表示它们等。

## 理解Applications程序集

多数经历过Oracle Applications实施的人都已经在一个或多个方面成了专家。具有讽刺意味的是，正当一个单位已经转到了Oracle Applications并急需专家的时候，实施人员却走了。公司显然需要大量依赖Oracle的顾问和独立的咨询公司，以便在需要的时候得到专业性的建议。除了那些密切相联的机构员工，否则，谁都不会知道过去是如何设置系统的。

产品知识对软件包的实施比对于定制开发要重要得多。就Applications来说，必须在不确切了解软件代码是如何编写的、而只是在知道这个“黑匣子”的外表的情况下进行工作。安装Financials并非“通常的开发”，而且也不能那样处理。

本书提供了填补产品知识空白的一种快速途径。第22章“数据转换、培训和开通”为Oracle和内部培训提供了一个全面计划。然而，短期内培训还是不能代替经验。几乎每个客户都能够通过聘请顾问人员得到帮助，以加速他们自己的Oracle Applications安装。

## 从用户处获得所需级别的委托

虽然一个Oracle Applications实施中的任务与传统数据处理开发中的一样，但那是对系统用户、而不是对MIS部门来说的，系统用户必须起主导作用。软件正常运转时的工作是决定如何使用它。这当然不是程序设计人员的工作，而是职能领域人员的工作。

这项工作富有挑战性，但也很混乱。用户必须承担两项工作：保持业务与旧系统一道运转和为Oracle作准备。

许多公司，特别是小公司，没有编写为其自动化系统所支持的政策、过程和实际业务文献的习惯。Oracle要求用户采用从数据处理领域借来的系统工程规范。这套规范的学习和应用都需要花时间。

几乎每个Oracle Applications都要替换其他类型的自动系统，而这些原有系统的用户拥有基本的数据。通过MIS部门可能的帮助，用户必须承担整理和转换数据的任务。这是一个不显眼、很困难但又很重要的工作。“废料入，废料出”，从一开始就要重视。

任何新的系统都需要培训。初级水平的用户在操作的前几周一般都需要专家的支持，专家们可以采取帮助台（help desk）或超级用户形式。虽然MIS部门可能会维护帮助台，但用户部门必须在其人员使用Oracle时提供最终的支持。

用户部门如果没有从头到尾地经历过一个主要系统的实施，就不可能认识到其承担工作的规模或所负责任的重要性。项目计划、顾问的联合作用、MIS部门以及用户自身都使得工作的规模相当可观。用户方的资深管理人员需要保证能够得到所需的人员。

## 人员配备

要取得成功需要综合公司业务、Oracle应用、数据处理规范及项目管理等方面的专家意见。这干系重大，远远超出了软件包、数据处理设备、顾问和转换的明显代价。

人员配备的目标应该是组织最可能满足所有项目目标的人员，这些项目目标包括：

- 在实施中支持现行业务操作。

- 恰当的软件包设置。
- 完成和整理数据转换。
- 最终用户培训和支持。
- 数据处理与网络底层结构，使得能够将系统交付给用户。
- 维护、加强和升级Oracle Applications的经过培训的人员。

人员可以是来自MIS部门、用户部门、Oracle咨询、大集成公司和独立的咨询人员。一个成功项目的特点在于其具有弹性：能够恰当地调配人员，不管这些人员来自何处都能够组织他们并激励他们。

### 注重实效

Oracle Applications包含了大多数业务都具有的所有功能。尽管每个公司都具有自己的特殊性，它们仍能很好地和完整地共同完成功能。有时，某个公司具有新颖的运作方式，这给该公司提供了独一无二的竞争优势。其他一些公司只发展自己特殊的、但是成功的业务运作模式。

Oracle Applications的购买者应具有将Oracle的这把“利器”用于广泛用途的打算。然而，也必须承认，在进行系统扩展时会出现偶然情况，有时甚至需要修改Oracle的代码。对于这样的修改，项目管理员将要适当地建立基本的规则来进行价格/效益分析，并采用第20章“定制和修改”中的思想来管理这样的修改。

### 获得Applications的最大价值

Oracle Applications能够改进业务。或者确切地说，它们能够帮助改进业务，并使用户能够将业内最好的经验用于自己的业务。

虽然回报是很值得的，但是价格也很重要。随着你的业务的发展和Oracle功能的改进，必须具有重组业务以最大利用Oracle功能的意愿。虽然Oracle Applications的许多初期效益的成本较大，但是在加快发展、增加管理和服务于客户的能力方面你会取得更大的效益。

### 小结

购买Oracle Financials将使你现在和将来都处于一个优越的位置。它的内建功能将能处理大多数业务。它开始的设计就是为了在Internet上使用。它充分地利用了Oracle的面向Web的技术积累、Oracle的产品和它们的集成。Oracle的开放系统承诺以及Applications所选择的技术方向将会使用户在Oracle Applications和其他系统、原有系统和第三方系统方面的投资得到保障。

用户从Oracle Applications所取得的效益与对它们进行的投资成比例。他们至少将会提供一致的财务账目，解决用户的2000年问题，并且使利用Applications的所有部门的运作一体化。如果愿意的话，用户可以利用Applications极大地改进自己公司内外的沟通。用户可以实现一些策略，通过这些策略能够削减业务经费、节约现金、让所有新客户浏览自己的报价并进行联机购买以及使用户的供应商成为合伙人等。用户可比过去更详细地估计利润。安装Oracle Applications将使用户不断地改进自己的运作成为可能。

安装和操作Oracle Applications与开发传统的软件不一样。所需技能（包括项目管理的技能）的组合完全是崭新的。虽然购买软件包使用户无须研究许多传统的数据处理方法，但是也引出了大量的新要求。我们希望通过本书帮助读者了解这些软件包能完成什么工作，并了解要成功地安装和使用它们需要哪些步骤。

# 目 录

## 前言

## 第一部分 起 步

|                                               |    |
|-----------------------------------------------|----|
| 第1章 E-Business Suite概述 .....                  | 1  |
| 1.1 E-Business Suite的结构和范围 .....              | 2  |
| 1.1.1 记账方法 .....                              | 4  |
| 1.1.2 运作控制 .....                              | 5  |
| 1.1.3 软件包系列 .....                             | 5  |
| 1.2 实施Oracle E-Business Suite的益处 .....        | 6  |
| 1.3 目标是什么呢? 设置参数和期望值 .....                    | 7  |
| 1.4 小结 .....                                  | 7  |
| 第2章 实施Oracle E-Business Suite<br>——基本概念 ..... | 9  |
| 2.1 收集与业务有关的财务信息 .....                        | 10 |
| 2.1.1 可变字段 .....                              | 10 |
| 2.1.2 工作流 .....                               | 15 |
| 2.1.3 业务会计 .....                              | 16 |
| 2.2 满足复杂机构的各种需求 .....                         | 18 |
| 2.3 全球业务环境中的Financials .....                  | 21 |
| 2.3.1 建立业务模型 .....                            | 21 |
| 2.3.2 全球化的Financial Applications .....        | 23 |
| 2.4 小结 .....                                  | 24 |

## 第二部分 软件包的用途

|                                 |    |
|---------------------------------|----|
| 第3章 Oracle General Ledger ..... | 27 |
| 3.1 会计系统的核心 .....               | 28 |
| 3.1.1 General Ledger概述 .....    | 29 |
| 3.1.2 与其他财务模块的集成 .....          | 34 |
| 3.1.3 总账设置 .....                | 42 |
| 3.2 会计可变字段的设计 .....             | 46 |
| 3.2.1 应考虑的方面 .....              | 47 |
| 3.2.2 良好的设计原则 .....             | 53 |

|                                                |    |
|------------------------------------------------|----|
| 3.2.3 应避免的问题 .....                             | 55 |
| 3.3 总账报表制作 .....                               | 56 |
| 3.3.1 Financial Statement Generator .....      | 56 |
| 3.3.2 Oracle Financial Analyzer .....          | 62 |
| 3.3.3 Applications Desktop<br>Integrator ..... | 62 |
| 3.3.4 Oracle General Ledger 数据模型 .....         | 63 |
| 3.3.5 旧系统的转换 .....                             | 64 |
| 3.4 小结 .....                                   | 67 |
| 第4章 预算与分摊 .....                                | 69 |
| 4.1 财务会计与管理会计间的差别 .....                        | 70 |
| 4.1.1 财务会计的局限性 .....                           | 70 |
| 4.1.2 不同类型的报表 .....                            | 70 |
| 4.1.3 财务会计与管理会计的结合 .....                       | 71 |
| 4.2 预算 .....                                   | 71 |
| 4.2.1 预算过程 .....                               | 71 |
| 4.2.2 非赢利机构的预算控制 .....                         | 72 |
| 4.2.3 预算的类型 .....                              | 72 |
| 4.3 分摊 .....                                   | 74 |
| 4.3.1 分摊公式 .....                               | 74 |
| 4.3.2 自动分摊工作台 .....                            | 75 |
| 4.4 小结 .....                                   | 75 |
| 第5章 Oracle Payables .....                      | 77 |
| 5.1 应付款系统的目的 .....                             | 78 |
| 5.2 业务中的Payables概述 .....                       | 81 |
| 5.2.1 决定债务的时间 .....                            | 81 |
| 5.2.2 Payables工作流程 .....                       | 81 |
| 5.2.3 Payables周期 .....                         | 83 |
| 5.2.4 供应商数据 .....                              | 89 |
| 5.2.5 记账方法 .....                               | 90 |
| 5.2.6 Payables业务的记账项目 .....                    | 91 |
| 5.2.7 与其他 Financials方法的集成 .....                | 93 |