

第32章 系统管理员设计器

本章要点：

设计器：Oracle 8i的常用计算机辅助工程解决方案

设计器一览

设计器管理

提高设计器性能

应用编程接口

故障检测设计器

32.1 设计器：Oracle 8i的常用计算机辅助工程解决方案

首字母缩写词CASE代表计算机辅助软件/系统工程（Computer-Aided Software/Systems Engineering）。传统上，系统开发过程几乎不包括在系统开始运行之前的计划，而且信息技术（IT）组也没有持续地与系统的用户进行交流，这导致系统不能满足用户的期望。今天，许多组织依靠一种正式的方法论去指导他们的系统开发任务，方法论中有特定的关于在项目的特定阶段完成什么工作的准则，这会产生更好的系统。购买CASE工具或正在考虑购买CASE工具以帮助进行系统开发任务的企业数量正在不断增长。Oracle 8i流行的CASE解决方案是设计者工具集。

32.1.1 系统开发生命周期

生命周期的概念指的是将一个大型的项目分割为一些较小的称为阶段的单元。每个阶段由许多特定的任务组成，这些任务必须在下一阶段开始之前结束。生命周期从最初的方案概念开始，随后是计划和设计、编程、测试、实现，并最后使用系统。通常，生命周期在初始阶段花费较多的时间，此时从用户中收集关键性的数据，并且形成适当的文档。企业必须理解这一阶段的重要性，如果他们匆匆完成这一阶段，结果的系统也许不是用户所期望的。

传统的系统开发生命周期（SDLC）至少由五个阶段组成：

计划——包括获得项目范围与项目如何进行的一个全面想法。

分析——包括决定用户对被提议的系统与项目计划的要求。

设计——包括数据库设计与程序模块化设计。

实现——编程与测试系统。

支持——系统投入使用，训练用户使用系统，并提供对系统的支持。

Oracle 8i公司有一个专用的方法论，称为Oracle 8i CASE方法。客户可以购买CASE*方法任务与交付使用（CASE*Method Tasks and Deliverables），其中有每一阶段的详细任务与结果。

在此有一些需要知道的术语：

应用——软件工程要素的逻辑组，它可以代表从高级概念模型到特定系统模型的任何

一种模型。

DES2K知识库/实例——一个拥有一套数据库对象的 Oracle 8i帐户。每当你在DES2K中创建、编辑或删除事物时，底层表的记录由 DES2K通过特定的包进行修改。

下属用户——可以通过同义词访问知识库的 Oracle 8i帐户。

知识库——由用于与数据交互的表与视图以及管理它们的过程组成。

32.1.2 上游CASE与下游CASE

当今常用的关于 CASE的术语是指给定的 CASE工具是上游 CASE还是下游 CASE。上游 CASE工具用于决定系统的需求并产生图表和文档，用户可以在项目的早期阶段查看这些表与文档。上游 CASE工具不用于生成真正的数据库或程序模块。下游 CASE工具主要考虑SDLC的设计阶段与建立阶段。它们通过生成创建数据库与程序的语法提高了生产力。就其本身来说，两种工具都不充分，因为上游 CASE工具没有增加生产力，而下游 CASE工具不产生用户用于验证的文档。

32.2 设计器一览

设计器提供了一个完全的上游 CASE与下游CASE解决方案并支持整个SDLC。

在分析阶段创建的逻辑模型成为物理模型的基础，物理模型在设计阶段创建。应用可以使用这个模型生成，因而任何商务变化可以立即反映到应用中。用户也可以使用非 Oracle 的 CASE方法。

安装设计器的最小要求如下列所示（客户端，然后是服务器端）：

奔腾90MHz处理器或更高的处理器。

CD-ROM驱动器。

300MB-600MB空闲硬盘空间，依赖于选择的选项。

最少40MB交换空间以运行应用。

32M可用RAM。

1GB硬盘。

NT4.0、Windows 95或Windows 98。

SQL*NET 2.3.4版本（如果在远程服务器上运行时，需要使用SQL*NET）。

SQL*Plus 3.3.4或SQL*Plus 8.0.4（对Oracle8数据库）。

NET8客户8.0.4（对Oracle8数据库）。

显示分辨率设置至少为 640 × 480。

服务器端要求如下：

Oracle7 7.3.4版本或更高版本

至少18MB共享池大小（Shared_pool_size）。

建议使用基于代价的优化模式。

至少30MB可用的RAM。

SYSTEM表空间中有60M的空间，用于存储PL/SQL包、过程与视图。

回滚（ROLLBACK）表空间中有70MB的空间用于知识库的安装。

SQL*Plus 3.3.4或SQL*Plus 8.0.4（对Oracle8i8数据库）。

Net8客户8.0.4 (对Oracle8i8数据库)。

对基于PC的服务器使用 Windows NT4.0、Windows 95或Windows 98。

对基于UNIX的服务器使用SUN Solaris 2.5或相当的UNIX系统。

注意 参考安装手册以决定存储要求。

关于产品可用性的最新信息, 请与 Oracle8i世界范围客户支持 (Oracle Worldwide Customer Support) 联系。在写本书的时候, 可用的最新版本是设计器 R2.1.2, 并且已在下列软件上做过测试:

Windows NT4.0、Windows 95与Windows 98。

Oracle 7.3.4或更高版本。

设计器对象扩展 (对象数据库设计器 R2.1.1)。

具有1998年10月应用补丁的开发器 2.1产品。

WEB 服务器3.0.1、3.0.1与4.0。

Netscape 3.04金版、Netscape 4.0.6、Internet Explorer 3.0.2与Internet Explorer 4.01。

Visual Basic 4.0或更高版本。

注意 应该确信所有的客户工具与知识库具有相同的版本级别, 以避免兼容性问题。

32.2.1 设计器/2000 R1.3.2的改变

设计器2.x版本提供了多种新的功能, 这些功能总结如下:

扩展服务器生成包括如下:

对Oracle7、Oracle Rdb与Oracle Web服务器的支持。

对非Oracle8i数据库的支持: MS SQL Server、Sybase、DB2/2与Informix。

对ODBC数据库的支持。

DML的API代与对表的锁过程。

客户生成包括如下:

设计重用: 在定义模型之后, 组件可以与其他模型共享。

应用逻辑: 存储PL/SQL、Visual Basic与Java代码。

生成器集成。

基于知识库中的公布设计信息生成数据库触发。

应用逻辑的100%逆向工程设计与100%的生成。

设计编辑器包括如下:

一个单独的、完全集成的设计与生成环境。

通过使用向导、拖放、就地编辑、鼠标右键菜单与同步更新的方法最大化使用的方便性。

升级到设计器R2.x的方法如下:

设计器R2.x可以执行在设计器2000 R1.x知识库上的更新。

应用升级包括:

在RON (Repository Object Navigator, 知识库对象导航器) 中的恢复工具, 在“设计器系统管理”中对RON有详细讨论, 它可以用于将一个早已存在的1版本应用或一组

应用到升级设计器 R2.x。一个两步过程将归档的应用升级为 1.3.2 可兼容的应用，然后再将其升级为 R2.x。这种方法的缺点是如果升级过程由于错误被中断，不得不使用最初的归档重新开始。为阻止这种事件的发生，在归档之前分析数据。

版本 1.x 与版本 2.0 中可以共存的，如下所示：

16 位版本 1.x——是，只要 16 位 1.x 版本与 32 位 2.x 版本在不同的 Oracle8i HOME 下。

32 位版本 1.x——是，将 2.x 版本安装在已有的 1.x 版本之上将不会产生问题，因为在 2.x 版本中缺省的路径名是不同的。

注意事项如下：

注册表中的 NLS_LANG 参数必须与服务器的 NLS_LANG 参数匹配，否则在安装与升级过程中装载包定义时会产生问题。

典型的升级将花费 2~3 小时用以删除现存的 API，还需要 2~3 小时装载新的 API 并执行实例转换与初始化。

如果 RAU (Repository Administration Utility，知识库系统管理工具——在“设计器系统管理”一节中详细讨论) 调用 SQL*PLUS，不要手工调用 SQL*PLUS，这会导致交出处理，甚至会毁坏知识库。

32.2.2 设计器组件

设计器工具集包括下列组件。

知识库服务包括如下工具：

知识库对象导航器 (RON)。

矩阵表。

知识库管理工具。

进程造模。

系统造模包括如下工具：

实体关系图表。

功能层次图表。

数据流图表。

系统设计器包括如下工具：

设计编辑器。

生成器。

设计器组件如图 32-1 所示。

设计器接口由不同部分组成，描述如下：

建模系统要求——这个区域中的工具可以用于建模商务处理过程、创建 E-R 图，并创建系统中的实体模型、功能模型与数据流模型。

生成初步的设计——这个区域中的数据仓库变换器与应用变换器可以用于从早期创建的模型中创建设计。

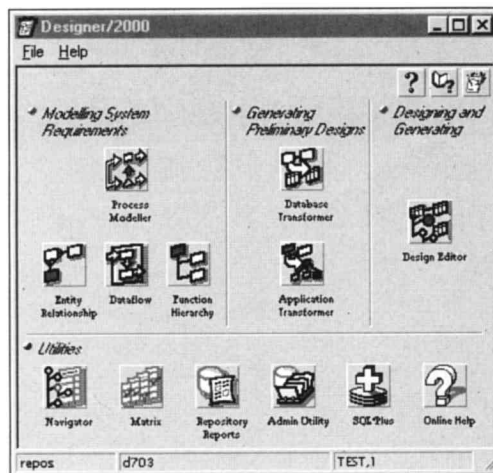


图32-1 设计器接口含有基于功能分组的不同组件

设计与生成——这个区域中的工具可以用于设计一个代表系统功能的系统。可以使用知识库中的信息创建服务器端组件与客户端应用。

工具——这个区域中的工具可以用于管理知识库，交互式地书写基于 SQL 的查询，并获得联机帮助。

32.2.3 设计编辑器

设计编辑器代替并强化了数据模型图 (Data Schema Diagrammer) 模型数据图 (Module

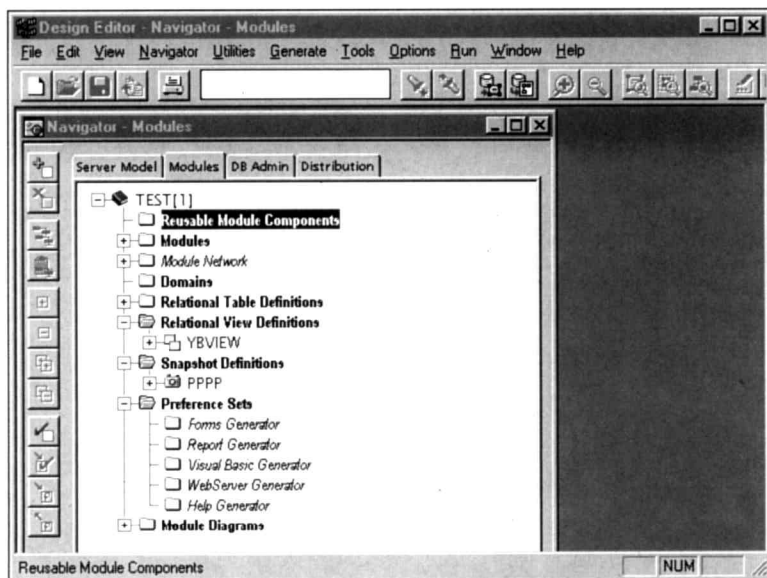


图32-2 设计编辑器导航器模型

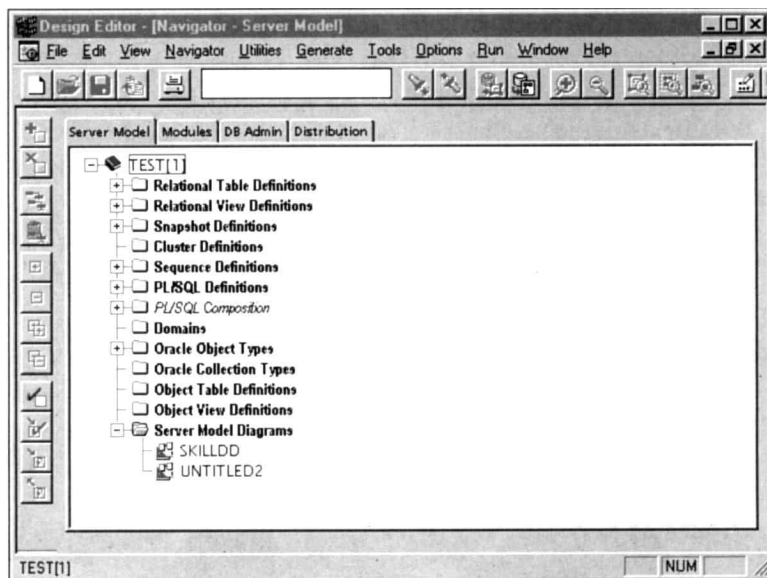


图32-3 设计编辑器导航器服务器模型

Data Diagrammer)、模型网络图(Module Network Diagrammer)、模型逻辑导航器(Module Logic Navigator)与优选导航器(Preferences Navigator)的功能。设计编辑器是一个集成的环境,使你可以在设计时执行不同的任务(参见图 32-2与图32-3),包括如下:

模型图的创建。

数据图的创建。

拖放功能。

导航器,可以在图表范围之外对数据模型进行操作。

32.2.4 理解知识库

设计器的核心是知识库,它可以记录所有输入设计器的信息,它是 SDLC实现阶段的基础阶段,包括以下功能特性:

知识库对象导航器(RON)——知识库服务的一部分,使用一种高度直接的方法查看与操作知识库对象,如图 32-4所示。

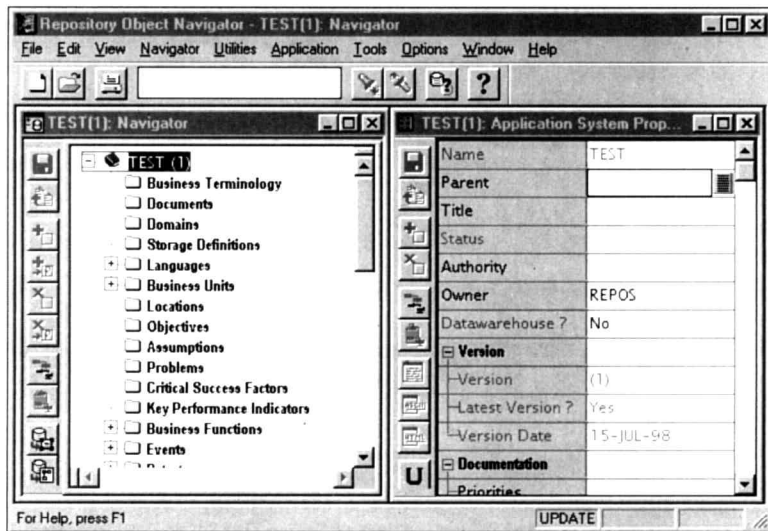


图32-4 知识库对象导航器组件

知识库基础——这允许多个用户并发访问,登记和注销工具,也使得可以从其他产品如SQL*PLUS访问数据。

版本控制——这使得应用可以从冻结的当前状态继续开发,它可以维护系统开发时的历史状态,也可以使用一个或多个版本进行应用的并行开发。例如,可以开发一个同时在美国与英国使用的存货系统——一个版本含有特定的英国细节,另一个含有特定的美国信息。

质量检测——通过相互参照与检测一致性规则,提供了在初始阶段的多种质量检测,可以在最初阶段修正错误。

文档——提供了一种报表选项的扩展集,以生成系统文档。在项目进行的过程中,所有的信息输入相应的表单与图表中。可以容易地获取报表。可以运行一百多种标准报表。

国际语言支持——这使设计器可以运行在以任何字符集建立的 Oracle8i 数据库上，包括多字节字符集。

还可以从知识库获得一些报表，如图 32-5 所示。

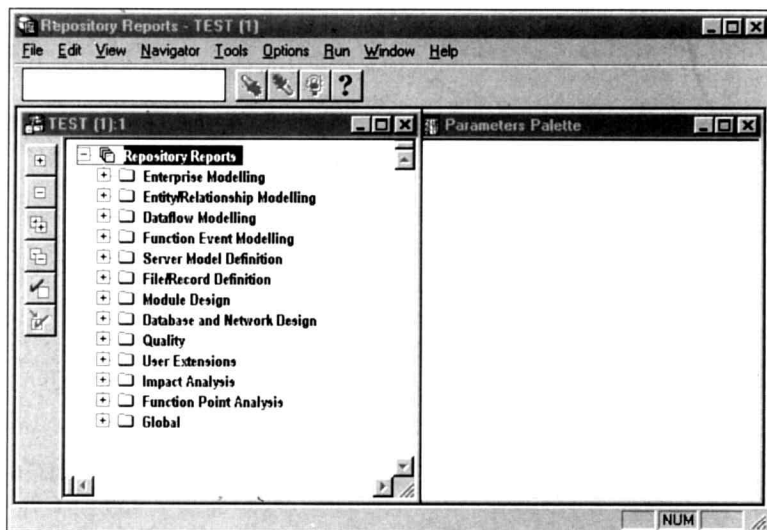


图32-5 知识库中可用的报表

32.2.5 使用图表

设计器工具集的所有图表的共同特性如下：

图形用户界面

点击能力。

下拉菜单。

上下文相关的帮助。

多窗口。

与知识库集成。

OLE 2.0集成。

分层视图。

与DES2K一起提供的所有的图表与工具具有一些共同特性，可以使用相似的过程与它们交互。在安装设计器之后，可以在设计器组中看到一个 Oracle8i设计器图标，双击这个图标，并注册到知识库所在的 Oracle8i数据库。如果正打开一个现有的应用系统，在该点选择它，否则，可以在该点创建一个新的应用系统。

注意 可以使用以下语法从命令行启动设计器窗口：

DES2KXX 用户名/口令@数据库 /A:apps_system, 版本/S。用户名/口令@数据库是连接串，/A指明应用系统与版本，/S用于关闭一闪而过的屏幕。

要创建一个新的应用系统，需要完成以下步骤：

- 1) 从设计编辑器屏幕选择知识库对象导航器 (RON)，依次选择 File | New Application。
- 2) 为应用系统创建一个名字。

可以继续使用这个应用系统工作，或通过从文件菜单中选择一个不同的应用系统，改变当前应用系统。

不同工具与应用的状态行含有关于当前会话的有价值的信息，如应用系统、版本与用户注册。所有的图表具有相似的外观与感觉，它们有一个多文档界面（MDI），使用户可以同时从不同的视图、通过不同的图表看到相同的对象。

图表大量使用鼠标来，操作对象与响应对话框。支持所有的标准 Windows 鼠标行为，如拖放、选择、双击改变属性、移动、重新设定大小等等。有一个完全功能的菜单系统，可以用于对对象与应用系统的简易维护，还提供了工具栏作为菜单系统的扩充，使开发者能够容易地访问最常使用的功能。DES2K 实用工具也有一个公共外观与感觉。使用两种类型的工具：

全窗口工具，如知识库对象导航器、知识库系统管理工具、知识库报表与矩阵图，提供与图表相似的 MDI 界面。这些工具提供了一个对象导航器与属性窗口以便能够容易地维护对象。

弹出窗口工具，如生成器与逆向工程工具，通常从另一个图表或工具中启动。

32.2.6 设计器使用的图表技术

在系统的逻辑设计与物理设计过程中，图表帮助你精确地形成计划系统的文档，并与用户和其他组成员交互。下面的设计器图表通常用于数据库的物理设计与逻辑设计过程：

实体关系图（Entity Relationship Diagrammer，ERD）——ERD 模拟一个组织所需要的信息。应该包括项目组与最终用户之间的交流与一致意见。实体关系建模包括确定组织中的重要事物（实体），这些事物的性质（属性），以及它们怎样相互关联（关系）。一个简单的两实体 ERD 如图 32-6 所示。

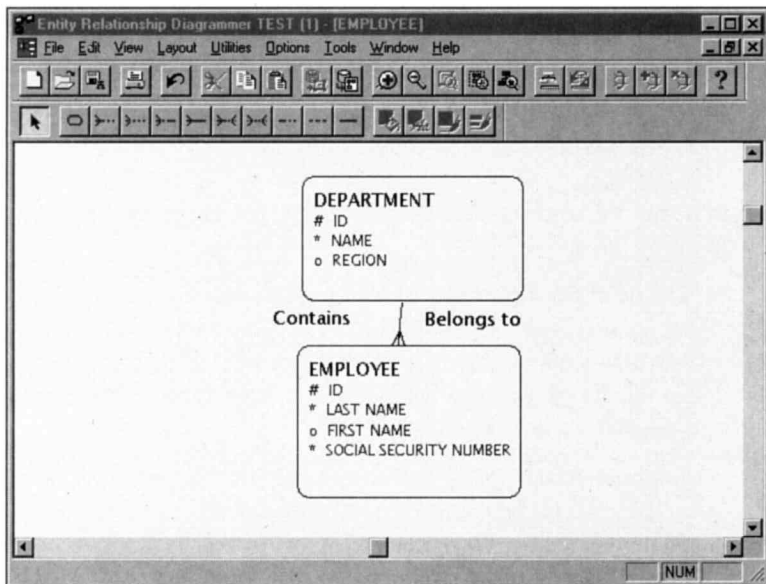


图32-6 两实体关系图示例

功能层次图（Function Hierarchy Diagrammer）——确定企业需要做的每一件事，并显示企业执行的功能与将来想要实现的功能。功能层次以全面的任务描述开始，然后分

解为一些主要的功能，并继续进行分解直至不能再分为止。

数据流程图（Dataflow Diagrammer，DFD）——DFD描述商务活动中的信息流程及它怎样在外部环境和系统接口之间流动。

矩阵图（Matrix Diagrammer）——设计器的知识库服务可以为系统生成多种矩阵图。矩阵的目的在于标识在逻辑设计中可能被忽略的区域。

过程流程图（Process Flow Diagrammer）——设计器的过程建模组件，支持过程建模技术，可以用于支持业务处理重组技术（Business Process Re-engineering，BPR）或原型。也可以作为一个有关系统要求分析的理解给予用户的可视化反馈。

32.2.7 生成器

设计器提供了一些代码生成器，这些知识库工具可以用于在系统开发生命周期的建立阶段生成完整的、没有故障的代码。生成器的集合可以被分为两种类型：服务器代码生成器（Server Code Generator）与前端代码生成器（Front-end Code Generator）。

1. 表单生成器（前端代码生成器）

这是一个DES2K组件，可以基于知识库信息快速建立 Developer/2000应用。表单生成器具有许多功能，包括如下：

- 生成完全特性与菜单的表单应用。

- 生成具有图形用户界面项的应用，如按钮、选择框、选择按钮组等等。

- 提供外键验证。

- 提供商务规则的实现。

- 启用Oracle 表单应用的逆向工程设计并将它们存入知识库。

- 重新生成表单，而不会丢失由开发者增加的代码。

- 提供对16位版本的VBX控件与32位版本的OCX控件的支持。

- 提供OLE 2.0支持，并与基于Windows的桌面应用集成。

2. 报表生成器（前端代码生成器）

报表生成器是一个DES2K的组件，可以基于知识库信息快速地生成 Oracle8报表应用。它的主要功能包括如下：

- 支持多重连接报表。

- 提供许多报表风格与布局，包括控制中断、主从与矩阵报表。

- 增加开发者的生产力。

- 提供标准化模板。

- 可以定制生成器操作。

- 提供多种总计函数。

- 可以启用现存报表进行逆向工程设计，并创建存储在知识库中的模型定义。

- 使用改变的定义重新生成报表。

3. 服务器生成器（服务器代码生成器）

这是一个设计器的组件，提供基于知识库信息创建数据库对象与 PL/SQL模型的手段。它的功能包括如下：

- 为数据库对象创建生成 ANSI标准的SQL。

生成检查约束。

生成并校验PK-FK关系。

支持Oracle 7基于角色的安全性。

生成PL/SQL模块，包括触发器、过程、函数与包。

支持快照。

逆向工程设计数据库对象并在知识库中存储它们的定义。

本质上，需要在知识库中定义对象，然后使用这个生成器指明希望脚本创建哪个对象。

4. Visual Basic生成器（前端代码生成器）

这个生成器与表单生成器相似，该生成器生成的代码可以被载入 Visual Basic6.0中，并可被用于创建一个可执行文件。

5. WebServer 生成器（前端代码生成器）

WebServer生成器可以用于创建Web应用。为创建一个WEB应用，应遵循以下步骤：

1) 在模型数据图中定义模型。

2) 使用WebServer从模型中创建PL/SQL包。

3) 在Oracle WebServer上运行生成的PL/SQL脚本以创建Web应用。

6. MS帮助生成器（前端代码生成器）

MS帮助生成器生成微软（MS）帮助文件格式的帮助文件，可以与表单和 Visual Basic应用一起使用。微软帮助生成器（MS Help Generator）将帮助文本放入一个文件中，并创建必要的链接与需要的跳转。还需要微软帮助编译器（MS Help Compiler），这是一个单独的产品，用以生成WinHelp格式的文件。

7. C++对象层生成器（服务器端代码生成器）

这个生成器使你能够通过创建 C++类定义并使用系统建模工具代码，访问对象关系型数据库。它可以用于为组成对象的数据与函数产生类。需要将生成的类与其他的 C++代码一起编译并将它与C++库链接。

32.2.8 模型重建机制

所有的生成后期工作应该以这种方式实现：模型的重新生成应该或多或少以一种简便的方式完成。下面的机制可以完成这项任务：

将所有信息放入知识库。这个方法确保改变自动地被应用到重新生成模型中。然而，这个方法导致对知识库系统管理的增加。

使用多个模板：一个基模板与其他特定表单模板。基模板可以用于系统中的所有表单，也可以将表单特定信息放入其他的模板，从而使对代码的管理变得容易。

32.2.9 Oracle CASE交换机

Oracle CASE交换机是一个可以在设计器与其他厂商的 CASE工具间进行信息双向交换的工具。这对于原先使用其他厂商的 CASE工具，而现在对使用设计器有兴趣的用户来说是非常有用的。如果一个组织使用其他的 CASE工具进行分析，如LBMS Automate Plus，现在想要使用设计器完成这个项目，那么 Oracle CASE交换机更为有用。DES2K的一个限制是它不能生成3GL代码。使用CASE交换机，可以将知识库中的信息转换到另一个 CASE工具中，并从另

一个CASE工具生成3GL代码。

Oracle CASE交换机支持下列CASE工具。

上游CASE支持包括如下：

Knowledgeware（智能软件）公司的ADW/IEW。

Intersolv公司的Excelerator。

Anderson Consulting公司的Design/1。

LBMS Automate Plus。

下游CASE支持包括如下：

ASK Ingres 6.0。

Pansophic Telon。

CASE交换机加载前提供的校验检测如下：

没有属性的实体。

没有实体的属性。

没有关系的实体。

没有唯一标识的实体。

与目标工具的可兼容性。

32.2.10 使用设计器的瀑布定向方法

瀑布方法论对于CASE学习很流行，它利用以下阶段实现：策略、分析、设计与实现。在这个过程中可以使用的不同的设计器组件如下所示：

1) 创建一个APPLICATION，它将是你的工作区域。使用知识库对象导航器，如图 32-7 所示（选择File | New Application）。

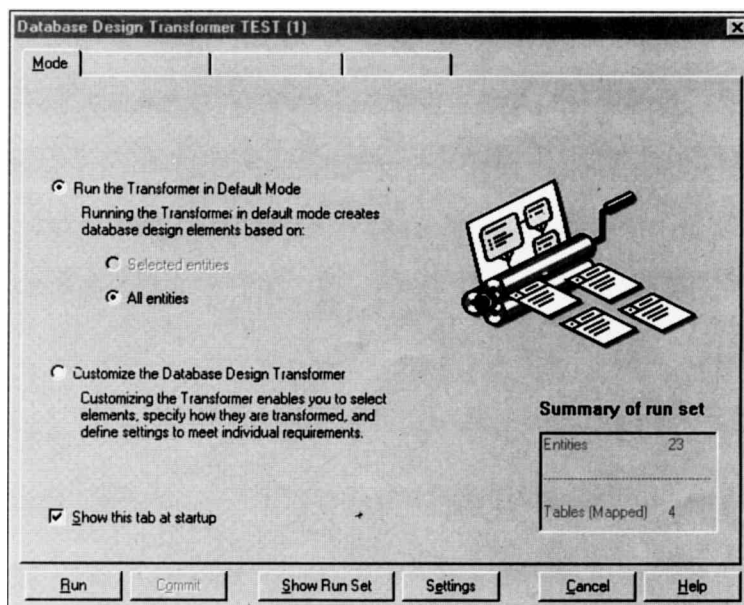


图32-7 数据库设计转换器组件

2) 使用实体关系图, 创建实体与它们的关系。使用 Edit | Domains 输入它们的有效值、它们的缺省值。双击一个实体以输入属性并指明它的作用域与关系, 同时增加缺省属性与有效值 (见图32-8)。

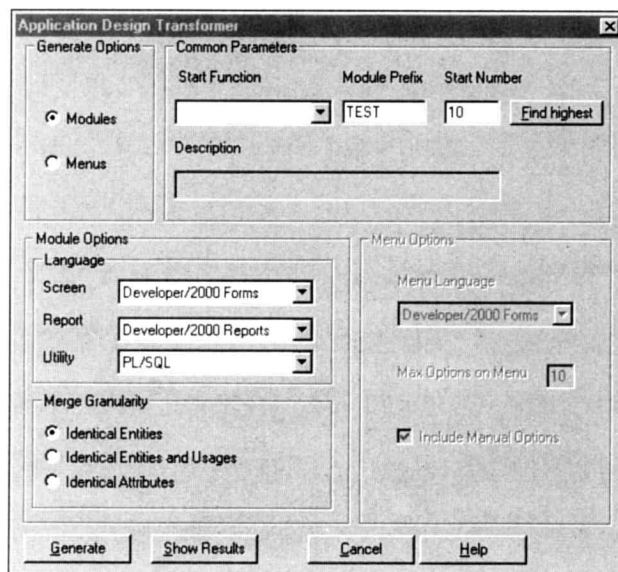


图32-8 应用设计转换器组件

3) 使用功能层次图创建这个图表。双击每个功能, 增加附加的信息以决定它是一个模型还是一个表单。

4) 使用矩阵图, 指明功能与实体属性的相关性。

5) 使用不同的报表以观察分析的质量。

6) 使用属性窗口定义一个数据库对象所在的区域。

7) 使用数据库设计转换器, 指明将被转化为表的所有实体。

8) 使用RON编辑表定义。

9) 使用RON增加违反规则的错误信息。

10) 使用RON检察约束以实现数据完整性约束。

11) 创建序列类型列以描绘主表的细节表 (detail table)。

12) 使用数据图编辑列定义。

13) 使用服务器生成器生成DDL。

14) 使用RON的应用向导, 创建应用的一个图形用户界面 (GUI) 模型。

15) 使用模型数据图编辑模型定义。

16) 使用相应的生成器 (表单、报表、菜单等等) 生成应用。

表32-1显示了最常用的任务, 以及可以实现它们目的的工具。

表32-1 用于特定任务的设计器工具

| 任 务 | 工 具 |
|-----------|-----------|
| 建立过程流程模型 | 过程模型或数据流图 |
| 创建一个实体关系图 | ER图 |
| 列出实体与过程 | 知识库报表 |

(续)

| 任 务 | 工 具 |
|---------------|-----------------------|
| 将实体使用映射为过程 | 矩阵图 |
| 功能分解 | 功能层次图 |
| 从一个阶段移动到另一个阶段 | RON (应用版本) 原型/生成器 |
| 实现GUI标准 | 设计编辑器 |
| 设计物理数据库 | 数据库设计转换器 (参见图 32-11) |
| 精炼物理设计 | 数据图与 RON |
| 应用设计 | 应用设计——转换器 (参见图 32-12) |
| 精炼应用设计 | 模型数据图、模型结构图、矩阵图、RON |
| 开发应用模型 | 生成器 |
| 数据库对象创建脚本 | 数据图、服务器生成器 |
| 设计的明智检查 | 矩阵图 |
| 文档 | 知识库报表 |
| 修正错误 | 依赖于应用 |
| 用户与知识库管理 | 知识库对象导航器、知识库系统管理工具 |

32.3 设计器管理

设计器管理由用于监控与管理一个项目数据的技术集组成。DES2K提供的执行这些功能的工具可以通过文件、程序、知识库对象导航器 (RON) 中的应用下拉菜单与知识库系统管理实用工具 (Repository Administration Utility, RAU) 获得。系统管理任务包括知识库建立、对知识库用户授予权限、升级、版本控制以及数据元素共享。图32-9显示了知识库系统管理工具的界面。



图32-9 知识库系统管理工具组件

32.3.1 理解知识库

知识库作为标准Oracle数据库而被实现，它由存储关于系统被分析、设计与生成信息的表

组成。具有足够权限的知识库用户可以创建一个或多个应用系统，并将应用系统级的访问权限授予其他知识库用户。只要另一个应用系统的拥有者给予了他们共享访问该系统的权限，应用系统的创建者/拥有者可以从另一个应用系统将对象共享到他们的应用系统中。然而，这些共享对象只能在它们被创建的应用系统中进行修改。

为有效地使用知识库，理解它的设计与组成它的元素很重要。有一些由 Oracle 作为设计器软件一部分发货的图表对于理解知识库非常有帮助，它们可以帮助理解为什么设计器以这种方式工作。知识库由元素、这些元素之间的关联、以及元素或关联的属性组成。在知识库中可以存在许多类型的元素与关联，如实体、列、功能实体等等。

有两组知识库元素：主访问控制（Primary Access Controlled, PAC）与次要访问控制（Secondary Access Controlled, SAC）。主访问控制元素由应用系统拥有，删除一个主访问控制元素将删除它的所有次要访问控制元素。PAC 的例子包括表、功能与模型。次要访问控制元素由 PAC 拥有，一些 SAC 可以具有它们自己的细节 SAC。SAC 的例子包括属性、ID 等等。关联也需要一个双亲，并且不能独立存在，这一点与 SAC 相似。事实上，关联具有两个双亲——它连结起来的两个 PAC，当任何一个双亲 PAC 被删除时，关联也被删除。关联的例子包括功能-实体使用与功能-商务单元使用。知识库模型不允许使用关联的关联。

32.3.2 知识库大小

知识库系统管理工具具有一个功能，可以计算出知识库的大小为小型、中型或大型。安装手册提供了一张表，可以使用这张表决定某个大小的知识库所需要的表空间。表空间与回滚段分配的灵活性最小化了安装与升级过程中用完资源的可能性。

32.3.3 保护设计器知识库

有两个主要的方法可以用来控制访问设计器知识库：

知识库级——可以控制一个用户是否具有管理员或用户权限。

应用系统级——可以控制一个用户对独立的应用是否具有只读、插入、更新、删除、共享或系统管理权限。如果需要更好的控制，可以使用设计器的加锁能力。

知识库系统管理工具（RAU）的 Maintain Users tab（维护用户标签）可以用于创建新的知识库用户。RAU 只能由知识库的拥有者访问。这个标签含有如下信息：

Oracle 用户名——需要访问知识库的用户的名字（在数据库中）。

用户名——Oracle 用户的描述。

类型——管理者（可以在知识库中创建新的应用系统）或用户。

描述——用户注释。

点击 RECONCILE（调整）按钮以保存修改。

使用知识库对象导航器（RON）授予知识库用户访问一个特定应用系统的权限。在用户中只有对该应用系统具有系统管理权限的应用系统拥有者可以执行这种授权功能。通过选择应用，授予访问授予一定权限的结果描述如下表所示：

选择——给予用户对应用系统的只读访问权限。

插入——使用户可以在应用系统中创建新的主访问元素。

更新——使用户可以修改主访问元素，创建或删除次访问元素。

删除——使用户可以删除主访问元素。

共享——使用户可以在另一个应用系统中共享这个应用系统的定义。

系统管理——使用户可以应用系统级控制访问。具有这个权限的用户是应用系统管理员，系统管理员可以做应用系统创建者能做的几乎所有工作。应用系统管理员可以将对应用系统的访问权限授予其他用户、从其他用户回收应用系统访问权限、设置应用系统级权限、冻结与解冻参数选择、移去元素集上的锁以及执行许多其他功能。唯一的一个应用系统拥有者可以做，而其他系统管理员不能做的工作是改变应用系统的拥有权。

32.3.4 共享对象与传送对象

这些实用程序使多个应用系统可以使用一个元素的定义。表 32-2描述了这些不同的命令以及它们的使用。例如，如果有在一个应用系统中定义的 Codes表，而另一个应用系统中需要这个Codes表，那么可以共享Codes表，并通过使用设计器的设计重用特性（Designer's Reuse feature）以确保Codes表在所有的应用系统中的结构相同。共享定义的应用系统应该在相同的知识库中（参见表32-3）。

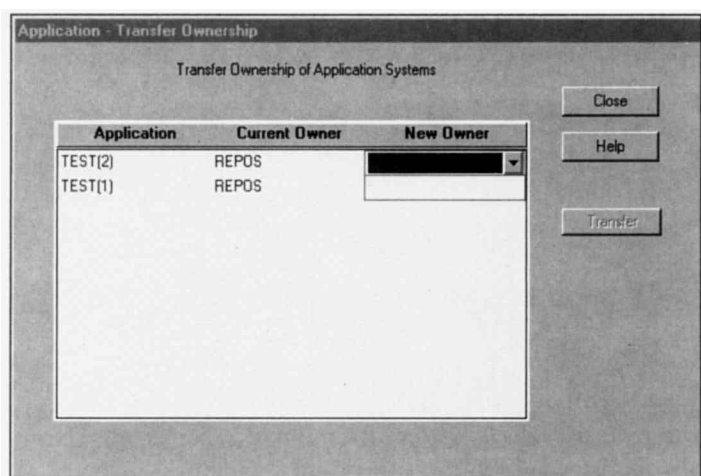


图32-10 转换应用拥有者

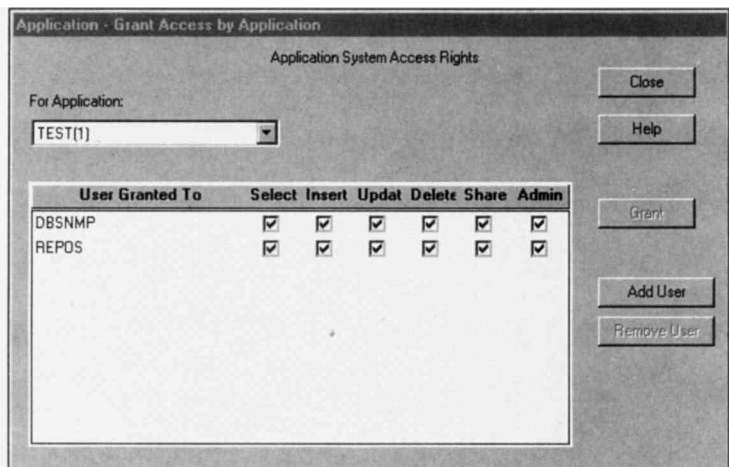


图32-11 授予访问权限

表32-2 在RON中共享与传输对象

| 命 令 | 描 述 |
|----------|------------------------------------|
| 应用、共享 | 应用系统声明允许另外一个应用系统使用该系统的对象。参见图 32-11 |
| 应用、不共享 | 一个应用系统声明不再使用另一个应用系统的元素定义 |
| 应用、转换拥有者 | 将一个主访问元素移动到另一个应用系统中，参见图 32-10 |

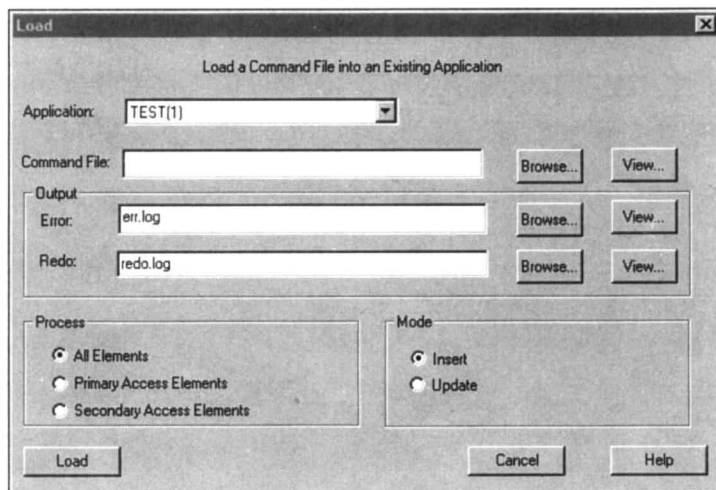


图32-12 知识库对象导航器装载工具

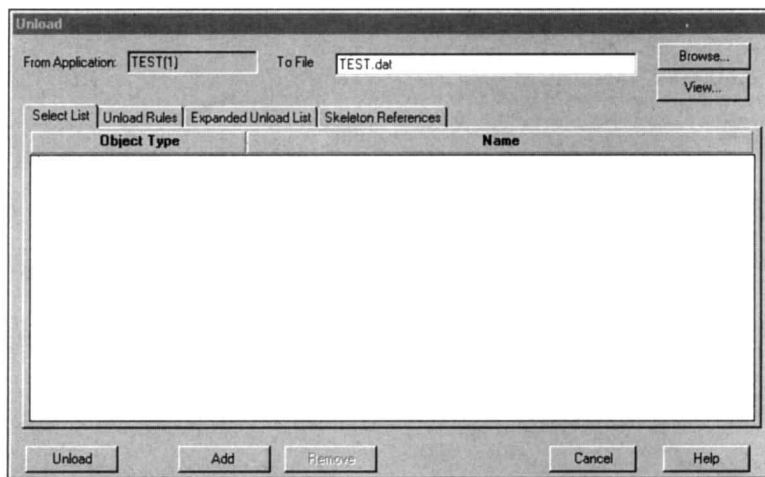


图32-13 知识库对象导航器卸载工具

表32-3 控制知识库中持有的声明

| RON中的命令 | 描 述 |
|----------|---|
| 实用程序、不装载 | 可以创建一个ASCII文件，该文件以专有形式存储以下定义：选中的主访问控制（PAC）对象、PAC的次要访问控制对象、与选中的对象相关的其余对象。参见图 32-12 |
| 实用程序、装载 | 与文件卸载相对。使装载的信息可以从另一个知识库或应用系统中卸载。参见图32-13 |

(续)

| RON中的命令 | 描 述 |
|--------------------------|--|
| 实用程序、检入、检出、 锁设置、解除锁设置 | 与装载和卸载相似，检入与检出允许在应用系统之间的相同的或不同的知识库移动选中的元素。检出还锁定源应用系统中的对象，因而阻止在检出期间对该对象的意外修改。检入解开对象上的锁。参见图 32-14与图32-15 |
| 应用、归档 | 提取知识库应用版本到一个特定的临时归档表集合 |
| 应用、导出 | 将临时归档表中的内容输出到一个操作系统文件中 |
| 应用、重建 | 用于将.DMP文件装载到临时归档表，然后载入知识库中 |
| 应用、复位 | 刷新临时归档表。知识库中应用的版本不发生改变 |

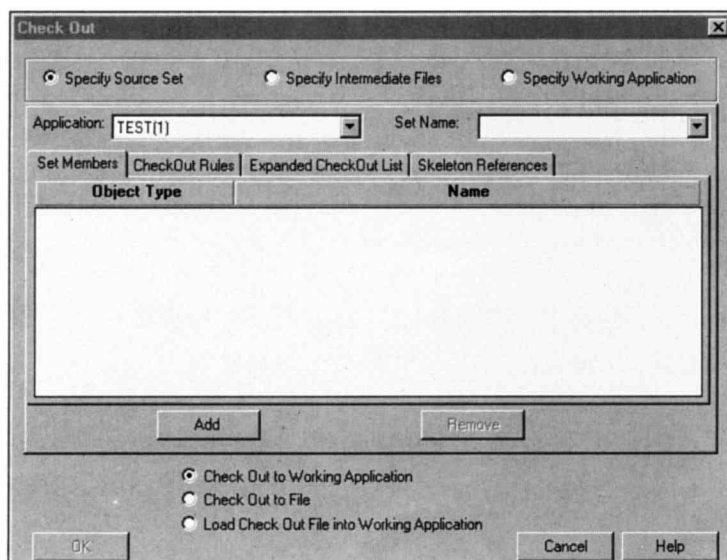


图32-14 知识库对象导航器检出

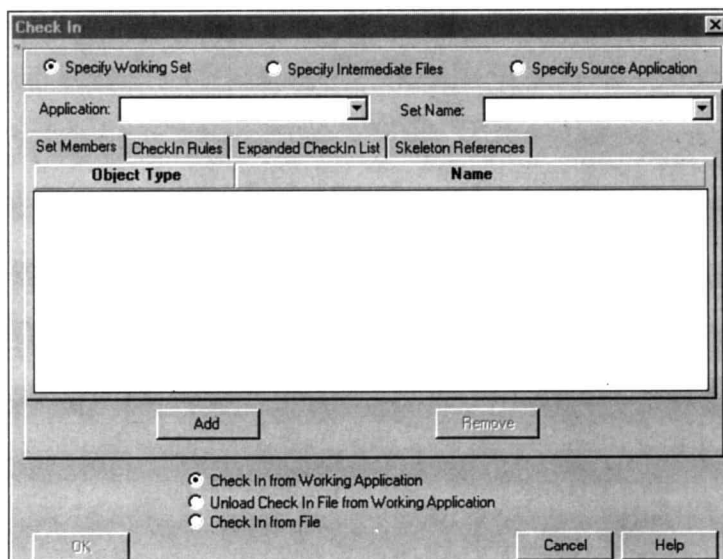


图32-15 知识库对象导航器检入

32.3.5 使用知识库的引用完整性

设计器 R2.x 具有一个工具，可用于检查知识库中含有数据的引用完整性。RAU（实用程序菜单）中的知识库分析器删除了冗余的或不合法的数据，确保了数据的完整性。为运行知识库分析器，知识库所有者必须被授予创建表与创建索引的权限。

执行以下 SQL 语句以检查知识库的状态。如果没有返回行，知识库的引用完整性是正确的。

```
SQL> select count (*)  
       from ckaz_actions  
       where check_no is not null  
       group by check_no, elmt_type, refd_elmt_type, action;
```

为使用基于成本的优化器，使用以下语句计算知识库表的统计信息：

```
SQL> analyze table <repository table> compute statistics;
```

Oracle 服务器可以使用 SQL 语句统计的最有效的路径访问被分析的对象。

32.3.6 版本控制与改变控制

对应用与管理类型具有管理访问权限的用户可以创建应用的一个新版本。为创建一个新版本的应用，应该连接到 DES2K，从 Design Editor 屏幕选择 RON 工具，然后依次选择 Application | New Version。

这会产生两个版本，一个版本是被冻结的，不能被修改，但可以被查看，而另外一个版本可以被修改。

如果一个应用系统与其他的应用系统共享元素，将为其他的应用创建占位程序的应用。

例如，如果有两个应用系统如存货（IV）与项目（ITM）共享对象，要创建存货应用系统的一个新版本，将创建 ITM 的占位程序应用（01ITM）。INV 的版本 1 与 01ITM 一致，INV 的版本 2 将与 ITM 一致。01ITM 被冻结。

要删除一个应用的某个版本，必须确保它的元素没有被其他应用使用。如果有正在被使用的元素，可利用如下步骤来删除应用：

- 不共享所有的共享元素。

- 同时删除所有的应用。

- 删除依赖的应用（假设没有内部依赖）。

需要考虑的问题如下：

- 版本控制是一个单用户工具，它将其他的用户锁在系统外。不要在普通商务时间或有人访问系统时运行它。

- 当运行时，它需要一个大的回滚空间，该空间与应用的大小成正比。

- 当创建一个新的版本后，应用在数据库中大约占据两倍的空间。

- 运行版本控制工具的时间与知识库的大小成正比，而不是与应用的大小成正比。

- 应用的版本可以在任意时刻被冻结或解冻，然而，解冻一个老的版本可能导致不一致性。

32.3.7 移植应用

下面几节描述在不同的应用之间移植软件设计的方法。

1. 导出/导入整个知识库

这个方法包括从源数据库将整个知识库转移到目标数据库中。转移的单元是 Oracle 帐户。必须在目标数据库中创建所需要的包。

需要完成下列步骤：

- 1) 使用RON，从备份组中选择Export Repository User（导出知识库用户）。
- 2) 将导出文件以二进制模式传送到目标机上。
- 3) 创建一个Oracle用户（参考安装指南获取详细信息）。
- 4) 使用RON工具，将用户导入到目标数据库中。
- 5) 在目标数据库中创建包。
- 6) 重新创建同义词。

当执行以下任务时，建议不要使用这种方法：

移动单个应用系统。

向一个早已具有其他应用的CASE实例中移入应用。

2. 卸载/装载一个应用系统

这个方法包括在知识库之间移动一个或多个应用系统，移植的单元是应用系统，通常用于在相同版本的DES2K之间移动应用。

需要完成如下步骤：

- 1) 作为对应用具有管理级与系统权限的用户访问 RON。
- 2) 使用Application | Reset清除提取表。
- 3) 使用Application | Archive, 将应用载入精炼表中。
- 4) 使用Application | Export将提取表导出到一个*.DMP文件中。
- 5) 将导出文件以二进制方式传送到目标机。
- 6) 在目标数据库中，使用 Application | Import，将*DMP文件装载入提取表中。
- 7) 使用Application | Restore将提取表载入基本表。

想要执行以下任务时，不要使用这种方法：

移动一个实例中的所有应用系统。

作CASE实例的备份。

32.3.8 移动主存取控制元素

通常有两种方法用于控制在CASE知识库之间移动主存取控制（Primary Access Controlled PAC）元素：卸载/装载与检入/检出。

1. 卸载/装载处理

这个方法被设计用于不需要任何限制的快速传输。元素被卸载装入一个扩展名为 .DAT的ASCII文件中，在保留元素之间的关联时，也卸载次要访问控制（SAC）元素。如果关联中的一个元素不是卸载集的一部分，它创建一个具有该元素名字的框架以完成关联。然后将元素从.DAT文件载入一个应用系统。

有两种装载模式用于解决冲突方案：

插入——导致冲突的元素将不被装载，在日志文件中记录下冲突之后，装载过程继续进行。

更新——使用.DAT文件中的信息解决这个冲突。

2. 检出/检入处理

用户定义集（User-defined set，UDS）是一个非共享的PAC元素，它在应用系统中具有与其他PAC元素的关联。一个元素可以与多个UDS关联，一个UDS可以按照需要被锁定与解锁。可以在一个被锁定的UDS上执行的操作有：拷贝元素、共享元素或更新在该元素与非锁定元素之间的关联。不可以在一个被锁定的UDS上执行的操作有：改变元素的所有权、更新元素的属性或锁定含有该元素的其他UDS。

检出与检入的处理在UDS上执行。检出时，一个UDS自动地在源应用系统中上锁，并在检入时自动地解锁。如果一个源应用系统中检出一个UDS，则创建了一个.RCO文件。如果在工作的应用系统中检出一个UDS，则产生一个.RCI文件，这两种文件格式含有UDS以及它的关联。

表32-4比较与对比了哪些装载/卸载可以使用，而检出/检入不能使用的不同情形。

表32-4 使用装载/卸载与检出/检入的情况

| 情 况 | 卸载/装载 | 检出/检入 |
|-------------------|-----------------------------|-------------------------------|
| 单用户环境 | 是，不需要追踪从一个应用系统中卸载的元素 | 否 |
| 多用户环境 | 否 | 是，需要持续追踪移植的元素。检出元素会被上锁，防止意外更新 |
| 使用其他应用系统的元素建立应用系统 | 是，这很简单，只进行一次处理，所以不需要定义一个UDS | 否 |
| 备份一个元素 | 是，这很简单 | 否，不需要定义一个UDS或备份整个应用系统 |
| 需要对元素严密控制 | 否，它非常灵活 | 是，对检出元素的严密控制可以被归档 |
| 分布式开发 | 否 | 是 |
| 需要对源进行控制 | 否 | 是，从源应用系统中检出的元素要被上锁，以阻止更新 |

32.3.9 在文档中放置设计器图表

设计器启用许多输出方法，并提供了将不同的图表输出捕捉到诸如字符处理器和基于HTML工具的目标的能力，这些目标包括 Netscape Navigator（Netscape导航器）与Microsoft Internet Explorer（微软互联网浏览器）。

1. 装载图表到MICROSOFT WORD

可以使用三种不同的方法完成这项任务：对象链接与嵌入（Object Linking and Embedding，OLE）、屏幕捕捉与打印文件。这些方法将在下面详细描述：

使用对象链接与嵌入（OLE）——设计器是V2.0 OLE兼容的，可以用作OLE包容器或OLE服务器。为将一个ERD加载到WORD，需要完成以下步骤：

- 启动MS WORD，然后打开一个新的文档。
- 选择Insert，从菜单中选择Object，选择Entity Relationship Diagrammer（ERD）。

为将一个WORD文档装载到ERD，需要完成以下步骤：

- 启动设计器。
- 从ERD中，选择Edit | Insert New Object 并包括一个WORD文档。

使用屏幕捕捉——有许多技术用于屏幕捕捉，包括使用 Alt+Print 打印屏幕，使用画图编辑并粘贴图象，或使用第三方工具如 PaintShop Pro 来捕捉屏幕。

使用打印文件——除非选择另一个打印机，否则设计器工具打印到缺省打印机。使用打印管理器，可以修改打印机设置以便使打印的目的地是文件。然后这个文件可以被加载到文档中。如果缺省打印机是一个传真机，可以直接从设计器发送传真。

2. 将设计器知识库数据放入万维网

使用一个第三方驱动器，如 Adobe Exchange，它支持 .PDF 文件格式，可以将设计器知识库数据放入万维网（World Wide Web）。使用这个驱动器作为缺省驱动器，将文件保存为 .PDF 格式。这个文件然后可以在 WebServer 上被下载，并且建立了一个 HTML 链接以能够使用任意 Web 浏览器，如 Netscape 或 Microsoft Internet Explorer，在 Web 上浏览该文件。

32.3.10 使用设计器的逆向工程

大多数组织拥有许多应用，他们想将这些应用载入设计器。完成这项任务有许多方法，下面是最常用的方法：

逆向工程 DDL——这个方法可以用于从现存的 Oracle 7 数据库中提取数据对象，并将其编写成知识库中的元素资料。然而，它不能演绎通过程序代码实现的约束。它可以从本地或远程 Oracle 8i 数据库逆向工程设计数据对象。使用这种方法要求知识库用户具有表、视图与快照上的选择权限，并且具有对函数、过程与包的执行权限。

逆向工程表单与报表——表单逆向工程工具提取 FORMS 版本 5.0 程序的信息，并将其放入 DES2K 模型定义与次要访问元素中。报表逆向工程工具与表单逆向工程工具非常相似，能够用多种语言逆向工程设计报表，包括 SQL*PLUS 报表、Oracle 预编译报表、SQL*ReportWriter 1.1 报表等等。

遵循以下步骤实现逆向工程设计表单和报表：

- 1) 在知识库中定义数据库。
- 2) 在知识库中定义主键约束与外键约束。
- 3) 将表单加载到 Oracle 数据库中，并通过一个批量作业或每次一个的形式逆向工程设计它们。
- 4) 手工插入任何没有得到逆向工程设计的逻辑。

注意 触发器、库附件、过程、过程调用不能被逆向工程设计。

从 RON、工具、表到实体更新改进——这个工具将物理数据模型转换为逻辑数据模型。这个方法的效果和使用依赖于物理模型与逻辑模型相互之间的匹配程度。当使用这个工具时最需要警惕的是，确保已在物理数据库设计中定义了所有的主键与外键约束。应该注意这种方法不能将一个不好的物理设计转换为一种好的逻辑设计。事实上，它只是按照原样进行转换，因此，如果想要节省转换完成后需要进行的大量工作，必须将物理模型尽可能地与所需要的逻辑模型相匹配。这种方法具有以下限制：

- 应该存在约束定义，否则所有的列会成为属性而不是关系。
- 不改进共享表定义。
- 不应该事先将表映射到实体，如果已经映射为实体，必须删除这个关联或实体的定义。

为运行改进工具，可以从 RON 或实体关系图（ERD）中选择 Utilities | Table Entity Retrofit。

通常执行完所有的改进之后，需要清除所产生的逻辑设计。要这么做，需要完成以下步骤：

- 1) 将关系名改为更具有意义的名字，因为由改进生成的名字直接取自外键限制名，通常并不直观。
- 2) 检查关系的类型，确保它们的确是所需要的关系。例如，强制外键成为选择性的 / 强制的关系，而选择性的外键成为选择性的 / 选择性的关系。
- 3) 重新命名属性，使其对用户更具有意义，因为在物理设计中，属性名可以缩写。
- 4) 对结果逻辑设计进行一次明智的检查，以确保实体与关系正是所假定的，因为物理设计可能会以不同的方式实现它们。

32.3.11 使用设计器进行数据管理配置

数据系统管理包括监控与控制一个项目的数据。为实现这个目的可以使用不同的配置，每一种配置都各有优缺点。设计器在 RON 与 RAU 中提供特性以简化这个功能。

数据管理配置由以下内容构成：清楚最适合你的组织结构的应用配置，同时知道这种结构最佳使用的知识库的数量以及需要支持的用户的数量。

1. 决定使用 CASE 应用的数量

在决定使用 CASE 应用的数量时，需要回答一些问题，包括如下：

- 组织的结构是什么？
- 团体之间的交流程度如何？
- 应用的大小是多少？
- 设计者对 CASE 技术熟悉吗？
- 控制与灵活性哪个更重要？

有一些机制可以使用，每一种有它自己的正面与负面效果。

2. 集中化配置

在集中化配置（centralized configuration）中，所有的数据由一个应用拥有，其他应用共享这些数据。

这种机制的优点包括如下：

- 易于管理，因为只有一组访问权限需要维护。
- 不需要元素传输，因为所有的元素都由一个应用拥有。
- 版本控制简单。

这种机制的缺点包括如下：

- 需要所有的用户访问一个数据库（单故障点）。
- 需要组之间更多的交流，因为他们共享相同的数据，因此不得不在许多事情上保持意见的一致性。
- 导致一个大的系统，因而工具的运行需要较长的时间。

3. 分散配置

分散配置（Decentralized configuration）意味着每个应用拥有它自己的数据，可以按需要将其元素的权限授予其他的应用。

这种机制的优点包括如下：

比集中方法具有更大的灵活性。

可以独立地进行应用系统的移植和版本控制。

导致较小的应用，所以工具运行速度较快。

可以访问的应用开发组数量的增加并不会增加相互锁定的可能性。

这种机制的缺点包括如下：

如果应用没有被很好地分区，会导致许多元素的共享与传输。

必须小心以确保没有冗余。

每个独立应用的界限和目标必须预先建立好，以使这个方式成功。

4. 联合配置

在联合配置（combined configuration）中，有一个主应用拥有多个应用通用的元素，并与需要它们的应用共享这些元素。也有一些应用含有应用特定的数据，这些数据不被其他应用共享。

这种机制的优点包括如下：

既可以共享元素，也可以进行独立应用系统的开发，同时具有两种方法的优势。

导致可管理的应用。

这种机制的缺点包括如下：

如果计划不当，会导致后期阶段复杂的版本控制与移植任务。

由于它所提供的灵活性，需要大量的系统管理精力。

5. 决定知识库的数量

当决定知识库的数量时，需要考虑几件事情，这些事情包括如下：

应用的数量以及它们之间的交互关系

- 如果有多个应用，它们共享元素，它们应该驻留在相同的知识库中，因为不支持知识库之间的共享。
- 一个应用只能存在于一个实例中，否则，会在后期阶段导致大量的合并。
- 没有共同点的应用应该放在不同的实例中，以便对它们进行更好的控制。

预期的用户的数量

- 某些工具，例如 Versioning（版本管理），运行在单用户模式下，如果这种工具被不同的应用系统并发运行，有必要将该应用放置在不同的实例中。
- 如果达到硬件的物理限制，需要考虑多个实例。
- 每个CASE用户需要一定数量的RAM、交换空间等等。有些情况下，考虑一个客户 / 服务器解决方案更有益处。

表32-5 使用不同配置的正面与负面效果

| 需要考虑的因素 | 集中化配置 | 分散化配置 | 混合配置 |
|--------------|-------|-------|------|
| 管理访问权限的时间 | 低 | 低 | 中 |
| 版本控制的容易程度 | 高 | 低 | 中 |
| 对并行系统开发的支持 | 低 | 高 | 中 |
| 版本经常变化的系统的性能 | 低 | 高 | 中 |
| 需要使用多实例 | 低 | 高 | 中 |
| 应用的相互独立性 | 高 | 低 | 中 |

使用多实例具有它的优点与缺点：

- 会产生一个容错系统。

- 不需要所有的用户访问相同的数据库。

使用不同配置的正面与负面效果在表 32-5中总结。

32.4 提高设计器性能

设计器的安装指南上有许多需要满足的硬件与软件要求。下面的讨论提供了改进设计器性能的建议（小心使用它们，因为这会影响其他的 Windows应用。）

32.4.1 优化客户机

客户服务器配置的设计器工作得很好。要求至少386机器，使用不同厂家的硬件没有显著区别，内存要求至少32MB，然而建议使用64MB内存。在系统中使用以下技巧优化内存的使用：

- 使用硬盘优化器定期清除硬盘上的碎片。

- 不要直接将工作目录设到LAN上。

- 定期清除Temp（临时）目录。

- 设计器最小需要40MB交换文件。

- 重新安排装载程序的顺序，驱动器可以帮助最小化碎片。

- 禁用屏幕保护程序。

- 卸载不用的字体。

32.4.2 优化网络

网络性能会影响整个系统的性能，应该细心监控。

得到SWL*NET追踪文件或置入探测器以确定网络性能。在哪里存在性能降低。通过在sqlnet.ora文件中设置sqlnet.expire_time=<time_in_minutes>使用死亡连接检测。这将使监听器可以每隔先前设定的一段时间就进行一次检查，以确保没有已死亡的连接。

32.4.3 优化设计器

要优化设计器，有许多事情可做：

- 重建CASE索引。

- 重新安排知识库数据以使用当前环境的正确配置。

- 决定使用的知识库的数量。

- 决定在应用系统之间是否正确地共享元素。

- 得到诊断信息以确保SQL语句被优化地使用。

知识库存储在数据库内自己的表中，因此，数据库优化技术，如消除磁盘碎片，也适用于设计器。

32.4.4 优化数据库服务器

数据库优化是一门艺术，由于数据库上条件的改变，必须持续不断地在数据库上执行优化操作。Oracle出版社的《调整Oracle》（Tuning Oracle）是一本非常好的书，含有可以用于

优化数据库性能的多种不同技巧。下面的技巧会有所帮助（参考本书的第 16至第20章，以获取调整数据库服务器的更多信息）。

- 使用正确的优化器模式（推荐对 R2.x使用基于COST的优化）

- 截断RM\$HASH_ELMS。

- 通过合理地固定对象并且不使用一个大的 sort_area_size减少磁盘碎片。

- 减少硬盘I/O。

- 减少竞争，保持一个高命中率。

- 设置LOG_CHECKPOINT_INTERVAL的值比重做日志的大小稍大一些，以便只在日志切换时发生一个检查点。

- 将DB_BLOCK_BUFFERS设为正确的值，以防止知识库因为 API回到硬盘。

- 增加重做日志的大小与数量。

- 确保数据文件与交换文件在不同的硬盘上。

- 在一个与含有数据文件磁盘不同的硬盘上，创建知识库索引。

- 将重做日志与回滚段放在不含有交换文件的硬盘上。

- 将API固定在共享池中。

32.5 应用编程界面

当使用设计器开发应用时，你会意识到有一些需要实现的活动甚至使用设计器所提供的最完善的工具也无法轻易实现。DES2K的用户扩展功能可以帮助将你所创建的扩展放入知识库中。API通过使你从知识库输入、输出信息以及创建用户扩展功能打开知识库。API基本上能够满足你使用任何方式维护知识库，因而越过了可能遇到的 DES2K功能性的限制。API允许使你可以根据项目需求，定制特定的功能，而不是被强迫地使用 DES2K功能性。这导致生产力的直接增长与生成的高比例。理解使用 API工作需要创建程序代码，这需要一个学习的曲折过程很重要。因此确保 DES2K没有真正地为你提供所寻找的功能是很重要的。从长远的观点来看，它将帮助你识别完成任务的最好方法。

32.5.1 使用API

API是一组在知识库拥有者模式中的数据库视图与 PL/SQL包，可以对知识库进行安全操作。知识库由一组表组成，并且这些表之间的关系没有形成文档。API由许多这些表的重要视图与PL/SQL包组成，允许你在这些表上不需要使用 DES2K界面就执行DML。

建议对SQL与PL/SQL应该有一个较好的了解，以便有效地使用 API。对你试图操纵的元素和属性有一个透彻的理解也很有用。在知道需要使用哪个属性与元素之后，查看 API文档，包括联机帮助以及与软件产品一同发货的内部模型图，以便对所能使用的 API有一个更好的理解。与设计器一起发货的内部模型图非常有用，显示了 API视图与它们的关系。这些图表分组为以下子集：

- 商务计划模型。

- 商务需求模型。

- 数据库管理模型。

- 数据库设计模型。

模块设计模型。

基于想达到的目标使用适当的模型。在理解了有关元素的细节以及 API的使用后，下一步工作是创建PL/SQL过程和包以进行必要的操作。

32.5.2 API视图与包

知识库基表的名字具有 SDD_与CDI_的前缀。API视图具有前缀 CI_。所有的知识库视图具有一些重要的列，包括如下：

CREATED_BY与CHANGED_BY，可以查询这些列找出元素的定义是何时被插入或更新的。

ID列，唯一识别知识库中的元素。

列，对应它们所代表元素的属性。

NAME列，保存在RON中所看到的名字。

联机帮助系统在决定视图描述时功能非常强大。

API包以CIO为前缀，有300多个PL/SQL包组成API，每一种知识库元素具有一个包，这些包含有具有下列前缀的过程：

INS（插入）。

UPD（更新）。

DEL（删除）。

SEL（选择）。

联机帮助系统还提供了有关包的有用信息。除了前面提到的 API调用之外，还有执行一组语句而不是单独的语句，因而作为事务模型的 API调用。事务模型在事务完成前临时禁止约束检查，结果是，可以建立复杂的关系，并且约束检查被延迟到事务提交的最后一刻。CDAPI包的主要责任是处理事务模型。

可以使用下面的步骤实现事务模型：

- 1) 声明将使用的应用系统与版本。
- 2) 打开事务。
- 3) 装载记录变量值。
- 4) 装载记录变量指示器。
- 5) 执行DML并进行校验。
- 6) 如果前面的步骤成功，则提交事务，否则，回滚事务。

32.5.3 API限制

虽然你已看到了API的强大而灵活的功能，但是它也有一些限制。例如，API不能操纵文本类型，如注释、描述、PL/SQL块以及WHERE子句。可以从 CDI_TEXT表查出文本，但建议不要在这些文本上执行DML命令。CDI_TEXT表的结构如表32-6中所描述。

表32-6 CDI_TEXT表的结构

| 列 名 | 是否可为空 | 类 型 | 注 释 |
|---------|----------|------------|------|
| TXT_REF | Not Null | NUMBER(38) | 元素ID |
| TXT_SEQ | Not Null | NUMBER(6) | 记录行号 |

(续)

| 列 名 | 是否可为空 | 类 型 | 注 释 |
|----------|----------|---------------|------|
| TXT_TYPE | Not Null | VARCHAR2(6) | 文本类型 |
| TXT_TEXT | Null | VARCHAR2(240) | 文本 |

API的另一个限制是不能直接从 Oracle Form中调用。

32.6 故障检定设计器

虽然设计器是“完美的”，也有发生非用户错误的时候，你需要判定导致错误的原因并解决它。这一节讨论一些常见的错误，以及用于诊断设计器错误的不同策略。

32.6.1 检查常见错误

在安装过程中，知识库系统管理工具检查它的工作，如果发生问题，则停止安装。常见的错误包括如下：

ORA-4030: out of process memory when trying to allocate <x> bytes

这是一个操作系统问题，原因是服务器已用完分配给进程全局区的内存。

解决方案：增加可用的交换内存或真正的内存，或减少服务器上运行的进程的数量。

ORA-4031: unable to allocate <x> bytes of shared memory

服务器上的SGA用光了内存。

解决方案：执行下面的命令：ALTER SYSTEM FLUSH SHARED_POOL;如果该命令不生效，增加shared_pool_size（查看建议的值）。

ORA-3113 or ORA-3114: no longer connected to the database

可能是一个网络错误，查看服务器的警告日志以获取详细信息。

warning: Package Body created with compilation errors

输入以下文本：SQL>show errors package body <package>。

32.6.2 使用诊断和追踪

有一些事物可以被用来诊断设计器组件，以便找出产生问题的底层 SQL，或生成一个 Oracle支持需要的追踪文件。表 32-7列出了可以使用的不同级别的追踪，级别的选择依赖于所需要的信息。

表32-7 诊断错误使用的追踪级别

| 级别 | 说明与使用 |
|----|------------------------|
| 0 | 不追踪，没有诊断输出 |
| 1 | 生成SQL追踪 |
| 2 | 将SQL语句作为一个数字显示，缩小有关的代码 |
| 3 | 显示相关的SQL语句的约束变量 |
| 4 | 执行线程 |
| 5 | 使用函数自变量执行线程 |

注意 不是所有的DES2K组件都支持上述的级别，换句话说，一些级别在某些组件中是无效的。

表32-8含有oracle.ini变量，每个设计器组件都放置在 [designer (设计器)] 部分中。

表32-8 每个设计器组件的oracle.ini变量

| 组 件 | 变 量 |
|-----------------|------------------------------|
| 过程模型 | DES2_BPMOD_DIAG_LEVEL |
| 实体关系 | DES2_SYSMOD20_ERD_DIAG_LEVEL |
| 功能层次 | DES2_SYSMOD20_FHD_DIAG_LEVEL |
| 数据流图 | DES2_SYSMOD20_DFD_DIAG_LEVEL |
| 矩阵图 | DES2_REPADM10_MD_DIAG_LEVEL |
| 数据图 | DES2_SYSDS10_DD_DIAG_LEVEL |
| 表单生成器 | DES2_CGENF45_DIAG_LEVEL |
| 报表生成器 | DES2_CGENR25_DIAG_LEVEL |
| RON | DES2_REPADM10_RON_DIAG_LEVEL |
| 知识库系统管理 | DES2_REPADM10_RAU_DIAG_LEVEL |
| 知识库报表 | DES2_REPADM10_REP_DIAG_LEVEL |
| 知识库工具 | DES2_REPADM10_UTL_DIAG_LEVEL |
| Visual Basic生成器 | DES2_VBGENI0_DIAG_LEVEL |
| C++生成器 | DES2_CPPGENI0_DIAG_LEVEL |
| DB设计向导 | DES2_DATWIZ55_DIAG_LEVEL |

例如，使用数据流图 (Data Flow Diagrammer) 时，如果接收到信息 ora-1422 exact fetch returns more than requested number of rows，应该编辑你的oracle.ini文件——将级别设为4——以便再次产生这个错误。检查追踪文件，找到与错误相关的额外线索。

下面只适用于Win95/NT：

追踪RON会话——可以修改注册表，使RON会话能够将追踪信息放入一个文件。当对注册表进行修改时必须非常小心，一定要在修改之前备份注册表。使用 REGEDIT或 REGEDT32，在\HKEY_LOCAL_MACHINE\SOFTWARE\Oracle\REPADM10结点下创建下面的键值。

DES2_REPADM10_RON_TRACE_FILE——提供了追踪文件的位置与文件名。

DES2_REPADM10_RON_DIAG_LEVEL——依据想要追踪的事务，将该值设定为某个追踪级别。表32-9列出了一个完全的清单。

表32-9 RON追踪级别

| 追踪类型 | 追踪值与意义 |
|------|---------------------------|
| SQL区 | SQL追踪 SQL语句定位 SQL语句 |
| 信息 | 致命信息 非致命信息 所有信息 |
| 调度 | 调度调用 |
| 外部调用 | winExec命令 实用程序参数 |
| 通常 | 功能追踪 版本信息 |

这些追踪级别可以被累积使用。例如，如果想要追踪所有致命错误信息的 SQL追踪和SQL语句位置，使用设置 19 (1+2+16)。图32-16显示了可以被设置的一些诊断事件。

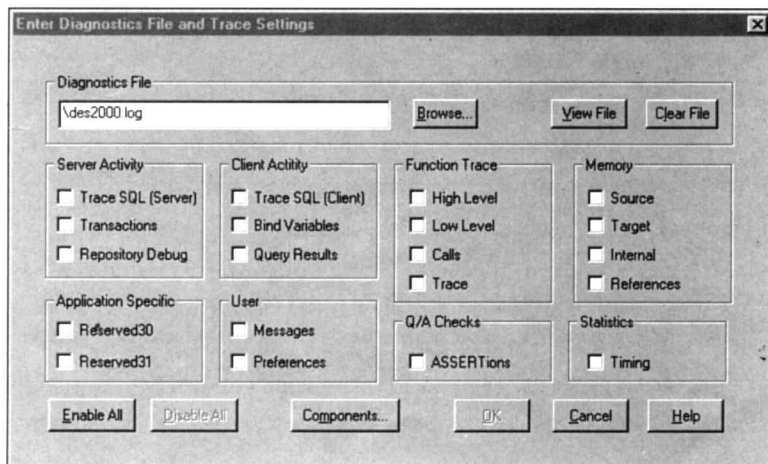


图32-16 诊断事件设置

32.6.3 从设计器有效地生成开发器应用的技巧

开发器是一族产品，可以用于开发客户服务器应用。开发器由以下主要组件组成：

Oracle表单——用于建立基于表单的应用。

Oracle报表——用于设计报表。

Oracle图表——用于建立生成商务图表的查询。

过程建立器——用于设计并管理PL/SQL代码。

表单生成器以文档中不明显的方式使用知识库。一些能够消除从设计器生成应用的困难的技巧如下面列表所示，这些技巧可以帮助系统管理员/开发者使用设计器有效地建立应用：

在正确的时间进行培训——如果人们在开始使用系统前的很长一段时间就经过培训，那么在开始使用系统时，他们会忘掉大部分所学过的东西。另一方面，如果他们在开发系统之后进行培训，他们会通过他们的自学与其他书领会事物，有可能不是使用最佳途径去完成工作。

建立用户界面原型，并征得用户的同意——这非常重要，因为在界面生成后再进行用户界面的修改会花费很大精力。原型会传达给用户这样的信息，设计者真正理解的用户需求是什么，以及用户预期从最终产品得到什么。用户界面设计可以对所有的系统设计产生影响，甚至在数据的物理布局设计上产生影响。在用户界面设计时需要考虑的事情包括但不限于以下内容：工具栏设计、颜色、字体、单窗口还是多窗口、菜单系统、屏幕布局等等。

域的使用使得对列的维护非常方便——域应该用于以下情况：

- 在多个实体中出现的属性。
- PK-FK属性。
- 可以被标准化的列定义。
- 具有一些相异可能值集的列。
- 域的使用可以简单地将变化一步传播到全部有影响的列。

理解设计的限制——不要向用户许诺一个使用当前的设计不能实现的界面，如果设计

不能反映用户所持有的最终想像，将花费大量人力与财力去解决表单后期生成。

规范化数据设计——规范化数据设计更易于维护，也更能响应要求的改变。然而，一个标准化的数据设计对用户是不直观的。例如，一个标准化的数据设计将消除数据冗余并将数据放在不同的表中，但用户希望在同一个表中看到一切事物。

有三种不同的替代方法可以解决这个问题：

- 规范化物理设计与非规范化用户显示——一个“标准”的数据显示对用户来说并不总是直观的，因此，用户需要争取一个非标准的数据显示形式。这个方法基本上总是会争取到一个标准的表单，但会导致一个高维护的系统。每当采取这个方法时，会导致生成后期的修改，如创建许多视图（非标准的显示），以协调两者之间的差别。
- 教育用户认识到标准化的使用——如果用户可以认识到数据标准化的方法，这个途径可能是最佳的。还有，用户需要更加清楚数据库特定的事物，如主-外键。他们可能会乐于理解这样的事情，也可能不愿意理解这样的事情。
- 非标准化物理设计，并将显示与用户所期望的相匹配——这一方法具有更高的生成（较低的生成后修改）的优势。然而，它对用户的需求具有较低的修改灵活性。在项目开始时，理解用户的需求非常重要，而且希望在应用开发的过程中这些要求不会改变太多。

记住，创建项目是为用户谋利的，因此，用户对项目最终结果的信心与满意对项目的成功非常重要。第一种方法通常涉及许多生成后的维护工作，因为设计没有满足用户的期望，因此，需要对表单进行修改或创建视图以调和项目设计与用户需求之间的差别。假设用户愿意接受对这种处理方式的学习，第二种方法非常有用，而且这种方法在不同的组织中是不同的。第三种方法涉及很小的生成后的管理工作，但很难将变化插入最初的需求中。这不是一个简单的选择，但如果开发组与用户不断地相互交流，这个过程会变得更具可管理性。

使用视图协调生成后的差别——通常开发者害怕使用视图，因为视图是附加的需要维护的对象，并且比直接查询表的效率要低。然而，记住视图只是存储的 SELECT 语句，可以用于生成后阶段以满足用户所需求的最小程度的改变，如新的报表或应用安全性的改变。

在创建模块之前，确保已输入“缺省”数据——在创建模块之前，应该在表定义中输入缺省数据。可以在每个表中查证下面的缺省值：显示数据类型、显示顺序、格式、提示、示意、缺省排序顺序、显示长度和显示宽度。在模块需要被重新开发时，这会节省很多时间。

显示有意义的错误信息——当用户在域中输入不合法的信息时，最好为用户提供有意义的错误信息，显示合法的值是什么，而不是给予他们毫无意义的系统生成的错误信息。

加强标准化——开发中的标准化会导致增加开发效率、减少维护工作以及减少新开发者学习时所走过的弯路。设计器提供了许多实现标准的报表，它还提供了一个 API，这个 API 由一套知识库表上的视图与一套允许知识库更新的 PL/SQL 包组成，它也可以用于产生报表，该报表列表/修复标准违反的情况。

学习表单与报表的生成优选项——在项目生成阶段，花费在理解优选项上的时间最终会通过降低生成后的维护工作得到回报。

为应用设置缺省优选项——熟悉可以在应用级设置，并且在生成的模块中保持不变的

优选项。对每个生成的模块，还有其他的优选项，该优选项根据需要可以一致于设计标准。

学习与使用表单和报表生成器——许多开发者喜欢使用设计器生成器生成一个表单或报表的框架，而使用 Developer/2000 做大部分的开发工作。然而，DES2K 生成器非常有用，如果花一些时间理解它们的能力，会产生一个高比率的生成。

使用户参加一定范围的项目设计——这对于得到用户同意原型设计有明显的帮助，与用户的不断交流也是必要的。然而，在某个特定点之后，不要允许用户再做出任何建议，否则，系统将会不断地改变，用户的期望也会越来越高。

应该在所有的模块生成之后开始生成后工作——如果生成后工作在早期阶段开始，数据设计的主要改变会难于与最终模块合并。

文档后生成步骤——使用 DES2K 知识库存储有关生成后步骤的详细描述，这样，如果需要重新应用这些步骤（通常会有这样的需要），开发者可以容易地再次运用这些步骤。可以使用模块文本域，如“注释”，或创建自己定制的文本域来存放这些信息。

定制模板——DES2K 提供了可以实现特定需要的一些定制的模板。

应该将 PL/SQL 放入库函数、过程与包中——这会简化模块的代码维护。模板代码可以访问这些库，如果在库中代码有任何修改，模板会立即反映这个修改，而不需要重新编译。

32.6.4 设计器与 Oracle 8i

Oracle8i 的引入使数据库以及在这些数据库上建立的应用达到一个全新的级别，包括在存储的数据量与支持的用户量方面都达到一个全新的级别。设计器为 Oracle8i 的关系型与对象构造提供了一套完整的建模与生成工具。

设计器 2.x 提供对所有 Oracle8i 可伸缩性功能的支持，包括如下：

分区表。

BLOB（二进制巨型对象）与 CLOB（字符型巨型对象）。

索引组织表（IOT）。

延迟约束检查。

类型表。

集合。

对象视图（虽然它是一个类型，但代表的却是一种关系型结构）。

嵌套类型。

VARRAYS（多值列）。

嵌套表（嵌套在另一个表中的表）。

引用（直接从另一个对象引用一个对象）。

设计器的模式建模已扩展到使用 Oracle8i 的可伸缩性与对象功能，而不需要损害使用的简易性。它允许你设计数据库模式或将现有的模式定义载入到知识库中，然后服务器生成器将图形表示转换为相应的 SQL DDL 语言来实现该模式。设计器使用统一的建模语言（Unified Modeling Language，UML），一个对象管理组（Object Management Group，OMG）的开放标准来显示它的类型模型。

表32-10详细列出了设计器支持的版本清单（请与 Oracle 全球支持联系以获得设计器 R2.x 的更新的清单）。

表32-10 设计器支持的版本

| 设计器2.1.2 | 组件版本 |
|--------------------|--------------------------------------|
| 平台 | Windows NT 4.0或Windows 95或Windows 98 |
| Oracle 服务器版本 | 7.3.4 or higher |
| 设计器知识库系统管理 | 2.0.20.5.0 |
| 设计器设计编辑器 | 2.0.20.5.0 |
| 设计器服务器生成器 | 7.0.20.5.0 |
| 设计器表单生成器 | 5.0.20.11.0 |
| 设计器报表生成器 | 3.0.20.6.0 |
| 设计器库生成器 | 2.0.20.5.0 |
| 设计器MS帮助生成器 | 2.0.20.5.0 |
| 设计器Visual Basic生成器 | 2.0.20.5.0 |
| 设计器Web服务器生成器 | 2.0.20.6.0 |
| Oracle安装器 | 3.3.0.1.3 |
| SQL*Plus | 8.0.4.0.0 |
| Oracle Net8产品 | 8.0.4.0.0 |
| 报表运行 | 3.0.5.8.0 |