

Oracle JDeveloper 10g概述

Oracle 白皮书
2004 年 3

引言.....	3
完整且集成.....	3
可选择开发方式.....	错误! 未定义书签。
标准、开放且可扩展.....	4
J2EE 开发与 ORACLE ADF.....	4
Oracle ADF 各层.....	5
视图层.....	5
控制器层.....	5
业务服务层.....	5
模型层.....	5
可视化和声明式 J2EE 开发.....	7
业务服务层.....	7
模型层.....	7
控制器层.....	8
视图层.....	8
完整周期支持.....	10
UML 建模与 MDA.....	10
调试和测试.....	10
代码监测和优化.....	11
部署.....	11
团队开发.....	11
远非一个 JAVA 工具.....	12
Web 服务开发.....	12
数据库开发.....	12
XML 开发.....	12
结论.....	13

Oracle JDeveloper 10g 概述

引言

Oracle JDeveloper 10g 是一个集成开发环境 (IDE)，通过它可以使使用最新的 Java、XML 和 SQL 行业标准来创建应用程序和 Web 服务。

Oracle JDeveloper 支持包括建模、编码、调试、测试、监测、调整和部署应用程序的完整的开发周期。

可视化和声明式开发方法与创新的 Oracle 应用程序开发框架 (Oracle ADF) 配合工作，简化了应用程序开发过程，减少了琐碎的编码任务，从而提供了前所未有的效率和技术选择。

完整且集成

Oracle JDeveloper 支持开发周期的每个步骤，包括建模、编码、调试、测试、监测、调整和部署应用程序。所有这些任务均从一个 IDE 中使用一组集成特性来完成。

Oracle JDeveloper 使用 J2EE、J2SE 或 J2ME 集中进行 Java 应用程序开发。此外，JDeveloper 提供了诸如 XML 模式建模器、XML 代码洞察和 XML 标记属性检查器等特性，因此它可以实现基于 XML 的应用程序开发。为了使开发人员的工具箱更加完善，Oracle JDeveloper 还为构建数据库对象和存储过程提供了完整的开发和建模环境。

Oracle JDeveloper 提供了单一、高度集成的、易用的 IDE，它具有一致的界面和开发体验。

提高开发效率

Oracle JDeveloper 10g 旨在简化 J2EE 开发。为实现这一目标，Oracle JDeveloper 专注于为 J2EE 开发提供可视化和声明式方法。进一步的简化是由 Oracle 应用程序开发框架 (Oracle ADF) 提供的，Oracle ADF 是一种采用设计模式并消除基础架构编码的 J2EE 开发框架。

Oracle JDeveloper 的独特性在于对各种技术系列使用相同的生产性开发体验。例如，开发人员可以选择使用简单的 Java 类、EJB、TopLink、Oracle ADF 业务组件或 Web 服务来实现持久层。不管选择什么技术，Oracle JDeveloper 都将为建立该层提供声明式方式，并提供拖放机制以便将用户界面组件绑定到这些实现中的任何一个。

由于意识到不同开发人员的 Java 技能不同并且由各自偏好的开发方法，因此，Oracle JDeveloper 提供了不同的开发方法选项，包括模型驱动的体系结构 (MDA)、声明式开发和手工编码。开发人员可以选择最适合其个人开发风格的方法。

使用 JDeveloper 开发的应用程序可以使用任何数据源，并且可以部署在任何与

J2EE 兼容的应用程序服务器上。

Oracle JDeveloper 是百分之百基于 Java 的工具，它是跨平台的 IDE，可以运行在 Windows、Linux 和各种基于 Unix 的系统上，因此，它允许开发人员选择自己的开发平台。

标准、开放且可扩展

Oracle JDeveloper 使开发人员能够使用最新行业标准来开发应用程序，所开发的应用程序可以在多种硬件和软件平台上运行。使用 Oracle JDeveloper 创建的应用程序可以部署在任何与符合 J2EE 规范的服务器上，并且可以访问任何与符合 JDBC 标准的数据库。

Oracle JDeveloper 接受各种流行的开放源代码框架和工具，它为 Struts、Ant、JUnit 和 CVS 提供了相应的内置特性。该集成使开发人员能够使用这些开放源代码工具简化他们的开发过程。

Oracle JDeveloper 提供了扩展 SDK，它允许开发人员添加功能和定制开发环境。Oracle JDeveloper 由核心 IDE 平台及其上的一组扩展构成。开发人员可以根据需要打开或关闭各扩展，以便按需定制 IDE。JDeveloper 小组用来开发产品的 API 也可提供给对集成 Oracle JDeveloper 和增强其功能感兴趣的开发人员和第三方公司使用。

J2EE 开发与 ORACLE ADF

J2EE 是使用 Java 语言构建多层应用程序的一组规范。J2EE 是一个强健、可伸缩和安全的平台，它构成了当今很多企业应用程序的基础。

经过多年的发展，针对 J2EE 平台的最佳实践和设计模式已经形成。但问题是，实现这些最佳实践通常需要编写大量的基础架构代码。Oracle JDeveloper 10g 旨在解决这个问题。

Oracle JDeveloper 10g 包括 Oracle 应用程序开发框架 (Oracle ADF)。这个框架通过尽量减少为实现设计模式和应用程序基础架构而编写代码的需要，从而简化了 J2EE 开发过程。Oracle ADF 将这些代码作为框架的一部分提供给开发人员。Oracle ADF 的突出特点是运行时服务和开发特性。

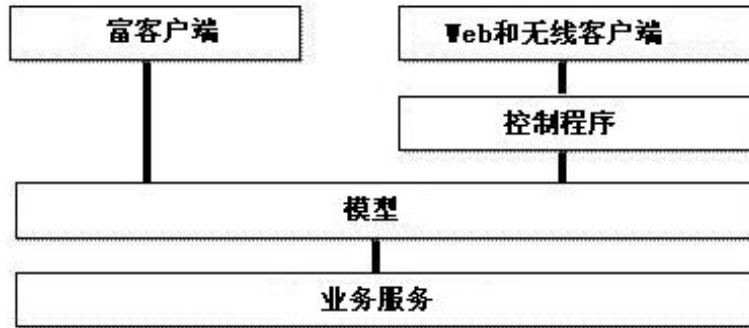
Oracle ADF 是对先前的 JDeveloper 版本所附带的框架的发展、提高和扩展。

Oracle ADF 基于模型-视图-控制器 (MVC) 设计模式。MVC 将应用程序体系结构划分为三层：

- 模型 — 处理与数据源的交互并运行业务逻辑
- 视图 — 处理应用程序用户界面
- 控制器 — 处理应用程序流，并充当模型层和视图层之间的接口

每个层彼此间的独立性导致了一个松散耦合的体系结构。通过让应用程序使用松散耦合的体系结构，简化了维护并增加了代码的重用性。Oracle ADF 为实现 MVC 体系结构提供了非常简便的方式。

Oracle ADF 层



Oracle ADF，高级的体系结构

Oracle ADF 基于四个层：

- 业务服务层 — 提供对各种数据源中数据的访问，处理业务逻辑。
- 模型层 — 在业务服务层之上提供一个抽象层，使视图层和控制器层能够以一致的方式使用业务服务的不同实现。
- 控制器层 — 为控制 Web 应用程序流提供一种机制。
- 视图层 — 提供应用程序的用户界面。

Oracle ADF 让开发人员能够在实现每个层时选择他们喜欢使用的技术。不管采用哪种技术，Oracle ADF 提供同样的可视开发和声明开发体验。

视图层

视图层提供应用程序用户界面。视图层使用 HTML、富 Java 组件或 XML 并且以不同的方式呈现用户界面。视图层可以是基于 Web、客户端服务器甚至是无线工具。

控制器层

控制器层用于控制应用程序流。基于 Web 的应用程序由多个包含动态内容的 Web 页面组成。控制器层管理这些页面之间的流。创建该层时可以使用不同的模型。基于 Java 的 Web 应用程序的最重要的体系结构依赖于充当控制器的一个 servlet。Apache Jakarta Struts 控制器（一个开放源代码框架控制器）是基于 Java 的 Web 系统的事实上的标准控制器。Oracle ADF 使用 Struts 控制器来管理 Web 应用程序流。

业务服务层

业务服务层用于管理与数据持久层的交互。它提供诸如数据持久性、对象/关系映射、事务管理和业务逻辑执行这样的服务。

可以将 Oracle ADF 中的业务服务层实现为简单的 Java 类、EJB、Web 服务、TopLink 对象或 Oracle ADF 业务组件。

模型层

模型层用于将业务服务连接到在其他层中使用它们的对象。Oracle ADF 提供位于业务服务之上的模型层实现，该实现提供可用来访问任何类型业务服务的单一接口。在将任何类型的业务服务层实现绑定到视图层和控制器层时，开发人员将获得相同的开发体验。Oracle ADF 中的模型层充当了 JSR-227 “用于 J2EE 的标准

数据绑定和数据访问设备”的基础。

Oracle ADF 中模型层的独特实现将面向服务的体系结构 (SOA) 的强大功能带给您的应用程序。模型层使您轻松地将应用程序作为可以在不同应用程序中使用和重用的一组服务来进行开发。模型层所提供的抽象让开发人员能够选择偏爱的方法来实现这些服务。

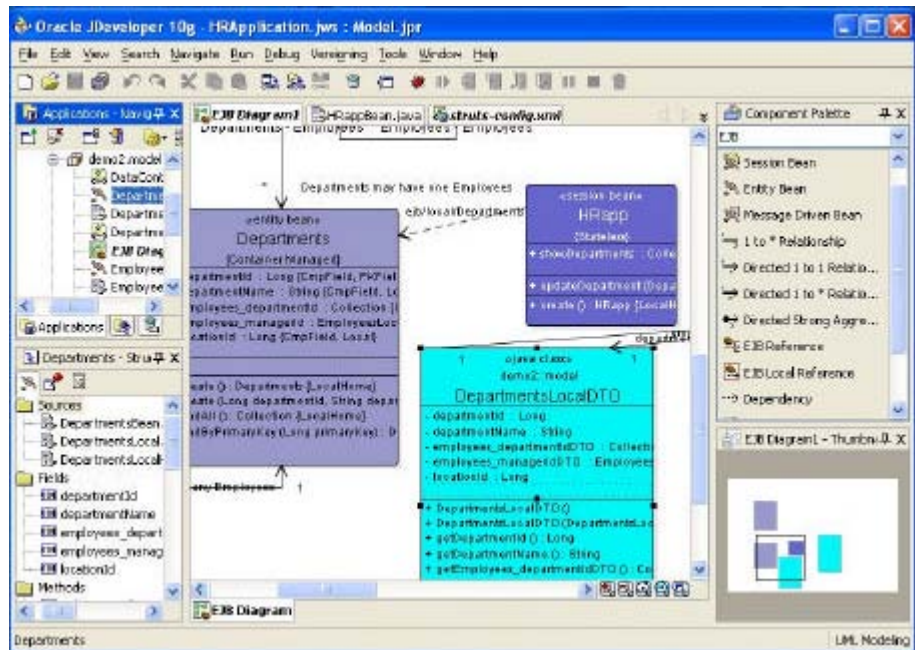
可视化和声明式 J2EE 开发

Oracle JDeveloper 通过为 Oracle ADF 的每一层提供可视化和声明式工具，简化了 J2EE 开发。这些工具给 Java 开发人员带来很多好处，即使他们不使用 Oracle ADF 的运行时特性。

业务服务层

Oracle JDeveloper 包括 UML 类图表，该图表用于建模和生成 EJB、Web 服务、TopLink 对象、简单 Java 类和 Oracle ADF 业务组件。开发人员可以将表从数据库浏览器拖放到图表上，以便生成用于为这些表提供 Java 接口的业务服务。

使用 UML 进行 EJB 建模



Oracle JDeveloper 提供的声明式接口简化了 EJB 的创建，最大程度减少了实现 EJB 接口所需的代码。单击按钮即可生成 EJB 设计模式的实现。

Oracle JDeveloper 集成的 TopLink 映射器可以帮助开发人员使用功能强大的 TopLink 持久层直观地将 Java 对象映射到数据库表。

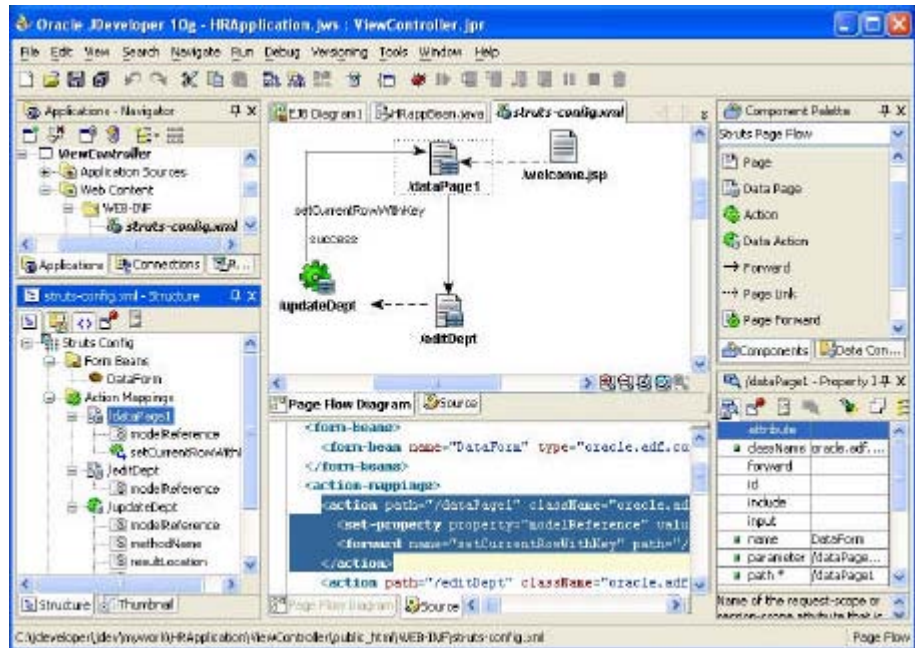
Oracle ADF 业务组件是一个专门用于以更好的声明式方式创建对象（这些对象在数据库上实现业务服务层）的框架。它提供了随取随用的服务，例如事务管理、资源池、锁定、声明式验证规则、转换和对象关系映射。Oracle ADF 业务组件使开发人员在创建访问其数据库的 Java 对象时能够利用 SQL 的完整功能。框架中内置的通用 J2EE 设计模式的实现确保了应用程序的性能和可伸缩性。

模型层

Oracle JDeveloper 使用创新的模型层实现，使开发人员轻松地将业务服务层中的组件绑定到控制器层和视图层。开发人员可以简单地拖放数据对象即可将它们绑定到它们的用户接口实现。相同的机制使开发人员能够很容易地将控制器操作绑定到业务服务层中定义的方法。

控制器层

Oracle JDeveloper 为开放源代码的 Apache Jakarta Struts 框架提供了一个页面流建模器。该建模器提供了可视界面以简化应用程序流的开发。开发人员只需简单地将 Struts 组件拖放到图表上，即可为他们的页面流建模。图表将自动与 struts-config.xml 文件中的源代码同步。使用数据控制面板，开发人员可以将业务服务方法与 Struts 操作关联起来。Struts 页面流图表为访问 Web 页的编辑功能和 Struts 操作的编码功能提供了下钻支持。



Struts 页面流建模器

视图层

Oracle JDeveloper 为基于 HTML 和 Swing 的用户界面都提供了可视的布局编辑器。开发人员使用组件面板即可将可视化组件添加到应用程序用户界面。使用任何标准 JavaBean 或 JSP 标记库即可扩展组件面板。开发人员可以使用属性检查器以声明式方式定义可视化组件的属性。可视化布局编辑器随时与源代码同步，因此开发人员可以选择其偏爱的编辑模式。

“数据控制面板”窗口提供了用于洞察业务服务层的视图。开发人员只需简单地从该面板进行拖放即可将用户界面组件绑定到业务服务。

对于基于 Web 的应用程序，Oracle ADF 还包括 ADF UIX。ADF UIX 是一组组件，开发人员可以使用这些组件以声明式方式定义包含丰富功能的高级 HTML 用户界面。

完整周期支持

应用程序的开发周期不止是编码，而是包含更多阶段。Oracle JDeveloper 支持所有开发阶段，包括建模、调试、测试、监测、优化和部署应用程序。

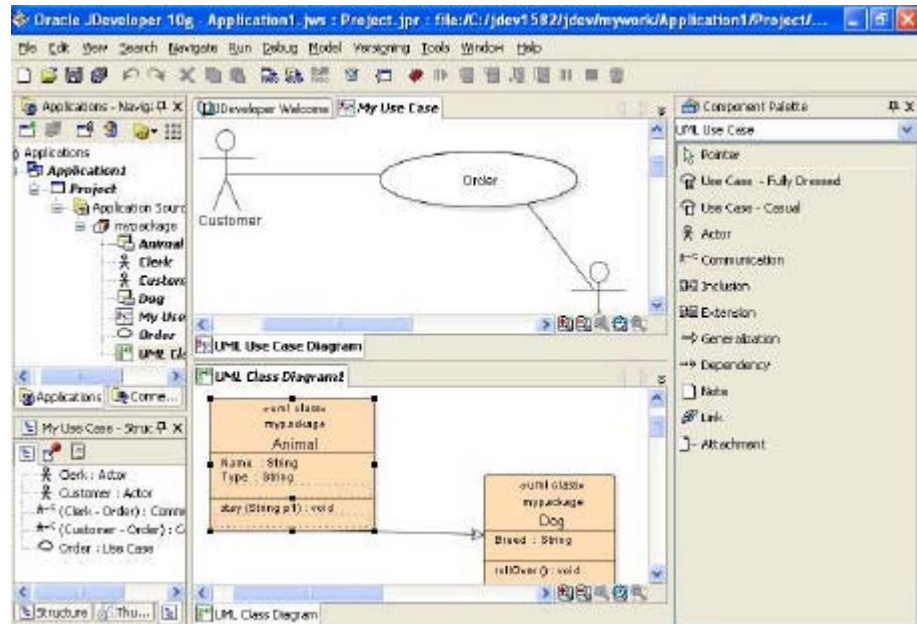
UML 建模与 MDA

Oracle JDeveloper 为进行应用程序分析和设计提供了内置的 UML 建模工具。JDeveloper 中包括类建模器、使用用例建模器和活动建模器。这些建模器用于对应用程序进行概念性建模和文档描述。

Oracle JDeveloper 中的其他建模功能和工具则为建模和生成代码以及对应用程序执行反向工程提供了可视化方式。它们包括：

- 类建模器配置文件，用于对 Java 类、EJB 和 Web 服务进行建模、生成和反向工程
- Struts 页面流建模器，用于对 struts-config.xml 文件进行可视化设计
- 数据库建模器，用于生成和捕获数据库模式的结构

Oracle JDeveloper 中的 UML 建模



调试和测试

调试代码是应用程序开发周期中的重要阶段。Oracle JDeveloper 中包括快速和强大的调试器，它为检查代码提供了可视化方式。JDeveloper 调试器支持热切换 (hot-swap) 调试，开发人员可以在调试会话过程中更改他们的代码。在调试时，开发人员可以设置断点以及向前和向后移动到源代码中的任何一点。调试器还提供了用于查看内存堆栈内部情况的视图。

对于基于 Swing 的应用程序的开发人员，JDeveloper 提供了独特的 UI 调试器。UI 调试器让开发人员很容易使用 UI 快照、事件跟踪和图形对象层次结构显示图来监视用户界面的执行情况。

对于使用 PL/SQL 的数据库开发人员，Oracle JDeveloper 提供了在数据库内部调试 PL/SQL 代码的功能。

Oracle JDeveloper 同时提供本地和远程调试功能，因此开发人员可以在代码正在远

程 J2EE 容器上执行时对其进行检查。对于 J2EE 应用程序，开发人员可以使用 Oracle JDeveloper 附带的内置 J2EE 容器来测试他们的 JSP、Servlet 和 EJB，而不需要安装独立版的应用程序服务器。

Oracle JDeveloper 集成了开放源代码 JUnit 测试框架。通过使用向导，可以很容易地为项目定义测试用例、测试周期和测试套件。可以在 IDE 中执行测试，以验证代码的功能。

代码监测和优化

Oracle JDeveloper 具备帮助开发人员定位和修复性能瓶颈和内存瓶颈的特性。

事件、执行和内存监测器的组合使开发人员能够监视应用程序执行，以定位需要关注的代码区域。

JDeveloper 中创新性的 CodeCoach 特性充当了个人 Java 导师，该特性扫描应用程序代码，并为如何更改以优化性能提供提示和建议。CodeCoach 甚至可以自动修复某些问题。

代码审计功能帮助组织实施编码标准。JDeveloper 的审计特性可以分析 Java 代码并检查代码是否符合编码标准。审计工具允许开发人员查找和修复 Java 源文件中的违规情况。可以向 JDeveloper 所附带的审计规则 and 规定中添加新的审计规则 and 规定。

代码量度使开发人员很容易通过检测 Java 代码的复杂性来评估它的结构。JDeveloper 提供的代码量度特性可以分析继承树的深度、语句数以及代码的圈复杂度。

部署

Oracle JDeveloper 简化了将 J2EE 应用程序部署到 J2EE 服务器的过程。它通过对话框为建立应用程序的部署描述符提供了声明式方式。EJB 校验器将校验代码的结构，并消除部署过程中的错误。打包向导将从项目创建标准的 WAR、EAR 和 Jar 文件。这些标准文件可以被部署到任何符合 J2EE 的应用程序服务器上。对于 Oracle 应用服务器、BEA Weblogic、JBoss 和 Tomcat，直接从 IDE 单击一下即可部署到 J2EE 容器。

对于基于 Swing 的应用程序，JDeveloper 提供了相应的向导，该向导可以将应用程序作为 Java Web-Start 应用程序打包，从而简化了客户机机器上的部署。

Oracle JDeveloper 在 IDE 中集成了开放源代码 Ant 构建工具。向导将帮助开发人员定义和运行用于管理构建任务的 Ant 脚本。

团队开发

Oracle JDeveloper 集成了软件配置工具来管理代码生命周期和实现团队开发。

JDeveloper 提供与 Oracle SCM、开放源代码 CVS 和 Rational ClearCase 现成的集成。开发人员使用向导和菜单选项就能从 IDE 内部调用这些工具。开发人员可以从库添加和删除文件、管理版本历史、签入和签出、锁定和比较文件版本。

Oracle JDeveloper 还为通过 HTTP 协议共享文件提供了 WebDAV 支持。可以使用 SCM 扩展 API 将其他软件配置工具集成到 IDE 中。

远非一个 JAVA 工具

Java 不是开发人员工具箱中的唯一工具。可以用其他语言和技术补充 Java，帮助开发人员交付完整的应用程序。开发人员可以扩展 Oracle JDeveloper 的功能以支持超越纯 Java 的开发领域。

Web 服务开发

Web 服务用于集成应用程序和跨越开发语言界限。使用基于 XML 的行业标准（例如，WSDL、SOAP 和 UDDI），不管代码的位置或在开发过程中所使用的语言如何，都可以重用代码组件。

Oracle JDeveloper 为开发新的 Web 服务和使用现有的 Web 服务提供了完整支持。

Oracle JDeveloper 可以生成将任何 Java 类或 PL/SQL 包暴露为 Web 服务所需的 WSDL 文件。JDeveloper 对 UDDI 的支持包括：将 Web 服务部署到 UDDI 库中、UDDI 浏览器以及能够生成用于激活 Web 服务的代码段。

SOAP 监视器可以监视来自和发往 SOAP 服务器的传入和外发的 SOAP 消息。

Oracle JDeveloper 还支持 Web 服务互操作性标准，并提供验证 Web 服务是否符合 WS-I 标准的功能。

可以在 Oracle ADF 中将 Web 服务用作业务服务层的数据源。Oracle ADF 简化了与 Web 服务交互的用户界面的创建工作。

数据库开发

大多数企业应用程序都依赖数据库来持久保存它们的数据，并将存储过程和函数中的代码放在数据库层。Oracle JDeveloper 为数据库开发人员提供了完整的环境。数据库建模器让开发人员能够设计数据库的结构，包括表、列和关系。它还提供了对现有数据库模式执行反向工程的功能。向导帮助开发人员创建数据库对象，如表、视图和代码组件。数据库浏览器提供了数据库组件的方便视图，而表查看器则显示任何表的结构和数据。

内置的 SQL 工作表让开发人员可以执行任何 SQL 命令，并显示它的解释计划，从而帮助优化数据库访问。

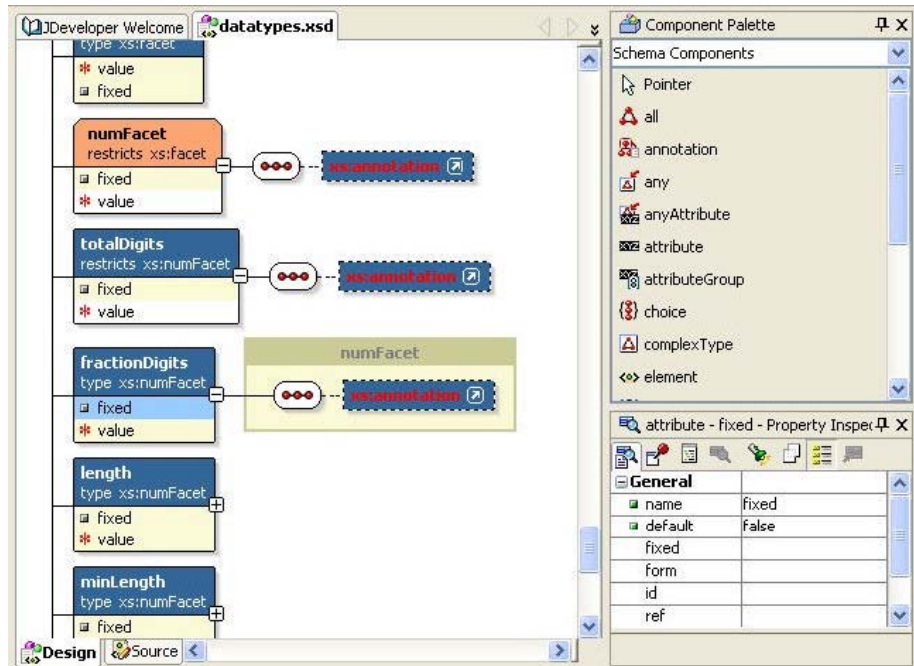
代码编辑器能够解析 PL/SQL 语法，并提供代码洞察和彩色编码，帮助开发人员更快地编写 PL/SQL 代码。开发人员还可以在 JDeveloper 中运行和调试他们的 PL/SQL 代码。

XML 开发

对于 XML 开发人员，Oracle JDeveloper 提供了可视化 XML 模式编辑器，它允许开发人员方便地浏览 XML 模式。可以从组件面板通过拖放来直观地构造 XML 模式。

能够解析 XML 的编辑器可以在创建 XML 文件时提供代码洞察功能，并且属性面板为以声明的方式定义标记属性提供了简单方法。

Oracle JDeveloper 包括 Oracle XML 开发人员套件 (Oracle XDK)，该套件所提供的库可以用来建立基于 XML 的应用程序。



XML 模式建模

结论

为了解决软件开发人员所面临的日益增长的需求，Oracle JDeveloper 10g 提供了集成的完整开发环境，它所具有的特性能够极大地提高生产力。Oracle JDeveloper 专注于 Java、SQL 和 Web 服务开发以及完整开发周期，它是开发新手和老手均可使用的功能强大的工具。



Oracle JDeveloper 10g 概述

2004 年 3 月

作者: Shay Shmeltzer 合作作者:

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

版权所有 © 2004, Oracle。保留所有权利。本文档只用于提供信息, 其中的内容如有更改, 恕不通知。本文档不保证没有错误, 也不受其它任何口头表达或法律暗示的担保或条件的约束, 包括对特定用途的适销性或适用性的暗示担保和条件。我们特别声明: 拒绝承担与本文档有关的任何责任, 本文档不直接或间接形成任何合约职责。未经预先书面许可, 不允许以任何形式或任何方式(电子方式或机械方式)、出于任何目, 复制或传播本文档。Oracle 是 Oracle Corporation 和/或其子公司的注册商标。其他名称可能是其各自所有者的商标。