Oracle JDeveloper 10g Overview

An Oracle White Paper
March 2004

ORACLE

Oracle JDeveloper 10g Overview

INTRODUCTION

Oracle JDeveloper 10*g* is an integrated development environment (IDE) for building applications and Web services using the latest industry standards for Java, XML, and SQL.

Oracle JDeveloper supports the complete development life cycle with integrated features for modeling, coding, debugging, testing, profiling, tuning, and deploying applications.

A visual and declarative development approach and the innovative Oracle Application Development Framework (Oracle ADF) work together to simplify application development and reduce mundane coding tasks, offering unparalleled productivity and a choice of technology stacks.

Complete and Integrated

Oracle JDeveloper supports every step of the development life cycle, including modeling, coding, debugging, testing, profiling, tuning, and deploying applications. All these tasks are done from a single IDE using a set of integrated features.

Oracle JDeveloper focuses on Java application development using J2EE, J2SE, or J2ME. In addition, JDeveloper enables XML-based application development with features such as the XML Schema Modeler, XML code insight, and the XML tag property inspector. To complete the developer's toolbox, Oracle JDeveloper also provides a full development and modeling environment for building database objects and stored procedures.

Oracle JDeveloper provides a single, highly integrated, developer-friendly IDE, with a consistent interface and development experience.

Productivity with Choice

The goal of Oracle JDeveloper 10g is to make J2EE development simpler and more accessible. To achieve this goal, Oracle JDeveloper focuses on a visual and declarative approach to J2EE development. Further simplification is provided by the Oracle Application Development Framework (Oracle ADF) - a J2EE development framework that implements design patterns and eliminates infrastructure coding.

A unique aspect of Oracle JDeveloper is that the same productive development experience is used for various technology stacks. For example, developers can choose to implement a persistence layer using simple Java classes, EJB, TopLink, Oracle ADF Business Components, or Web services. Regardless of the chosen technology, Oracle JDeveloper will provide a declarative way to create this layer, as well as drag-and-drop mechanisms to bind user interface components to any of these implementations.

Realizing that different developers have a different Java skill level and their own preferred development approach, Oracle JDeveloper provides a choice of development approaches that includes Model Driven Architecture (MDA), declarative development, and hand-coding. Developers can choose the approach that best suits their personal development style.

Applications developed with JDeveloper work with any data source and can be deployed on any J2EE-compatible application server.

Oracle JDeveloper, a 100% Java-based tool, is a cross-platform IDE that runs on Windows, Linux, and various Unix-based systems, letting developers choose their development platform.

Standard, Open, and Extensible

Oracle JDeveloper enables developers to use the latest industry standards to develop applications that can operate across multiple hardware and software platforms. Applications built with Oracle JDeveloper can be deployed to any J2EE-compliant server and can access any JDBC-compliant database.

Oracle JDeveloper embraces popular open source frameworks and tools, providing built-in features for Struts, Ant, JUnit, and CVS. This integration enables developers to use these open source tools to streamline their development process.

Oracle JDeveloper offers an Extension SDK that lets developers add capabilities and customize the development environment. Oracle JDeveloper is built as a set of extensions on top of a core IDE platform. Developers can turn extensions on or off as they wish, customizing the IDE for their needs. The same API that is used by the JDeveloper team to develop the product is available to developers and third party companies who are interested in integrating with and enhancing Oracle JDeveloper.

J2EE DEVELOPMENT AND ORACLE ADF

J2EE is a set of specifications for building multi tier applications using the Java language. J2EE is a robust, scalable, and secure platform that forms the basis for many of today's enterprise applications.

Over the years, best practices and design patterns have evolved for the J2EE platform. The problem is that implementing these best practices usually involves writing a lot of infrastructure code. Oracle JDeveloper 10g aims to solve this problem.

Oracle JDeveloper 10g includes the Oracle Application Development Framework (Oracle ADF). This framework simplifies J2EE development by minimizing the need to write code that implements design patterns and application's infrastructure. Oracle ADF provides them as part of the framework. Oracle ADF features both runtime services and development features.

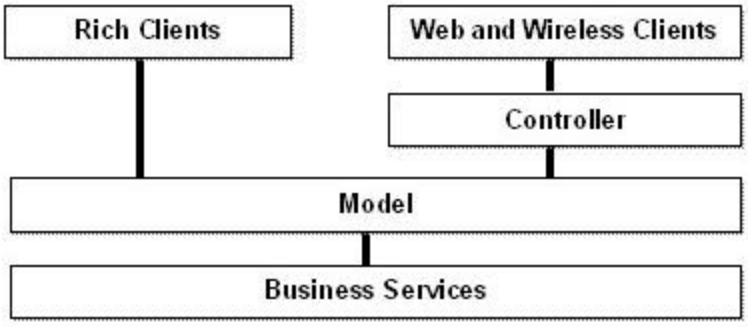
Oracle ADF is an evolution, an improvement, and an extension of frameworks that were included with previous versions of JDeveloper.

Oracle ADF is based on the Model-View-Controller (MVC) design pattern. MVC separates the application architecture into three layers:

- Model - handles interaction with data-sources and runs the business logic
- View - handles the application user interface
- Controller - handles the application flow and acts as the interface between the Model and the View layers

The independence of each layer from the others results in a loosely-coupled architecture. Using a loosely-coupled architecture for applications simplifies maintaining and increases code-reusing. Oracle ADF provides an easy way to implement the MVC architecture.

The Oracle ADF Layers



Oracle ADF a high level architecture

Oracle ADF is based on four layers:

- The Business Services layer - provides access to data from various sources and handles business logic.
- The Model layer - provides an abstraction layer on top of the Business Services layer, enabling the View and Controller layers to work with different implementations of Business Services in a consistent way.
- The Controller layer - provides a mechanism to control the flow of the Web application.
- The View layer - provides the user interface to the application.

Oracle ADF lets developers choose the technology they prefer to use when implementing each of the layers. Oracle ADF provides the same visual and declarative development experience regardless of the technology stack used.

The View Layer

The View layer provides the user interface to the application. The view layer uses HTML, rich Java components, or XML and its variations to render the user interface. The View layer can be Web-based, client server-based, or even a wireless implementation.

The Controller Layer

The Controller layer controls the application's flow. Web-based applications are composed of multiple Web pages with dynamic content. The controller layer manages the flow between these pages. Different models can be used when building this layer. The most prominent architecture for Java-based Web applications relies on a servlet that acts as the controller. The Apache Jakarta Struts controller, an open source framework controller, is the de facto standard

controller for Java-based Web systems. Oracle ADF uses the Struts controller to manage the flow of Web applications.

The Business Services Layer

The Business Services layer manages interaction with a data persistence layer. It provides such services as data persistence, object/relational mapping, transaction management, and business logic execution.

The Business Services layer in Oracle ADF can be implemented as simple Java classes, EJB, Web services, TopLink objects, or Oracle ADF Business Components.

The Model Layer

The Model layer connects the Business Services to the objects that use them in the other layers. Oracle ADF provides a Model layer implementation that sits on top of Business Services, providing a single interface that can be used to access any type of Business Service. Developers get the same development experience when binding any type of Business Service layer implementation to the View and Controller layers. The Model layer in Oracle ADF served as the basis for JSR-227 “A Standard Data Binding & Data Access Facility for J2EE”.

The Model layer’s unique implementation in Oracle ADF brings the power of Service Oriented Architecture (SOA) to your application. The Model layer makes it easy to develop the application as a set of services that can be used and reused in different applications. The abstraction provided by the Model layer lets developers choose their preferred method of implementing these services.

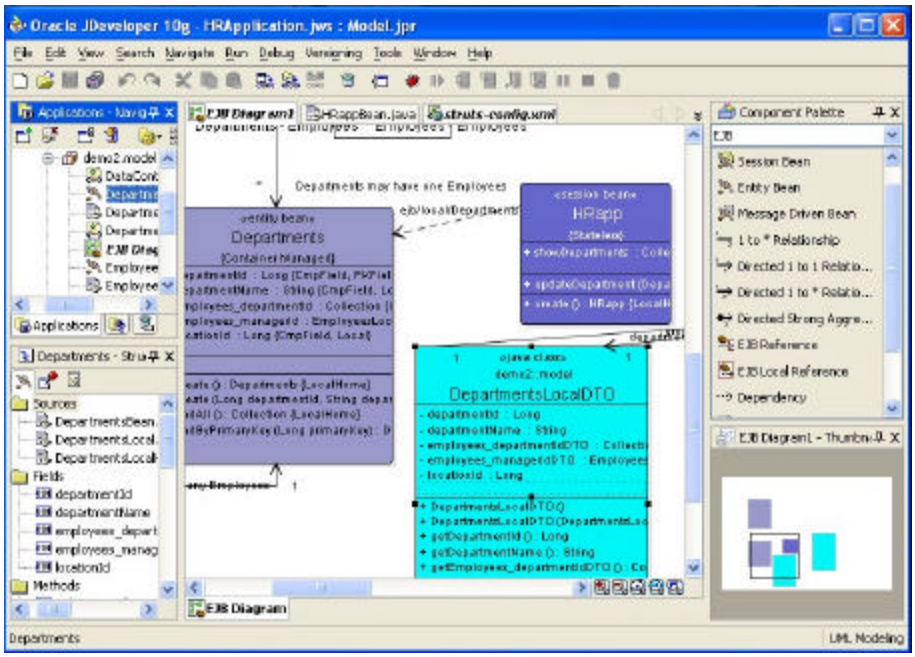
VISUAL AND DECLARATIVE J2EE DEVELOPMENT

Oracle JDeveloper simplifies J2EE development by offering visual and declarative tools for each layer of Oracle ADF. These tools benefit Java developers even if they don't use the runtime features of Oracle ADF.

The Business Services layer

Oracle JDeveloper includes a UML class diagram that models and generates EJB, Web Services, TopLink objects, Simple Java classes, and Oracle ADF Business Components. Developers can drag-and-drop tables from the database browser onto the diagram to generate Business Services that provide Java interfaces to these tables.

EJB Modeling using UML



Oracle JDeveloper provides declarative interfaces that simplify the creation of EJB minimizing the code needed to implement EJB interfaces. EJB design pattern implementations can be generated with a click of a button.

Oracle JDeveloper integrates the TopLink Mapper to help developers visually map Java objects to database tables using the powerful TopLink persistence layer.

Oracle ADF Business Components is a framework focused on creating objects, which implement the Business Services layer on top of a database, in a more declarative way. It provides out-of-the-box services such as transaction management, resource pooling, locking, declarative validation rules, translation, and object-relational mapping. Oracle ADF Business Components let developers

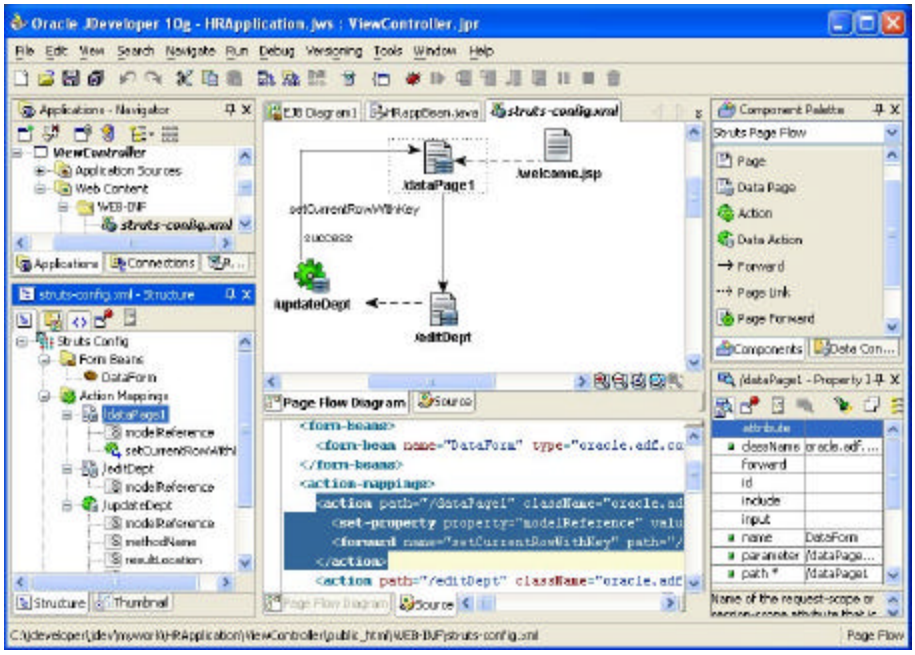
leverage the full power of SQL when building Java objects that access their databases. With built-in implementation of common J2EE design patterns in the framework, the performance and scalability of the application is assured.

The Model Layer

Oracle JDeveloper provides a very easy way to bind components from the Business Services layer to your Controller and View layers using an innovative implementation of the Model layer. Developers can simply drag-and-drop data objects and bind them to their user interface implementation. The same mechanism enables an easy binding of controller actions to methods defined in the Business Services layer.

The Controller Layer

Oracle JDeveloper provides a page flow modeler for the open-source Apache Jakarta Struts framework. This modeler provides a visual interface that simplifies the development of the application flow. Developers model their page flows using simple drag and drop of Struts components onto the diagram. The diagram is automatically synchronized with the source in the struts-config.xml file. Using the Data Control Palette, developers can associate Business Services methods with Struts actions. The Struts page flow diagram provides drill-down access to Web page editing and Struts action coding.



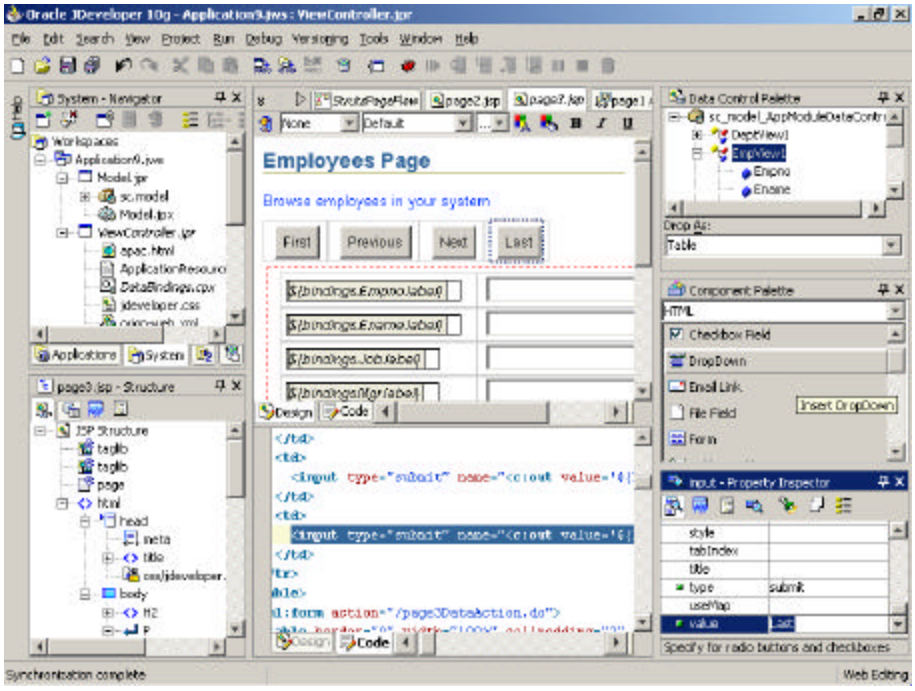
Struts Page Flow Modeler

The View layer

Oracle JDeveloper provides a visual layout editor for both HTML and Swing based user interfaces. Developers use the component palette to add visual components to the application's user interface. The component palette can be extended with any standard JavaBean or JSP Tag Library. Developers can use the Property Inspector to declaratively define the attributes of visual components. The visual layout editor is synchronized with the source code at all times, so developers can choose their preferred editing mode.

The Data Control Palette window provides a view into the Business Services layer. Developer can bind user interface components to Business Service with a simple drag-and-drop from this palette.

For Web based applications Oracle ADF also includes ADF UIX – a set of components that developers can use to declaratively define advanced HTML user interfaces with rich functionality.



Visual JSP editing

FULL LIFE CYCLE SUPPORT

The development life cycle of an application has more stages then just coding. Oracle JDeveloper supports all the development stages including modeling, debugging, testing, profiling, optimizing, and deploying applications.

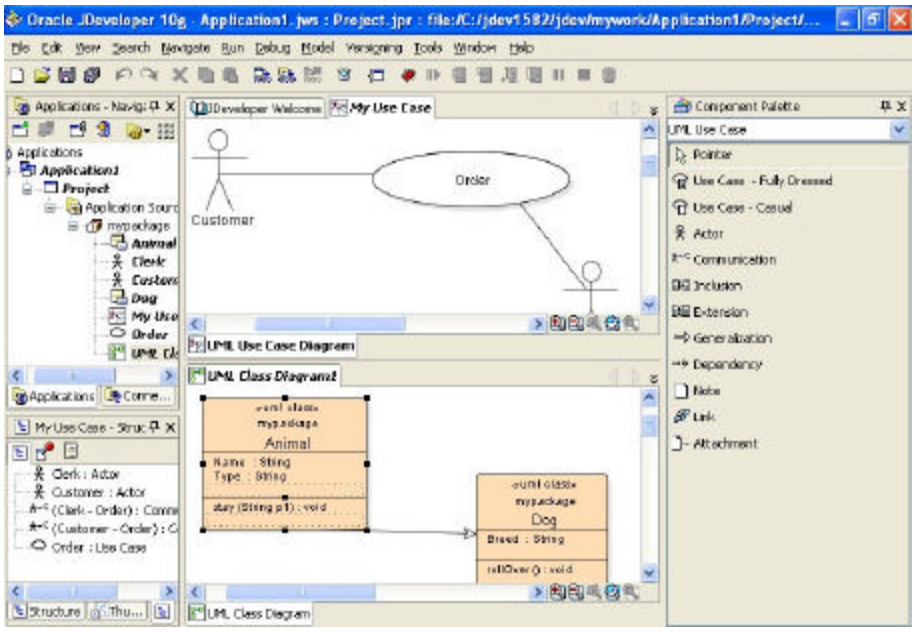
UML Modeling and MDA

Oracle JDeveloper provides built-in UML modeling tools for application analysis and design. JDeveloper includes a Class Modeler, a Use Case Modeler, and an Activity Modeler. These modelers are used for conceptual modeling and documentation of the application.

Additional modeling capabilities and tools in Oracle JDeveloper provide a visual way to model and generate code as well as reverse engineer applications. These include:

- Class modeler profiles for modeling, generating and reverse engineering of Java classes, EJB, and Web services
- Struts Page flow modeler for visual design of the struts-config.xml file
- Database modeler for generating and capturing the structure of database schemas

UML modeling in Oracle JDeveloper



Debugging and Testing

Debugging code is an essential part of the application development life cycle. Oracle JDeveloper includes a fast and powerful debugger that provides a visual way to examine code. The JDeveloper debugger supports hot-swap debugging, so developers can change their code during their debug session. Developers can set breakpoints and move forward and backward to any point in the source code while debugging. The debugger also provides a view into the memory stack.

For developers of Swing based applications JDeveloper offers a unique UI debugger. The UI debugger provides an easy way to monitor user interface execution with UI snapshots, event tracking, and graphical object hierarchy display.

For database developers using PL/SQL, Oracle JDeveloper includes the capability to debug PL/SQL code inside the database.

Oracle JDeveloper provides both local and remote debugging, so developers can examine code as it is executed on remote J2EE containers. For J2EE applications developer can use the built-in J2EE container that comes with Oracle JDeveloper to test their JSP, Servlets, and EJB without the need for an installation of a stand-alone application server.

Oracle JDeveloper integrates with the open source JUnit testing framework. Wizards provide an easy way to define test cases, test fixtures, and test suites for projects. Tests can be executed from within the IDE to validate code functionality.

Code Profiling and Optimization

Oracle JDeveloper includes features that help developers locate and fix performance and memory bottlenecks.

A combination of the event, execution, and memory profilers enables developers to monitor applications execution to locate code areas that need attention.

Acting as a personal Java guru, the innovative CodeCoach feature in JDeveloper scans the application code and provides hints and tips on changes that can be made to optimize performance. CodeCoach can even fix some of the issues automatically.

Code Audit functionality helps organizations implement coding standards. JDeveloper's auditing feature analyzes Java code and checks for adherence to coding standards. The auditing tool allows developers to find and fix rule violations in Java source files. New auditing rules and regulations can be added to those supplied with JDeveloper.

Code Metrics provide an easy way to evaluate the structure of the Java code by analyzing its complexity. JDeveloper provides code metrics features that analyze

the depth of the inheritance tree, the number of statements, and the cyclomatic complexity of the code.

Deployment

Oracle JDeveloper simplifies the deployment of J2EE applications to J2EE servers. Dialogs provide a declarative way of creating deployment descriptor for applications. The EJB verifier verifies the structure of the code and eliminates errors in the deployment process. Packaging wizards create standard WAR, EAR, and JAR files from the project. These standard files can be deployed to any J2EE compliant application server. One click deploy directly from the IDE into a J2EE container is supported for Oracle Application Server, BEA's Weblogic, JBoss, and Tomcat.

For Swing based applications, JDeveloper provides wizards that package the application as a Java Web-Start application simplifying deployment on client machines.

Oracle JDeveloper integrates the open source Ant build tool into the IDE. Wizards help developers define and run Ant scripts that manage build tasks.

Team Development

Oracle JDeveloper integrates with software configuration tools to manage code life cycle and to enable team development.

Out of the box, JDeveloper integrates with Oracle SCM, the open source CVS, and Rational ClearCase. Wizards and menu options let developers invoke these tools from inside the IDE. Developers can add and remove files from the repository, manage versions history, check in and out, lock, and compare file versions.

Oracle JDeveloper also offers WebDAV support for sharing files over the HTTP protocol. An SCM Extension API can be used to integrate other software configuration tools into the IDE.

MORE THAN JUST A JAVA TOOL

Java is not the only tool in the developer's toolbox. Other languages and technologies complement Java and help the developer deliver a complete application. Oracle JDeveloper includes features that extend its functionality to support development areas beyond pure Java.

Web Services Development

Web services are used for integrating applications and crossing the boundaries between development languages. Using XML-based industry standards such as WSDL, SOAP and UDDI, code components can be reused regardless of their location or the language used in their development.

Oracle JDeveloper provides full support for developing new Web services and consuming existing Web services.

Oracle JDeveloper generates the necessary WSDL file to expose any Java class or PL/SQL package as a Web service. JDeveloper's Support for UDDI includes the deployment of Web services into UDDI repositories, a UDDI browser, and the ability to generate code stubs that activate Web services.

The SOAP monitor monitors the incoming and outgoing SOAP messages from and to the SOAP server.

Oracle JDeveloper also supports the Web Services Interoperability standards, and provides the capability to verify that Web Services confirm to the WS-I standard.

Web services can be used in Oracle ADF as a data source for the Business Services layer. Oracle ADF makes it easy to build user interfaces that interact with Web services.

Database Development

Most enterprise applications rely on a database to persist their data and include code in stored procedures and functions in the database tier. Oracle JDeveloper provides a complete environment for the database developer. The database modeler lets developers design the structure of their database including tables, columns, and relationships. It also provides the capability to reverse engineer existing database schemas. Wizards let developer create database objects such as tables, views, and code components. The database browser provides an easy view of database components, and the table viewer shows the structure and data of any table.

The built-in SQL Worksheet lets developers execute any SQL command and shows its explain plan, helping optimize the database access.

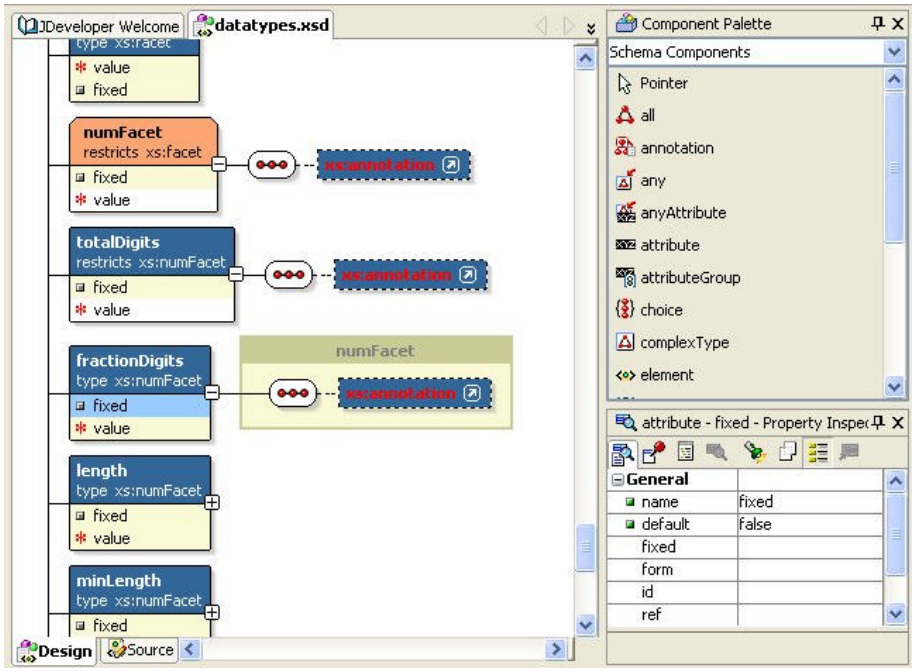
The code editor is aware of the PL/SQL syntax and provides code-insight and color-coding, helping developers code PL/SQL faster. Developer can also run and debug their PL/SQL code inside JDeveloper.

XML Development

For XML developers Oracle JDeveloper provides a visual XML Schema editor, that lets them browse XML schemas easily. XML Schemas can be constructed visually using drag-and-drop from the component palette.

The XML aware editor provides code insight while creating XML files, and the property palette provide an easy way to define tag attributes declaratively.

Oracle JDeveloper includes the Oracle XML developer kit (Oracle XDK) that offers libraries that can be used to create XML based applications.



XML Schema Modeling

CONCLUSION

Addressing the ever-increasing demands placed on software developers, Oracle JDeveloper 10g provides a complete and integrated development environment with exceptional productivity boosting features. Addressing Java, SQL, and Web services development as well as the full development life cycle, Oracle JDeveloper is a powerful tool for both new and experienced developers.

Oracle JDeveloper 10g Overview
March 2004
Author: Shay Shmeltzer
Contributing Authors:

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com