

ORACLE9i DESIGNER IN THE TWENTY FIRST CENTURY

Sue Harper, Oracle Corporation

INTRODUCTION

Oracle9i Designer delivers dramatic increases in productivity for database application developers with a complete toolset to model, generate and capture the requirements and design of applications. Using Designer's RAD-like, intuitive, task-oriented development environment, and developers can focus on the needs of the business and not the technology to develop and generate automatically the functionality needed for enterprise applications.

Developers can combine Designer's wizard-based, integrated toolset with Oracle9i JDeveloper to create one effective environment for developing three-tier, component based J2EE database applications. Designer provides extensive capabilities for modeling and generating database-oriented, web-based applications deployed using Java, XML, Oracle9i Forms and Reports, and dynamic HTML. Core to this capability is Designer's powerful database design, generation, and capture functionality, globally recognized as the only complete solution for Oracle9i database design.

DESIGNER STATUS

The Oracle Designer team is pleased to announce several enhancements to the Oracle9i release. These enhancements include improvements to Web PL/SQL generation and the addition of a new browser-based UI for the repository. We plan to release these and other new features in the regular maintenance releases of Designer as we continue our commitment to quality.

Releasing Designer in Oracle9i Developer Suite (9iDS) was no small accomplishment. In years past, we had the luxury of letting other products complete their development cycles before finalizing our own, but for 9iDS we needed to be synchronized with the full suite at release time. And, as Designer is now a mature product with 15 years of development history, our focus has shifted. Rather than developing substantial new features in Designer, we have moved toward supporting the development of UML tools in JDeveloper.

ORACLE DESIGNER 6i

Following the major release of Designer 6i, the development team has commenced a regular patch cycle with the released and planned patches as follows:

- Release 4.2 – March 2002
- Release 4.3 – May 2002 (**NB** This release of Designer is dependent on using Forms patch10)
- Release 4.4 – August 2002 (Production)
- Release 4.5 – November 2002
- Release 4.6 – February 2003 (This release includes ROB)

DESIGNER 6i OBSOLESCENCE

The current plans are to support Designer 6i and SCM 6i through to 31-Dec-2004, with the intention to review to extend support for a further two years until 30-Jun-2004, if we can determine there is a requirement from customers to do so. In either situation extended support beyond the de-support date remains an option.

GENERATING CLIENT SERVER FORMS

The reason for the extended Designer 6i support is due to client server generation for Forms 6i, which is the terminal release for building client server forms. Oracle9i Forms provides an environment for building forms for deploying

on the web and as such 9i Designer only generates forms for deploying on the web. For more information on this please refer to the documentation under the Forms section on OTN (<http://otn.oracle.com/products/forms/content.html>)

ORACLE9iDESIGNER

The 9iDS release of Designer is a consolidation of Designer 6i against the new software stack. 9i Designer designs, generates, captures, and installs into an Oracle9i database. It co-exists in the same Oracle Home as the other 9i products, and generates and captures the sister tools in the suite: 9i Forms and 9i Reports. Generated Web PL/SQL application runs on a 9iAS V2 server.

9i DESIGNER STATUS

9i Designer is part of Oracle9i Developer Suite Release 2 and Oracle9iDS 9.0.2.0.1, which was a re-release of the 9iDS pack for 9iAS NT. The development team has commenced a regular patch cycle:

- Release 9.0.2.1 – June 2002
- Release 9.0.2.2 – August 2002 (*Production*)
- Release 9.0.2.3 – November 2002
- Release 9.0.2.4 – February 2003 (includes ROB)

DESIGNER FUTURE STRATEGY

It is not our intent to integrate Designer and JDeveloper more than to provide basic compatibility within 9iDS. Designer tools will not be available from the JDeveloper IDE, nor will JDeveloper tools be available from Designer. So will you see Designer functionality in JDeveloper? The development team is addressing this issue now. It will take time to extract the best ideas from Designer, reengineer them in Java for J2EE architectures, within the context of a single IDE, all the while fully adhering to the Unified Modeling Language (UML) that has become the *lingua franca* of the modeling world.

We are also working on a strategy that will allow Designer users to retain and leverage the intellectual property reflected in their repositories as they move into J2EE development. There are potentially a number of ways in which Designer structured data could be made available to JDeveloper components. One option is to generate BC4J components based on Designer's table definitions, module components, and module definitions. This feature is now in its final development cycle and is planned for release in JDeveloper 9.0.4.

A FEW ENHANCEMENT REQUESTS

Designer development polled the larger user community for enhancement requests. One frequent request was for a Web interface to the repository, one that would provide the business user with ready access to the repository. The solution presented itself at our doorstep and, as a result, Designer development has assumed responsibility for a former Oracle Consulting solution, Oracle Designer Web Assistant. Renamed the Repository Object Browser, it will become an integrated part of the Designer repository.

Other planned enhancements will incorporate updates of the Database Design Transformer: improving the initial dialog, ensuring that entities are sorted alphabetically, and improving overall usability. We will also be implementing a long-standing request to return the Domain column to the Attributes tab of the Edit Entity dialog in the ERD.

With the Oracle9i stack stabilized in the first release of Oracle9i Designer, we plan to create some new features Designer post 9i. There will be better support for existing database objects, further support for new Oracle9i database objects and structures and some much-requested enhancements to the Web PL/SQL generator and the Database Design Transformer.

BC4J GENERATION

BC4J FRAMEWORK

Oracle Business Components for Java is a standards-based, server-side framework for creating scalable, high-performance J2EE applications. The framework provides design-time facilities and runtime services to simplify the task of building, debugging, customizing, and reusing business components. The applications developed with Business Components for Java can be deployed on any J2EE platform and can be accessed from a wide variety of clients including Web browsers, professional desktop clients, XML clients, and wireless devices.

BC4J GENERATION FROM DESIGNER

Oracle Development has produced an extensible framework that enabled the development of a BC4J generator. In time, we would like to see partners use this framework to build other similar generators or transformers. The BC4J Generator is the first of the tools to make use of the transformer framework, reading Designer's rich meta-model to extract table and module definitions to generate BC4J components.

At a high level the generator provides the following mappings:

- Table Definitions -> Entity Objects
- Module Components -> View Objects
- Module Definition -> Application Module

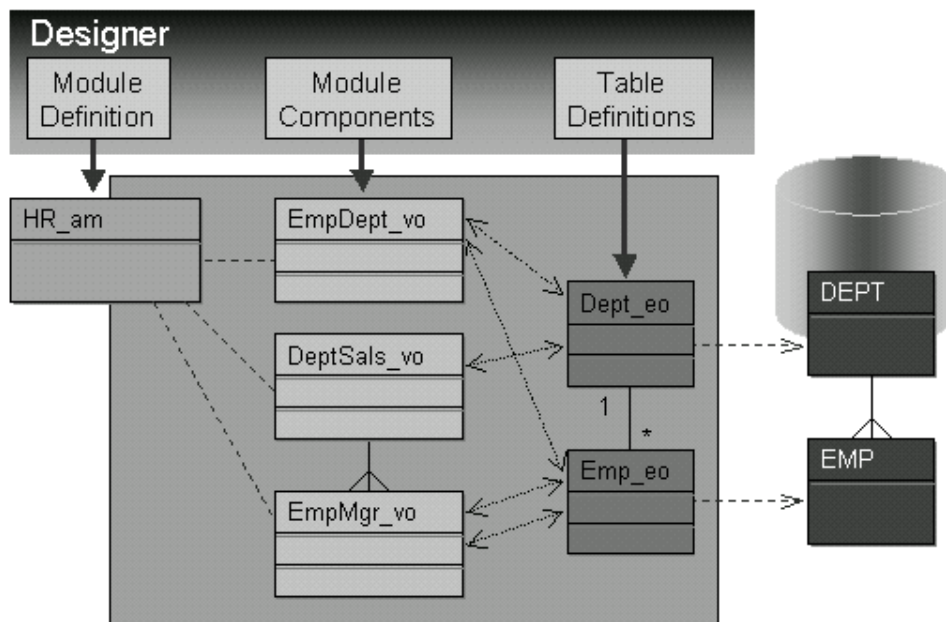


Figure 1: Mapping Designer Module Components to BC4J

WHY BC4J GENERATION FROM DESIGNER?

Developers using BC4J with Oracle9i JDeveloper benefit from interactive wizards, editors, testing tools, performance profilers to design, build and debug code for building fully automated J2EE application packaging and deployment. Using the JDeveloper IDE, development teams can focus immediately, and only, on writing business logic and user interfaces instead of on designing, coding, and debugging handcrafted application-plumbing code.

If JDeveloper provides such a rich and interactive development environment, why is there a need for a BC4J

Generator from Designer? Users with modules defined in Designer can make use of the rich meta-data stored with in the Designer repository to generate BC4J components instead of starting with the wizards in JDeveloper.

The BC4J Generator makes use of table usages and certain UI details, such as prompt and hint detail, to build entity objects in JDeveloper.

Companies may find that they have made significant inroads in design decisions and have designed many, often complex, modules and yet with the changing markets are finding that their development teams consist of more Java developers than Designer and PL/SQL developers. They may want to make use of their design investment and start development on applications for the J2EE platform.

By using the generator, companies can generate middle tier business definitions based on their Designer definitions.

BC4J GENERATION FROM DESIGNER: THE DETAILS...

TABLE DEFINITIONS ARE GENERATED AS BC4J ENTITY OBJECTS

The initial part of the generation uses Designer table definitions, together with column definitions, as a source to produce the entity objects in JDeveloper. In Designer it is possible to set up domain definitions that hold valid values and associate these with one or more columns. Additionally you can define a range of valid values for a column. For both of these scenarios, the generator creates a BC4J List Validation Bean.

Primary Key constraint definitions become BC4J keys, while Foreign Key constraints definitions are generated as BC4J Associations.

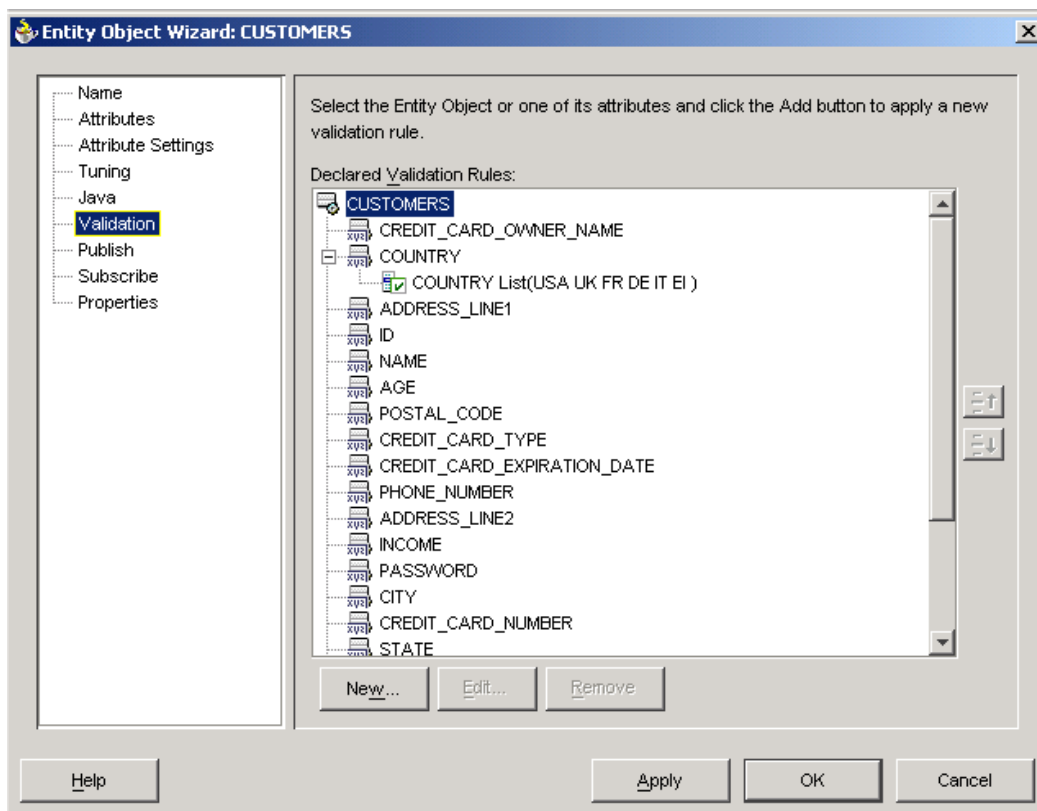


Figure 2: Entity Object wizard, displaying generated list validation for an entity attribute

MODULE COMPONENTS ARE GENERATED AS BC4J VIEW OBJECTS

The BC4J Generator reads the Designer repository definitions for module components to produce entity objects. Module component bound items are generated as view object attributes, with Key Based Links generated as BC4J View Links. Any UI details set such as prompts, hints, and item width and height are also generated into the attribute properties.

Module Components with computed unbound items become separate BC4J View Objects. Unbound items of the type 'SQL expression' or 'Server Side Function' become BC4J calculated view attribute and unbound item of type 'Computed' are generated as BC4J summary view attribute.

If a Where Clause is defined in Designer it is appended to the View Object SQL query. (See Figure 3, below)

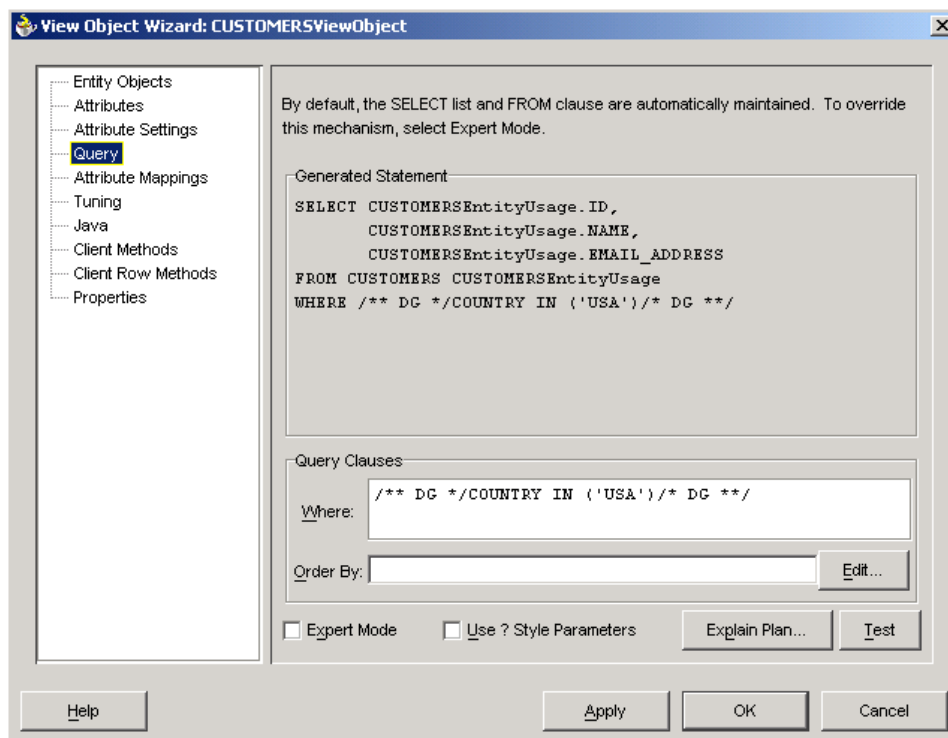


Figure 3: View Object wizard displaying the generated query including a Where Clause

ITERATIVE GENERATION

The generation process need not be a single iteration. If a where clause is added to the query in JDeveloper, future generations append any additional Where Clause statements defined in Designer.

DESIGNER LIST OF VALUES GENERATED AS A BC4J VIEW OBJECT

The generation process creates a separate view object for a List of Values (LOV) defined in Designer. This means that a separate query with its own Where Clause from the LOV is created. The LOV usage in Designer becomes a BC4J List Validation Bean.

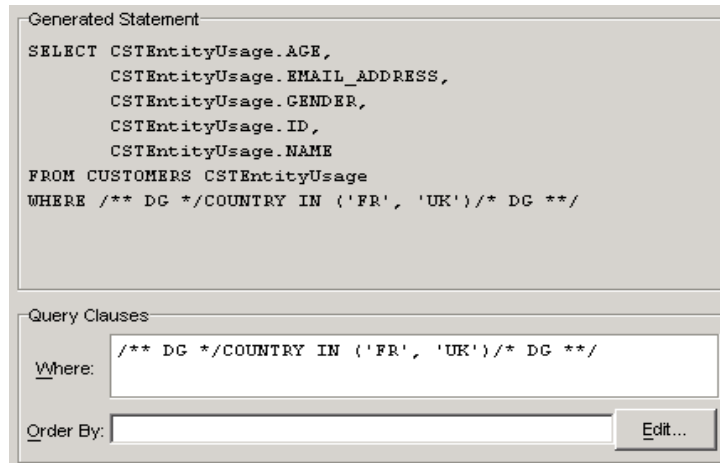


Figure 4: List of Values SQL query includes a generated Where Clause

MODULE -> APPLICATION MODULE

The application module is made up of all the view objects defined, based on view links created.

BC4J GENERATION FROM DESIGNER: THE PROCESS*DESIGNER INPUT*

The BC4J Generator reads application modules for Forms, Reports, Web PL/SQL and Visual Basic, picking up the module component and related details as described above.

THE GENERATION PROCESS

Starting in JDeveloper, you need a workspace and project to work in. Once these are set up, you can invoke the generation wizard.

During the generation process you need to supply the connection information for:

- The repository user who owns the modules for generation.
- The user who has access to the database objects that the BC4J components need to bind to.

You need to supply the generation wizard with the workarea and application that owns the modules required for generation.

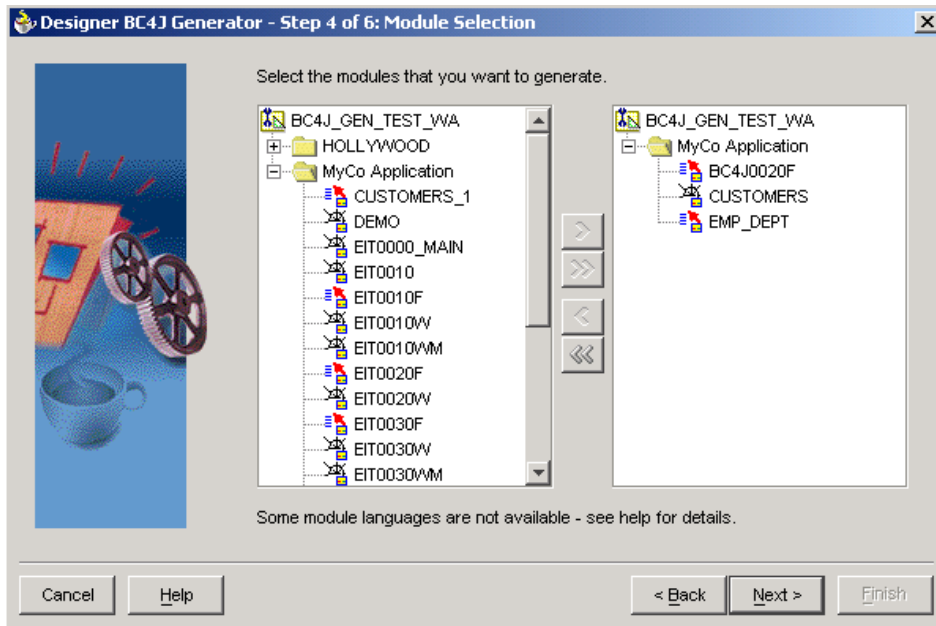


Figure 5: Selecting Designer application modules for Generation

JDEVELOPER BC4J GENERATED OUTPUT

The output from the generation process includes Entity Objects, View Objects, and Application Modules, View Links and Associations.

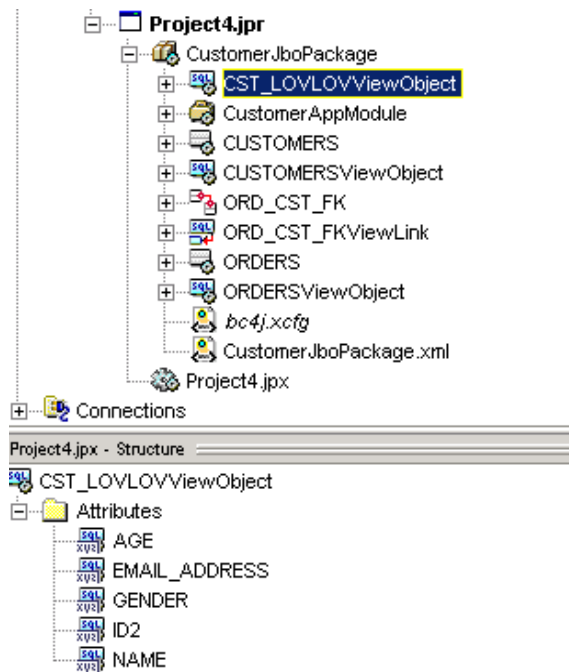


Figure 6: The generated output showing the components including the additional LOV View Object and structure

TESTING THE RESULTS

The Business Components Browser is a Java client provided with JDeveloper. The client allows you to view and update data, without having to code a custom client and GUI.

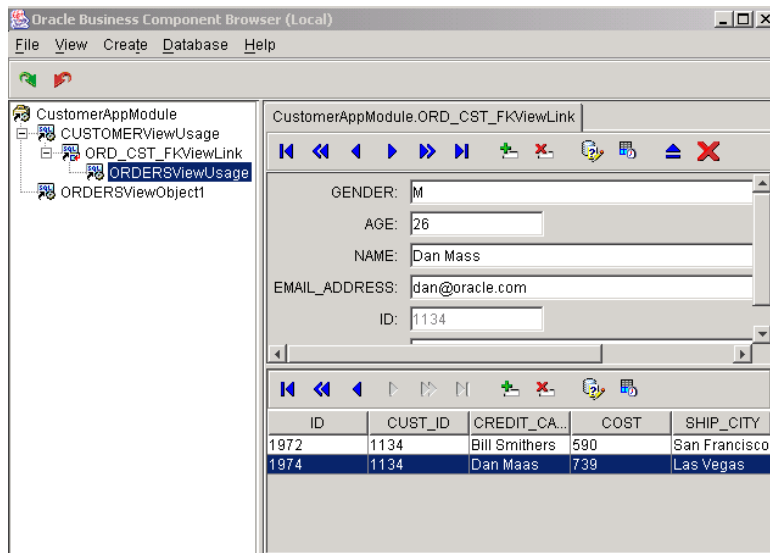


Figure 7: Generated output displayed in the Business Components Browser

WAYS OF WORKING

Using the BC4J generator you can create a first cut BC4J application based on existing Designer modules. It is equally feasible to start working in a JDeveloper project, building business components and then to invoke the BC4J Generator to add additional components from a Designer module. A regeneration facility allows you to modify your modules in Designer and regenerate these into the project, thus updating the project.

BUILDING USER INTERFACES

The BC4J Generator does not generate any UI. While it does set up certain properties, such as prompt and width and height as previously mentioned, you still need to build a UI. Use the JDeveloper wizards to assist you.

You have two main options for producing the UI of your generated application. You can choose a Java/Swing or a JSP/HTML UI. There are many parallels between the Forms UI and the Java clients that you can build using JDeveloper.

- If you are accustomed to the Generated forms user interface, using JSP/HTML implies not only a change in the user interface, but also a basic change in the fundamental way end users interact with the application. For example, you don't have the flexibility of the rich Forms menus and keyboard short cuts in HTML and the concept of a stacked canvas is not an HTML construct.
- A Java UI is very rich with a vast array of visual controls, menus, pop-up dialogs and potentially sophisticated navigation including stacking of screens and dialogs.

ORACLE UIX WEB APPLICATION FRAMEWORK

The Oracle UIX web application framework simplifies the implementation of complete, sophisticated web applications by providing reusable technology to assist with many of the aspects you would typically have to write code to handle yourself. For example, UIX provides off-the-shelf assistance with web page rendering, data binding, event handling, page-to-page navigation, and customization. It offers a comprehensive set of pre-built page layout

elements to deliver web applications with a rich, consistent look and feel across many different client agent types. UIX provides built-in support for BC4J data binding, too, so including or modifying data from any BC4J view object is easy. JDeveloper provides a wizard to help you build a basic UIX Client.

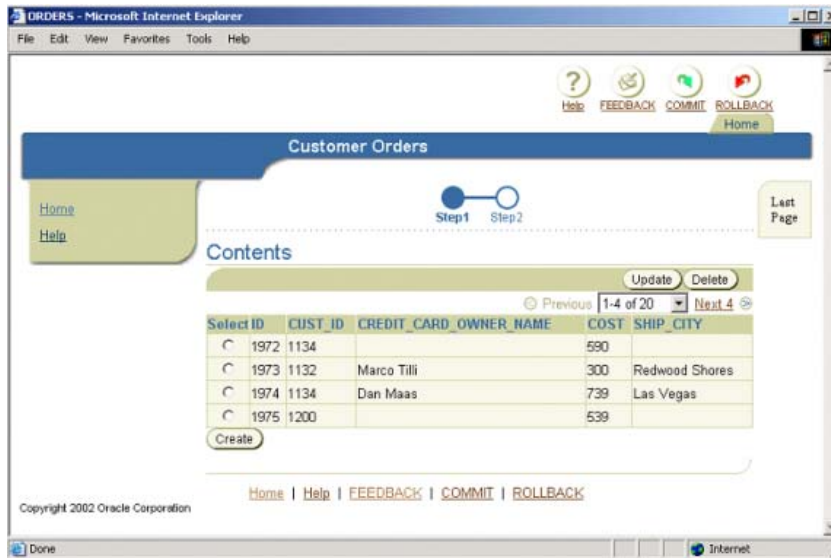


Figure 8: UIX Client

DATABOUND JSP PAGES WITH THE BC4J DATA TAGS

BC4J Data Tags are a JSP 1.1-compatible tag library for easily rendering, navigating, and updating data from any view object in any application module. Oracle9i JDeveloper contains integrated support for using the Data Tags in your JSP pages, including wizard-driven tag assistance and context-sensitive "Tag Insight" help for always knowing what tags and attributes are legal anywhere that you are typing in your page. In addition, JDeveloper supports running and debugging JSP pages locally for an even quicker test-fix-retest cycle, avoiding a deployment step while still in development. JDeveloper provides a wizard to help you build a basic JSP Client.

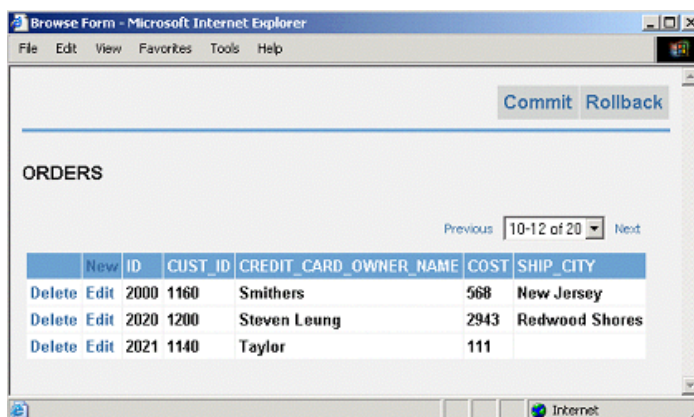


Figure 9: JSP Client

AVAILABILITY

The BC4J Generator will be incorporated in JDeveloper 9.0.4

WHERE DOES JHEADSTART FIT IN?

Oracle Consulting has long been involved in offering framework services to customers, JHeadStart being the latest of these. Among other features, the JHeadStart service provides a mechanism to take existing Designer artifacts and generate BC4J components from them.

JHeadStart is a rapid component based development approach for building J2EE applications. It enables fast, reliable, and repeatable development of complex transactional systems. It combines three frameworks to implement the MVC architecture, providing generic reusable components on top of the component library of the MVC Framework. It includes the JHeadStart Application Generator, that allows you to declaratively specify your application definition in an XML-file with simple to use property editors and enables you to generate into the above frameworks. Since all frameworks are XML-based you can easily extend or customize the generated application. In addition with the Designer Generator, JHeadStart offers an intuitive migration path from Designer/Forms to Java/J2EE.

REPOSITORY OBJECT BROWSER

OVERVIEW

The Repository Object Browser (ROB), like ODWA, is the solution for web-publishing meta-data stored in a Designer repository. An easy browser-based UI affords access to all the meta-data of your information systems. It now also gives you the opportunity to set up a web-based project collaboration environment with the Designer repository as its foundation. Without having to install, use or even know about Designer, project staff can check in and upload project documentation through an easy to use interface running in a standard web browser.



Figure 10: Front Screen of the Designer Repository Object Browser

ROB FUNCTIONALITY

In the initial release, ROB functionality will be as close to that of ODWA as possible. There are some updates to the UI and a few other minor modifications. The tool is primarily a PL/SQL application that offers a dynamic browser-based UI to navigate Designer repository meta-data.

ROB consists of the following tools:

REPOSITORY NAVIGATOR

The Repository Navigator offers the user a navigator-style interface to browse through repository work areas and configurations. It also supports check out, check in, undo check out, file download, and file upload.



Figure 11: Repository Navigator

REPOSITORY REPORTS

Repository Reports offers a number of reports based on the element types in Designer. For many element types, ROB provides reports with all the relevant details of selected objects. Some Designer sites may have used other Oracle Consulting offerings and as a result have CDM-style objects in their repositories. These are specific to Oracle iDevelopment Acceleration Suite implementations and so are not available to all Designer users. ROB has an additional preference interface to allow users to hide these CDM-style objects.

REPOSITORY SEARCH

Repository Search is a flexible, easy-to-use yet powerful tool to search the Designer repository. It also offers more complex search facilities such as searches based on version information.

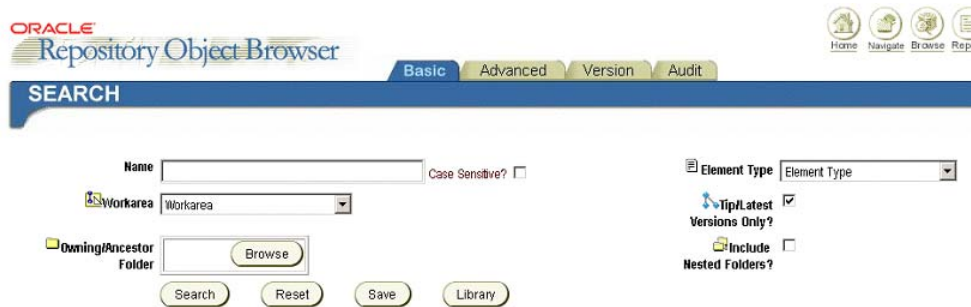


Figure 12: Repository Search

REPOSITORY OBJECT BROWSER

Repository Object Browser is read-only UI that provides similar functionality to the Repository Object Navigator (RON). Once an item is selected in the navigator, a panel displays the object's properties. Users can drill down through the repository selecting objects and finding details about these objects.

MANAGEMENT FACILITY

The management facility allows users to view user access to workareas and applications. Additionally users can include diagrams and associate them with items, creating hotspots to allow the browser to navigate to the associated object.

ONLINE HELP

Online help provides comprehensive information about all the functionality and components of the ROB.

RELEASE TO MARKET

Repository Object Browser functionality is released in the following maintenance releases:

- Oracle Designer 6i Release 4.6
- Oracle9i Designer 9.0.2.4

These releases are planned for early 2003.

LICENSING

In order to use Repository Object Browser a user must have licensed access to:

- An Oracle database
- An HTTP server supporting mod_plsql

The licensing requirements are therefore the same as for any other database application that uses an HTTP server. This HTTP server can either be that supplied with Oracle9iAS or the one installed with the Oracle8i or Oracle9i database.

Organizations already licensed to access the Oracle database are also licensed to use the HTTP Server supplied with the database. Similarly, existing Oracle9iAS licenses will deliver the required HTTP server requirements - especially if the Oracle9iAS licensing agreement is CPU-based (as opposed to user-based).

Users wishing to access the Designer repository using ROB alone do not require a Designer client install. In this case there is no requirement for any additional Oracle9iDS licenses. Such ROB-only users will however still require Oracle database licensing, and potentially 9iAS licensing.

SUMMARY

Oracle9i Designer offers a comprehensive toolset that enables you to 'fast track' your development activities for e-business and web applications. It supports the latest release of the Oracle databases, Oracle8i **and** Oracle9i and integrates fully with other development tools such as 9i Forms. Coupled with Oracle SCM, 9i Designer delivers a productive environment for developing small or larger scale, complex applications. 9i Designer offers a powerful, flexible, efficient, accurate and automatic approach that is easy to manage and maintain. Use it to reduce the amount of coding necessary, focus on analysis and design and therefore improve the quality of the finished application by hitting the target first time.