

An Oracle White Paper
September 2010

Maximizing Development Productivity with Oracle Data Integrator *11g*

ORACLE®

Disclaimer

This document is for informational purposes. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

Executive Overview

Oracle Data Integrator (ODI) is a best-of-breed data integration platform focused on fast bulk data movement and handling complex data transformations. The 11g version of ODI advances this state of the art technology even further ahead of the rest of the industry.

One of the key product objectives with Oracle Data Integrator 11g is to maximize ETL developer productivity and efficiency by providing new features that help quickly deliver high quality integration processes and meet project requirements ahead of schedule and under project budget.

This whitepaper describes in detail these new productivity features and capabilities offered in the ODI 11g data integration platform.

Defining Productivity

Development Productivity can be defined as “*the rate at which one can develop quality deliverables that meet the requirements*”.

While meeting the project requirements may not be at the heart of every productivity discussion, it is the single most important driver for any project, regardless of the particular development methodology in place. Meeting these projects requirements *on time* is a totally different story, and developer productivity is the main factor in meeting this challenge of time.

Productivity discussions for ETL projects can include a variety of concerns, including code/mapping creation, maintenance efficiency, detecting issues and avoiding errors.

Oracle Data Integrator 11g introduces new capabilities to support productivity, including:

- An Enterprise Scale Infrastructure for Development
- An Efficient Design-Time Experience
- Better Error Control & Management
- A Java API for Task Automation
- An Enhanced Declarative Design Approach

Enterprise Scale Infrastructure for Development

A reliable, efficient and secure infrastructure supporting a powerful solution is critical for any development effort. In short, no one wants developers to spend their time “waiting for the system to respond or to be back online”. An unreliable, unavailable or insecure development infrastructure is a waste of time, money and productivity.

Oracle Data Integrator 11g provides several new features for an *Enterprise-Scale Architecture*. These features guarantee that:

- Any component of Oracle Data Integrator can be always available
- Any unexpected failures in the infrastructure can be quickly identified and resolved
- The infrastructure is integrated with popular enterprise security systems.

Features for an Enterprise-Scale Architecture include:

High-Availability (HA) and Scalability supported via clustered deployments for Java EE and Standalone components. Java EE components deployed in WebLogic Server benefit from the capabilities of the WLS cluster and Coherence data grid for scalability, including JDBC Connection Pooling and Load Balancing. Standalone agents support the ODI built-in load-

balancing feature and can be protected using Oracle Process Manager and Notification Server (OPMN).

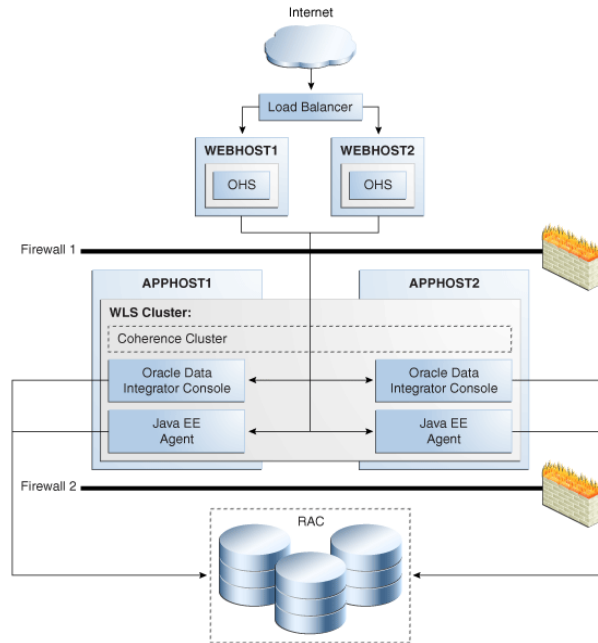


Figure 1- High Availability Configuration for Oracle Data Integrator JEE Components

Connection Retry mechanism to transparently recover sessions with repositories stored in highly available database engines such as Oracle RAC.

Stale Sessions Management: Sessions stopped due to a repository or the run-time agent failure are automatically identified as stale and moved from the *Running* state to the *Error* state for review.

Unified management via a plug-in that integrates in Enterprise Manager Fusion Middleware Control to monitor the status, metrics, and notifications of Oracle Data Integrator components, including Repositories and Agents.

New Oracle Data Integrator Console that replaces and unifies the former 10g Metadata Navigator and Lightweight Designer components. This web interface for monitoring and metadata browsing has been rewritten using the Oracle ADF-Faces Ajax Framework for a rich user experience.

Support for external password storage to safeguard data servers and context passwords into a corporate credential store.

Support for external authentication to have ODI users authenticate against an external enterprise identity store (LDAP, Oracle Internet Directory, Active Directory), using their Single Sign-On (SSO).

Efficient Design-Time Experience

Oracle Data Integrator 11g introduces a new JDeveloper-based Integrated Development Environment (IDE) called the Oracle Data Integrator Studio. This client is redesigned in this release to dramatically increase the developer's productivity and simplify access to advanced features.

New IDE Based on JDeveloper

The new Oracle Data Integrator Studio, based on Oracle JDeveloper replaces all Oracle Data Integrator 10g graphical modules (Designer, Topology, Operator, and Security Manager).

The new IDE provide state-of-the-art features for developer efficiency, including redesigned and reorganized editors in addition to enhanced windows, improved docking, and document navigation and management.

Single IDE with Multiple Navigators

The features of the ODI modules now appear as Navigators within the Oracle Data Integrator Studio window. This new organization provides the following features:

Navigators can be docked/undocked and displayed/hidden using the View menu. These Navigators allow access to the former module-specific actions from their Navigator toolbar menu (for example, the export/import master repository operations in the Topology Navigator)

Accordions group the tree views that appear in the Navigators (for example the Project and Models accordions in the Designer Navigator). Accordions that are not frequently used can be minimized into the lower section of the Navigator to allow more room for the other tree views. Accordions allow access to the tree view-specific actions from their toolbar menu (for example, import project from the Project Accordion in the Designer Navigator).

Tree Views objects are provided with context menus and markers, they can be dragged and dropped within a tree view or across tree views for defining the security policies. Double clicking an object opens by default the corresponding Object Editor.

Context Menus have been reorganized into groups with separators and normalized across the interface.

Window Management

The windows, editors and navigators in the Oracle Data Integrator Studio benefit from the JDeveloper IDE features, including:

Full Docking Support: All windows, editors and navigators can now be docked, undocked, stacked and organized intuitively. Stacked items appear with a tab as their title.

Fast maximize and restore: Quickly maximize or restore a dockable window or editor by double-clicking on the title bar.

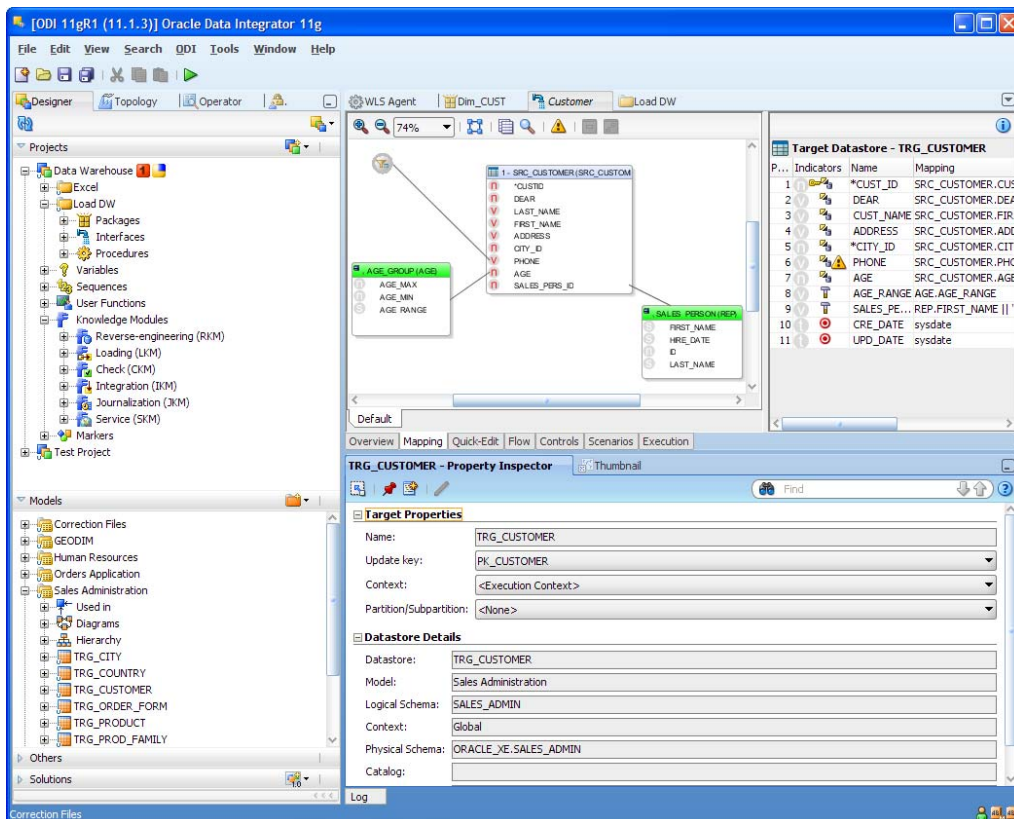


Figure 2 – In Oracle Data Integrator Studio, Editing an interface (visible editor), a model, a package and a physical agent.

Document Management and Navigation

Object edition has been enhanced with the Oracle Data Integrator Studio. This includes:

Save and close multiple editors: You can easily save all your work with a single click using the **File > Save All** option and close all opened editors similarly. You can also close all the editors but the current one.

Forward and back buttons: Now you can easily return to a previously visited document with the convenient browser-style forward and back buttons on the main toolbar. These buttons

maintain a history, so you can drop down the back or forward button to get a list of the documents and edit locations you have visited. **Alt+Left** and **Alt+Right** activate the back and forward buttons.

Quick document switching: Switching between editors and navigators is also possible. Now when you press **Ctrl+Tab** or **Ctrl+F6**, you can choose which document you want to switch from a list ordered by the most recently used. You can use the same technique to switch between open dockable windows by first placing focus in a dockable window, then pressing **Ctrl+Tab** or **Ctrl+F6**.

Redesigned Editors

All object editors in Oracle Data Integrator have been redesigned for better usability.

Main changes include:

Tabs are organized as finger tabs on the left hand-side of the editor. Complex editors (as for example Interface or Package Editors) have also tabs appearing in the bottom of the editor.

Fields have been grouped under headers. These field groups implement an expand/collapse behavior.

Fields and labels have been organized in a standard way for all editors for a better readability of the editors.

Text Buttons in the editors are transformed into hyperlinks, and all buttons appearing in editors have been redesigned.

Knowledge Modules, Actions and Procedure editors have been redesigned in order to edit the Lines directly from the main editor instead of opening a separate editor.

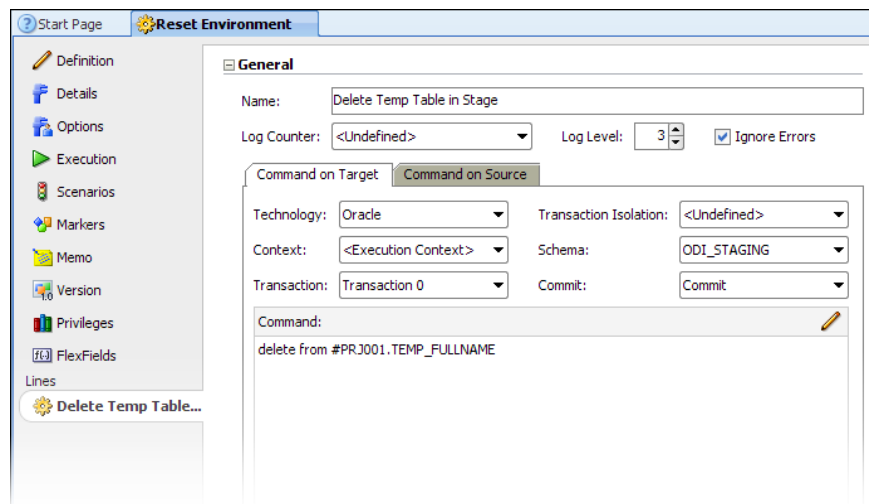


Figure 3 - Redesigned Editor Example: Procedure Editor.

Redesigned Interface Editor

The editor used to create the integration interfaces has been entirely redesigned to use the JDeveloper diagramming framework.

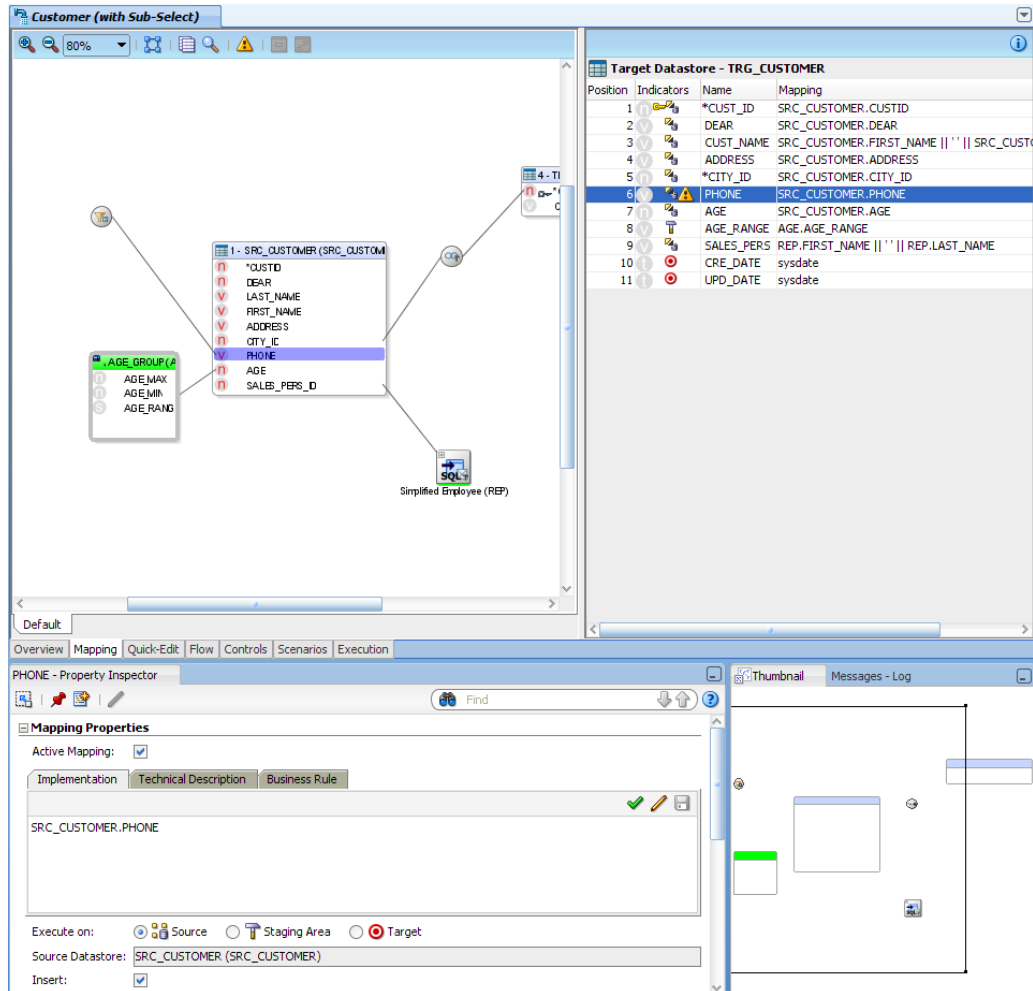


Figure 4- Interface Editor (Mapping) with Thumbnail view and compacted lookup object.

This new diagram provides the following benefits:

- Improved look and feel and better user experience

- Support for graphical options on diagram objects. For example, compact and expanded view can be used for better readability.

- Thumbnail and zoom in/out is supported on the sources and flow diagram to navigate large diagrams.

- Multiple source columns can be dropped directly onto a target column for faster mapping.

Target mapping table is improved. Mapping properties (Position, Indicator, Name and Mapping expression) can be displayed selectively and sorted.

Sources, targets, filters, joins can be selected and edited directly in the flow diagram.

Source column drag and drop in the target mapping

Reorganized and searchable Property Inspector for interface components (joins, lookups, etc)

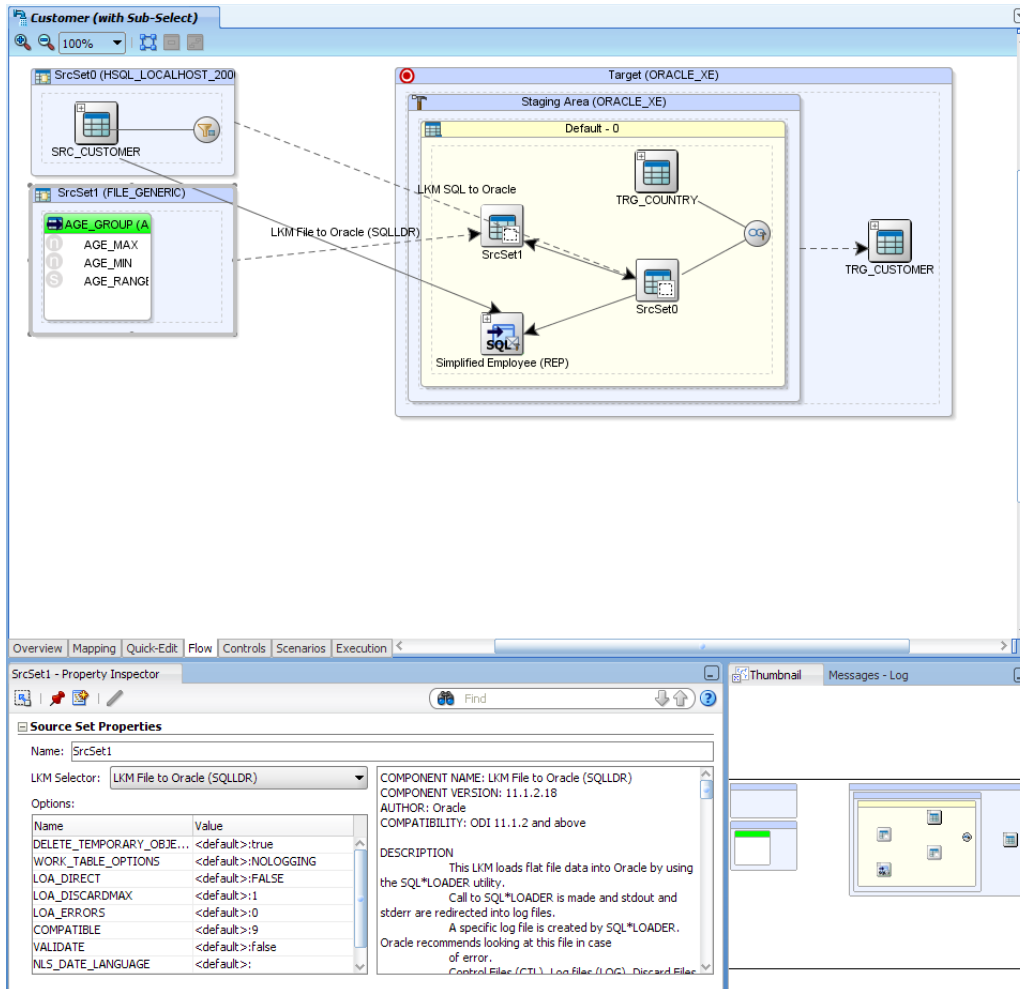


Figure 5- Interface Editor (Flow), with expanded source datastore and thumbnail view.

Quick-Edit

The new interface editor includes a new Quick-Edit tab to edit the interface diagram faster. The Quick-Edit tab displays the interface components in a tabular form, and allows faster creation of edition of an interface.

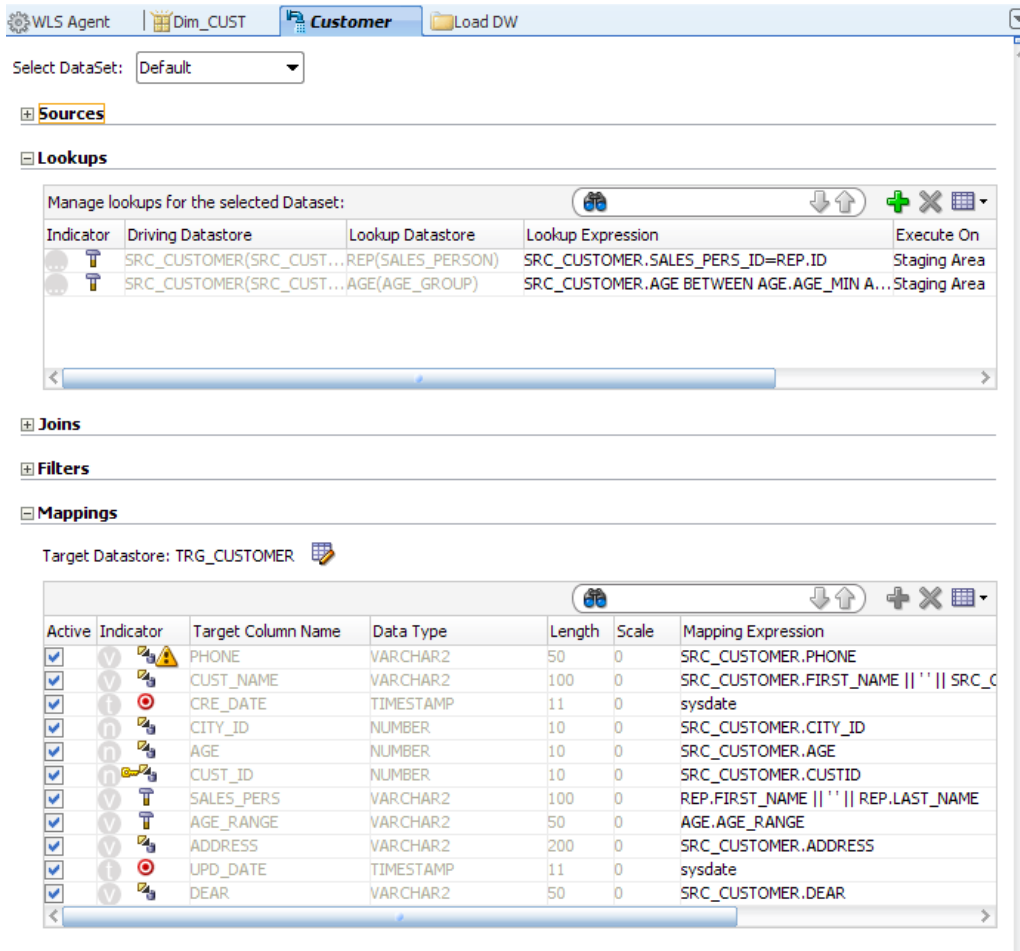


Figure 6 –Interface Editor - Quick Edit View

Quick Edit supports

- Intuitive and Accessible Keyboard Navigation
- Multiple cell updates and Copy-paste operations
- Column displayed selectively and can be sorted

Quick Edit facilitates interface development, review and maintenance by providing a simple and compact view of the interface content.

Other Usability Features

In addition to these major changes, other improvements have been made in the ODI Studio to make the developer's life easier. For example:

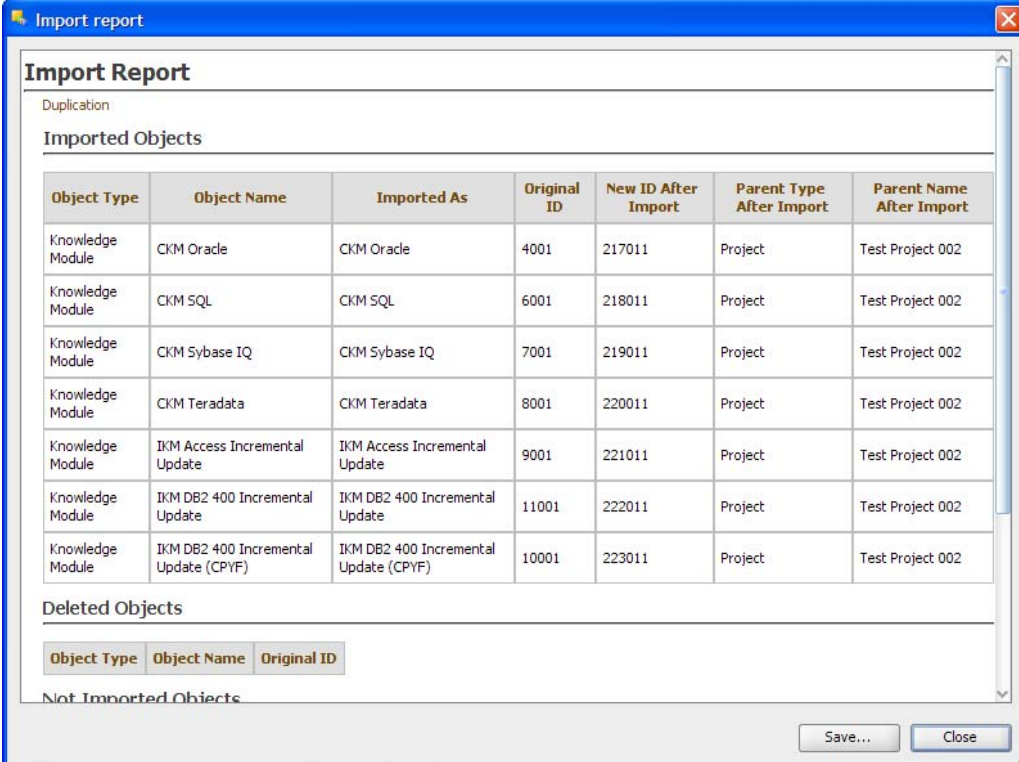
Reverse-Engineering -When a model is created, the reverse-engineering context is automatically set to the default context. In addition, when performing a selective reverse-engineering, the system tables are now hidden from the display.

Scenario Naming Convention – A user can define a pattern (using for example the object name, folder path, or project name) for the naming convention that is automatically applied to new scenarios created by this user.

Long Object Names - Object names have been extended to support long database object names (128 characters) and repository object labels (400 characters).

Purge Log on Session Count - The OdiPurgeLog tool has been enhanced to support a purge of the log while retaining only a number of sessions in the log. Purged sessions can be automatically archived by the tool before performing the purge.

Import Report - After objects have been imported, an import report displays the objects that have been imported or deleted in the target repository. Missing objects referenced by the imported objects are indicated as missing references. Such missing references can be fixed by a subsequent import. Import reports can be saved in XML or HTML format. With this feature, import becomes a fully traceable operation as all changes can be identified and archived.



The screenshot shows a window titled 'Import report' with a sub-header 'Import Report'. Below the header, there is a section for 'Imported Objects' containing a table with the following data:

Object Type	Object Name	Imported As	Original ID	New ID After Import	Parent Type After Import	Parent Name After Import
Knowledge Module	CKM Oracle	CKM Oracle	4001	217011	Project	Test Project 002
Knowledge Module	CKM SQL	CKM SQL	6001	218011	Project	Test Project 002
Knowledge Module	CKM Sybase IQ	CKM Sybase IQ	7001	219011	Project	Test Project 002
Knowledge Module	CKM Teradata	CKM Teradata	8001	220011	Project	Test Project 002
Knowledge Module	IKM Access Incremental Update	IKM Access Incremental Update	9001	221011	Project	Test Project 002
Knowledge Module	IKM DB2 400 Incremental Update	IKM DB2 400 Incremental Update	11001	222011	Project	Test Project 002
Knowledge Module	IKM DB2 400 Incremental Update (CPYF)	IKM DB2 400 Incremental Update (CPYF)	10001	223011	Project	Test Project 002

Below the table, there are sections for 'Deleted Objects' and 'Not Imported Objects'. The 'Deleted Objects' section has a header with columns 'Object Type', 'Object Name', and 'Original ID'. The 'Not Imported Objects' section is currently empty. At the bottom right of the window, there are 'Save...' and 'Close' buttons.

Figure 7 - Import Report indicates any object altered by an import operation.

Better Error Control & Management

Development is not about “dumping code” as fast as possible, and development in Oracle Data Integrator is not simply about creating interfaces and packages and hope that they will work. High quality development with ODI is about producing interfaces and package that are tested, validated and optimized for better performances.

Before reaching this test/validation/optimization phase, during the development phase, developers, regardless of their experience with the product may fall into common (or more complex) traps, as a consequence, the quality effort must start as early as possible in the design phase.

Preventing Errors

This new release of Oracle Data Integrator includes two new features for detecting and preventing errors at design-time. These features are *Auto-Fixing* and *Code Simulation*.

Auto-Fixing & Validation Report

When saving an interface or clicking the *Errors* button in the interface editor toolbar, the list of detected design errors is displayed with meaningful messages and tips. Automatic fixes are suggested and can be applied with a single click. In addition, when saving and attempting to execute interfaces, a report is raised with all remaining issues.

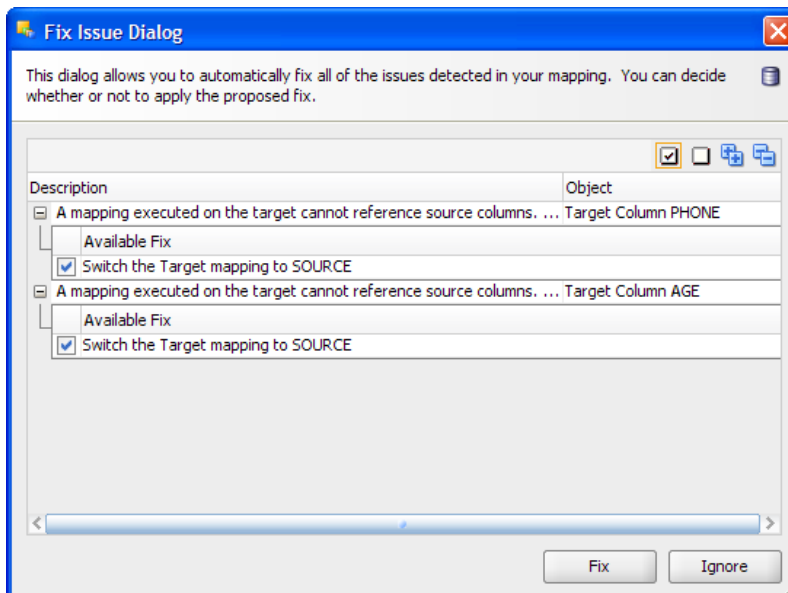


Figure 8- Auto-fixing suggests fixed to interface issues.

Code Simulation

When running a session at design-time, it is possible to simulate the generated code instead of running a full execution. This report includes complete session, step, and task information and contains the full generated code. The session simulation report can be reviewed and saved in XML or HTML format.

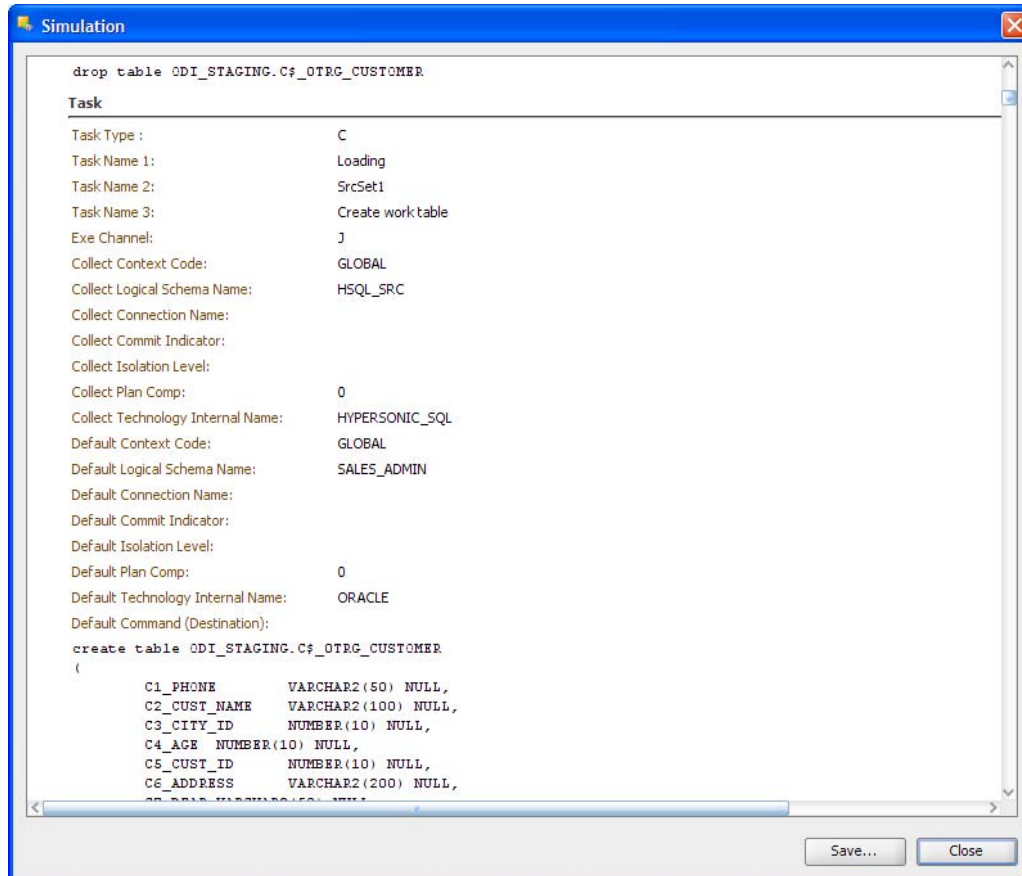


Figure 9- Code Simulation provides a preview of the generated code without executing it

With these new error prevention features Oracle Data Integrator developers can easily review the generated code for troubleshooting, debugging, optimization purposes, and archive this generated code for documentation.

Troubleshooting Errors

Occasionally errors will find their way into your ETL mapping, so Oracle Data Integrator also includes new features that can help with troubleshooting errors once the integration processes have been executed.

Aborting Session

Sometimes, a session is started that causes a long running statement over a source or target engine, this kind of problem can be due to a variety of reasons (large volume of data, incorrect join design, incorrect indexing, etc). Such sessions can now be stopped in an *immediate* mode. This new mode aborts the long running statement in the database engine instead of waiting for its completion to stop the session.

Enhanced Error Messages

Error messages raised by Oracle Data Integrator Components and Sessions have been enhanced and provide accurate information to:

Administrators and production operators for troubleshooting and fixing the status of the infrastructure.

Developer for debugging their executions or detecting infrastructure or topology issues.

Enhanced messages cover component lifecycle (agent startup, shutdown, schedule refresh and so forth) as well as session lifecycle (incorrect scenario version, load balancing issue, agent not available and so forth). They render clearer errors starting at the lowest level of the session (steps and task). These error messages are enriched with context and infrastructure information to help development or production users quickly find the cause and solution for an issue.

The screenshot displays the 'Session: Refresh Customer (119011)' details in the Oracle Data Integrator interface. The session status is 'Error'. The error message is as follows:

```

ODI-1217: Session Refresh Customer (119011) fails with return code 8000.
ODI-1226: Step Load Customer fails after 1 attempt(s).
ODI-1240: Flow Load Customer fails while performing a Loading operation. This flow loads target table NorthwindEmps.
ODI-1227: Task SrcSet0 (Loading) fails on the source MICROSOFT_SQL_SERVER connection MSSQL.
Caused By: java.sql.SQLException: [OWLS][SQLServer JDBC Driver]Error establishing socket to host and port: localhost:1433. Reason:
Connection refused: connect

```

Additional session details shown include:

- Session Name: Refresh Customer
- Status: Error
- Scenario Name: (empty)
- Execution Context: Global
- Scenario Version: (empty)
- Execution Agent: Local (No Agent)
- Start: Aug 26, 2010 11:30:03 AM
- End: Aug 26, 2010 11:30:30 AM
- Duration (seconds): 27
- Return Code: 8000

Record Statistics:

- No. of Inserts: 0
- No. of Updates: 0
- No. of Deletes: 0
- No. of Errors: 0
- No. of Rows: 0

Summary statistics at the bottom:

- No. of running child sessions: 0
- No. of successful child sessions: 0
- No. of failed child sessions: 0

Figure 10- Error messages are enhanced with contextual and infrastructure information.

Row-By-Row KMs for Debugging

Generic knowledge modules supporting row-by-row loading (*LKM SQL to SQL (row by row)*) and integration (*IKM SQL Incremental Update (row by row)*) have been introduced for debugging purposes. With these KMs, developers may trace each processed row and debug data-related errors.

Java API for Task Automation

Oracle Data Integrator 11g introduces a new Java API for creating and managing both design-time and run-time operations.

Using this API, it is for example possible to perform the following operations:

- Create or modify the topology contents
- Set up projects and models
- Create and browse interfaces, procedures, and packages
- Start and manage sessions and monitor their status

This API allows developers to programmatically perform operations. It can be used for:

- Building new applications accessing the Oracle Data Integrator Repository or interacting with run-time component (Starting and monitoring sessions).
- Script repetitive actions instead of doing them through Oracle Data Integrator Studio.
- Extend the Oracle Data Integrator Studio IDE by developing JDeveloper extensions using the API.

Enhanced Declarative Design Approach

With Declarative Design¹, Oracle Data Integrator uses an approach that dramatically reduces the effort at design-time. E-LT² provides maximum performance at run-time.

New core features have been built in Oracle Data Integrator 11g around the E-LT architecture and the declarative design approach to support a more productive development process and better performance at run-time.

Datasets and Set-Based Operators

A major new enhancement introduces the concept of datasets in interfaces. A dataset represents the data flow coming from a group of joined and filtered source datastores. Each dataset includes the target mappings for this group of sources. Several datasets can be merged into the interface target datastore using set-based operators such as Union and Intersect.

Figure 11 illustrates the use of datasets to UNION data from a source file and from an HR system. Each of these datasets (From File, From HR) is equivalent to an ODI 10g interface with its own set of source tables, joins, filters and mappings.

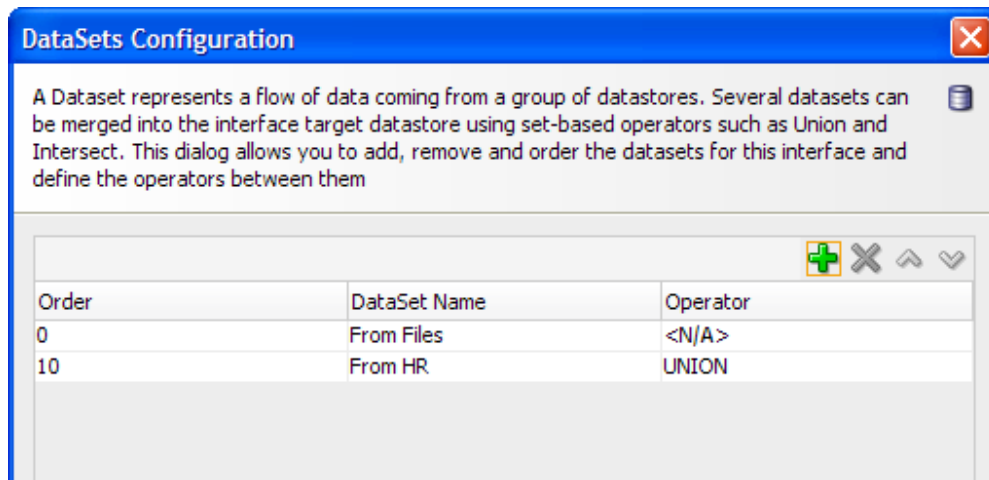


Figure 11 – One interface merges with a UNION operator flows from two diagrams

¹ The *Declarative Design* approach focuses on designing declarative rules separately from the flow into which these rules are executed. Oracle Data Integrator generates the flow using these rules and code templates named *Knowledge Modules*. This approach is opposed to the *Flow-based Design* that focuses on designing the execution flow logic.

² The *Extract-Load Transform* architecture uses the data servers in place (source, target or other) to perform the data integration processing, pushing optimized code to these servers. This approach is opposed to the *Extract-Transform-Load* approach that uses a dedicated and proprietary server to run all the data integration jobs.

These two flows appear in different diagram tabs in the Interface Editor, as shown in Figure 12.

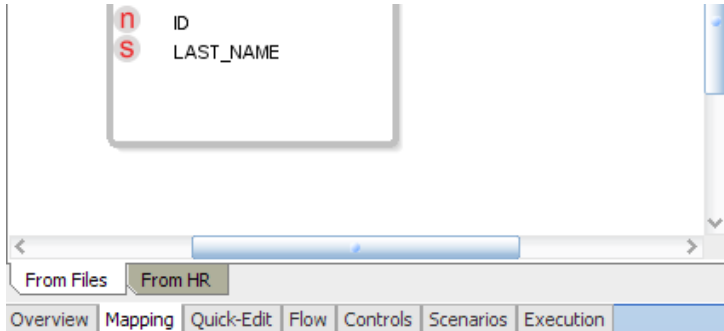


Figure 12 – Two diagram tabs for the datasets in the Interface Editor.

The management of the union datasets and of the sources, joins, filters, lookups and mappings for each of the datasets is done *within a single interface*.

To perform the same operations with previous versions of Oracle Data Integrator, the following artifacts would have been designed:

- A first interface to load and persist a temporary datastore with the data *From Files*.
- A second interface to load and persist a temporary datastore with the data *From HR*.
- A procedure to Union and load both these temporary datastores into the target.
- A Package to chain the two interfaces and the procedure into a sequence.

This feature accelerates the interface design and reduces the number of interfaces needed to merge several data flows into the same target datastore.

Lookups

A wizard is available in the interface editor to design a lookup from a driving source datastore to a lookup datastore or interface. These lookups appear as a compact graphical object in the Sources diagram of the interface. The user can choose how the lookup is executed: as a Left Outer Join in the **FROM** clause or as an expression in the **SELECT** clause (in-memory lookup with nested loop). This second syntax is frequently more efficient on small lookup tables.

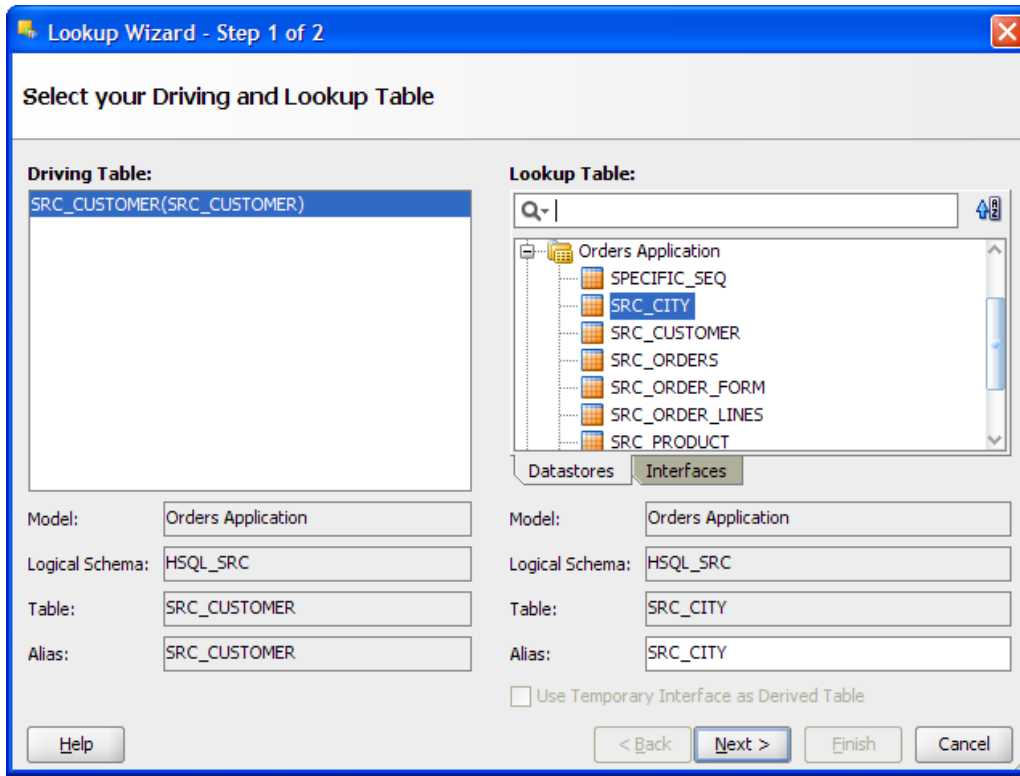


Figure 13 - Lookup Creation Wizard

Previous version of Oracle Data Integrator handled Lookups as regular Joins. This feature accelerates the design and simplifies reading interfaces. It also allows optimized execution of these lookups.

Derived Select for Temporary Interfaces

When using a temporary interface as a source in another interface, it is possible to avoid persisting the target datastore of the temporary interface by generating instead a Derived Select (sub-select) statement corresponding to the execution of this temporary interface. Consequently, the temporary interface no longer needs to be executed to load the temporary datastore.

In addition, the code generated for the sub-select can be optionally customized in an IKM.

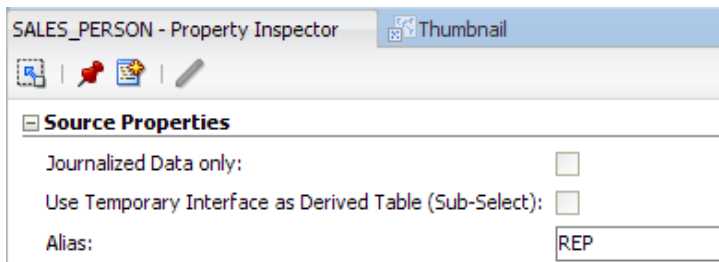


Figure 14 - Use Temporary Interface as Derived Table option can be checked when an interface is used as a source.

This feature eliminates the need for complex packages to handle sequenced loading of temporary interfaces. It dramatically simplifies the execution of cascades of temporary interfaces.

Other E-LT Features

Partitioning

Oracle Data Integrator now supports partitioning features of the data servers. Partitions can be reverse-engineered using RKMs or manually created in models. When designing an interface, it is possible to define the partition to address on the sources and target datastores. Oracle Data Integrator code generation handles the partition usage syntax for each technology that supports this feature.

Support for Native Sequences

Oracle Data Integrator 11g provides support for a new type of sequences that directly maps to database-defined sequences. Such a sequence is selected from a list retrieved from the database when the sequence is created. Native Sequences are used as regular Oracle Data Integrator sequences, and the code generation automatically handles technology-specific syntax for sequences.

This feature simplifies the use of native sequences in all expressions and enables cross-referencing on native sequence usage.

Support for Natural Joins

Oracle Data Integrator now provides support for Natural joins, defined at technology level. A natural join does not require any join expression to be specified, and is handled by the engine that processes it. This engine matches automatically columns with the same name.

Automatic Temporary Index Management

When creating joins or filters on source tables, it is possible to have Oracle Data Integrator automatically generate temporary indexes for optimizing the execution of these joins or filters. The user simply selects the type of index from a list for the given technology. Knowledge

modules automatically generate the code for creating and dropping the temporary index as needed.

This feature accelerates join and filter execution and enables better performances for integration interfaces.



Figure 15 – Two temporary Indexes (Non-Unique and Unique) will be created for optimizing the execution of the join between SRC_CUSTOMER and AGE_GROUP.

Conclusion

Oracle Data Integrator 11g introduces a new architecture, as well as features for dramatically improving productivity in data integration projects. A new IDE based on JDeveloper accelerate efficiency at design-time. Core features, such as datasets, lookups, derived select increase again development productivity and run-time performance while preserving the key product differentiators: Declarative Design, Knowledge Modules, Hot-Pluggability, and E-LT.



Maximizing Development Productivity with
Oracle Data Integrator 11g
September 2010

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
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2010, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

0109