An Oracle White Paper
February 2009

The Oracle Data Integrator Enterprise Edition Architecture

Disclaimer

Introduction

Oracle Data Integrator Enterprise Edition is built on several components all working together around a centralized metadata repository. These components—graphical modules, runtime components, and a Web interface—in conjunction with other advanced features, make ODI-EE a lightweight, legacy-free, state-of-the-art integration platform. This technical brief describes the ODI-EE architecture in detail.

Today Oracle includes Oracle Data Integrator and Oracle Warehouse Builder Enterprise ETL, formerly an add-on option to Oracle Database, as the two components of ODI-EE. This paper focuses on the Oracle Data Integrator component architecture. Going forward, these products will merge into a single unified data integration technology platform. This strategy fully preserves any existing development investments of all Oracle data integration customers and will provide a seamless, easy upgrade path from the current components to the unified platform. Customers can safely choose either component as the basis for implementations today. For further advice on the implementation choices, please contact your Oracle sales representative

Architecture Overview

The ODI-EE architecture is organized around a modular repository, which is accessed in client-server mode by components—graphical modules and execution agents—that are written entirely in Java. The architecture also includes a Web application, Metadata Navigator, which enables users to access information through a Web interface.

Graphical Modules

The four graphical modules are Designer, Operator, Topology Manager, and Security Manager. These modules can be installed on any graphical platform that supports Java Virtual Machine 1.5 (J2SE), including Windows, Linux, HP-UX, Solaris, AIX, and Mac OS, among others.

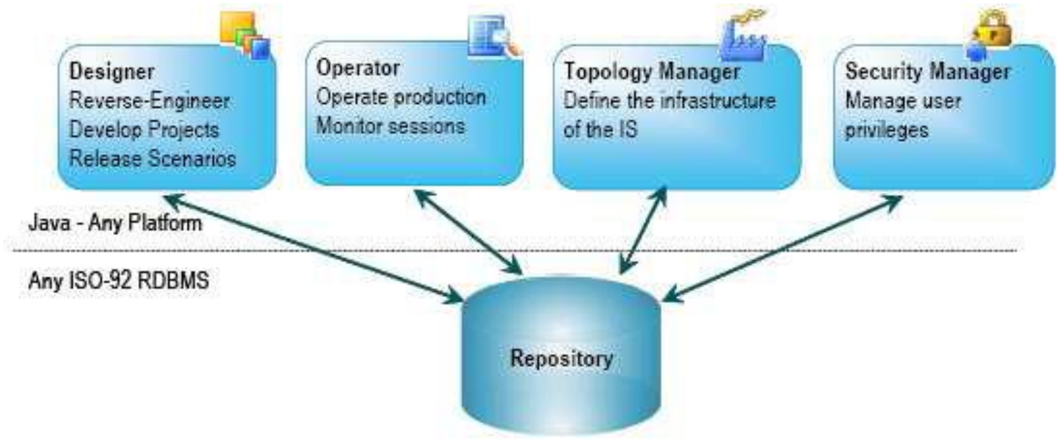

Figure 1. Graphical modules connect the repository.

The graphical modules are as follows:

- **Designer** defines declarative rules for data transformation and data integrity. All project development takes place in this module; this is where database and application metadata are imported and defined. The Designer module uses metadata and rules to generate scenarios for production. This is the core module for developers and metadata administrators.
- **Operator** manages and monitors production. It is designed for production operators and shows execution logs with error counts, the number of rows processed, execution statistics, the actual code that is executed, and so on. At design time, developers can also use the Operator module for debugging purposes.
- **Topology Manager** defines the physical and logical architecture of the infrastructure. The administrators of the infrastructure or project register servers, schemas, and agents in the master repository through this module.
- **Security Manager** manages user profiles and their access privileges. Security Manager can also assign access privileges to objects and features. Security administrators generally use this module. All modules store their information in the centralized repository.

Runtime Components

At runtime, the Scheduler Agent coordinates the execution of the scenarios. The Scheduler Agent can be installed on any platform that supports a Java Virtual Machine (J2SE), including Windows, Linux, HP-UX, Solaris, and IBM AIX, iSeries/AS400, zSeries/OS/390. Execution can be launched from one of the graphical modules, or the built-in scheduler or a thirdparty scheduler can trigger it.

With the Extract-Load Transform (E-LT) architecture, the Scheduler Agent rarely performs any transformation. It simply retrieves code from the execution repository and then requests database servers, operating systems, or scripting engines to execute that code. When the execution is completed, the Scheduler Agent updates the execution logs in the repository and then reports error messages and execution statistics. Users can view the execution logs from the Operator module or the Metadata Navigator Web interface.

It is important to understand that although the Scheduler Agent can act as a transformation engine, it is rarely used for that purpose. Agents are installed at tactical locations in the information system to coordinate the integration processes and leverage existing systems. They are multithreaded, load-balanced, lightweight components in this distributed integration architecture.

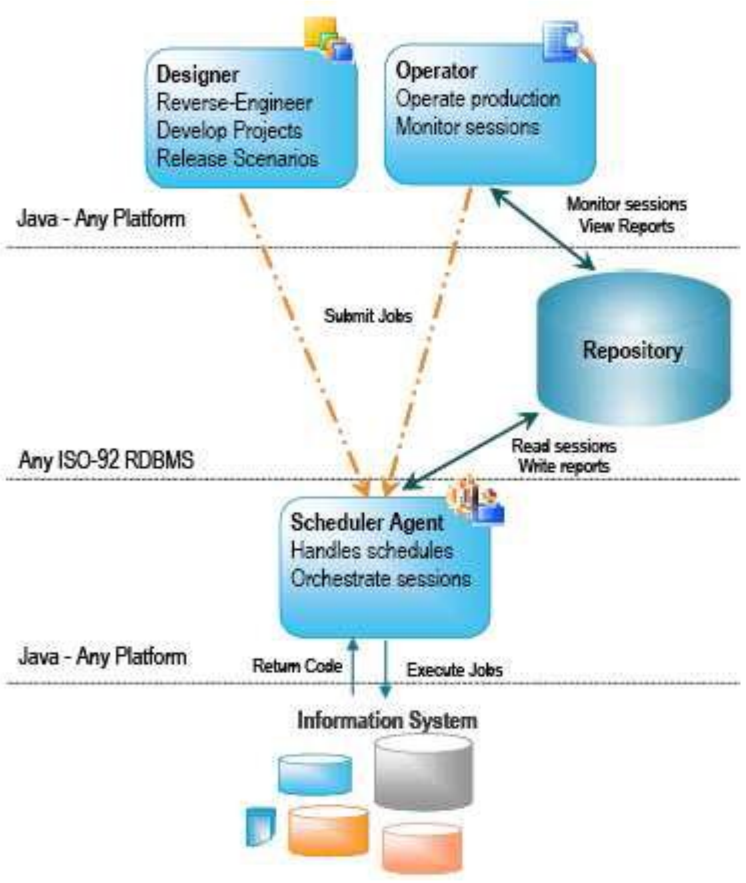

Figure 2. Runtime Components

Repositories

The repository consists of a master repository and several work repositories. These repositories are databases stored in relational database management systems. All objects that the modules configure, develop, or use are stored in one of these repositories, and are accessed in client-server mode by the various components of the architecture.

There is usually only one master repository, which contains the security information (user profiles and privileges), the topology information (definitions of technologies and servers), and the versions of the objects. The information contained in the master repository is maintained with Topology Manager and Security Manager. All modules have access to the master repository, because they all store topology and security information there.

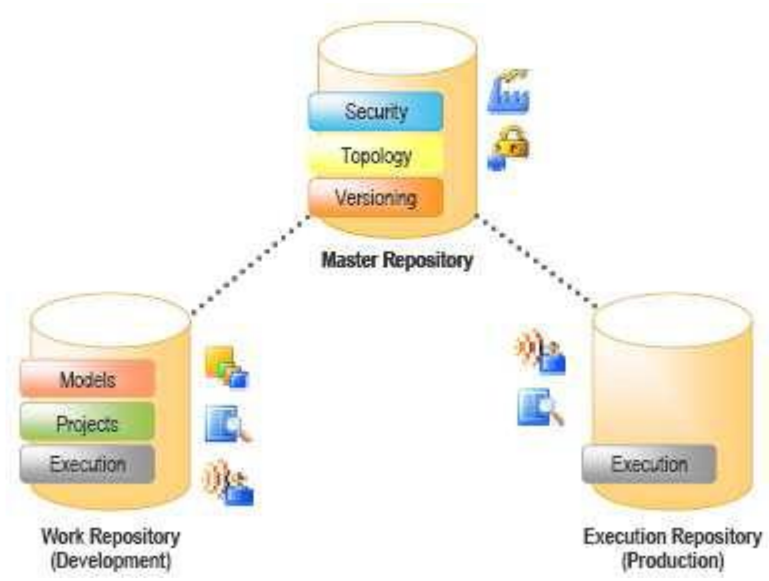

Figure 3. Master repository and work repositories.

Project objects are stored on the work repository. Several work repositories can coexist in the same installation. This is useful for maintaining separate environments or to reflect a particular versioning lifecycle—for example, development, qualification, and production environments.

A work repository stores information for

- Models—including datastores, columns, data integrity constraints, cross references, and data lineage
- Projects—including declarative rules, packages, procedures, folders, knowledge modules, and variables
- Runtime information—including scenarios, scheduling information, and logs

Users manage the content of a work repository with the Designer and Operatormodules. The Agent at runtime also accesses work repositories. When a work repository is used only to store execution information (typically for production purposes), it is called an execution repository. An execution repository is accessed at runtime with the Operator interface and by Agents. It is important to remember that all work repositories are always attached to one and only one master repository.

Metadata Navigator

Metadata Navigator is a Java 2 Enterprise Edition (J2EE) application that provides Web access to repositories. It allows users to browse objects, including projects, models, and execution logs. Metadata Navigator can be installed on an application server such as Oracle Container for Java (OC4J) or Apache Tomcat. Business users, developers, operators, and administrators can access Metadata Navigator via a Web browser. Through its comprehensive Web interface, users can see flow maps, trace the source of all data, and even drill down to the field level to understand the transformations used to build the data. They can also launch and monitor scenarios from a Web browser through Metadata Navigator.

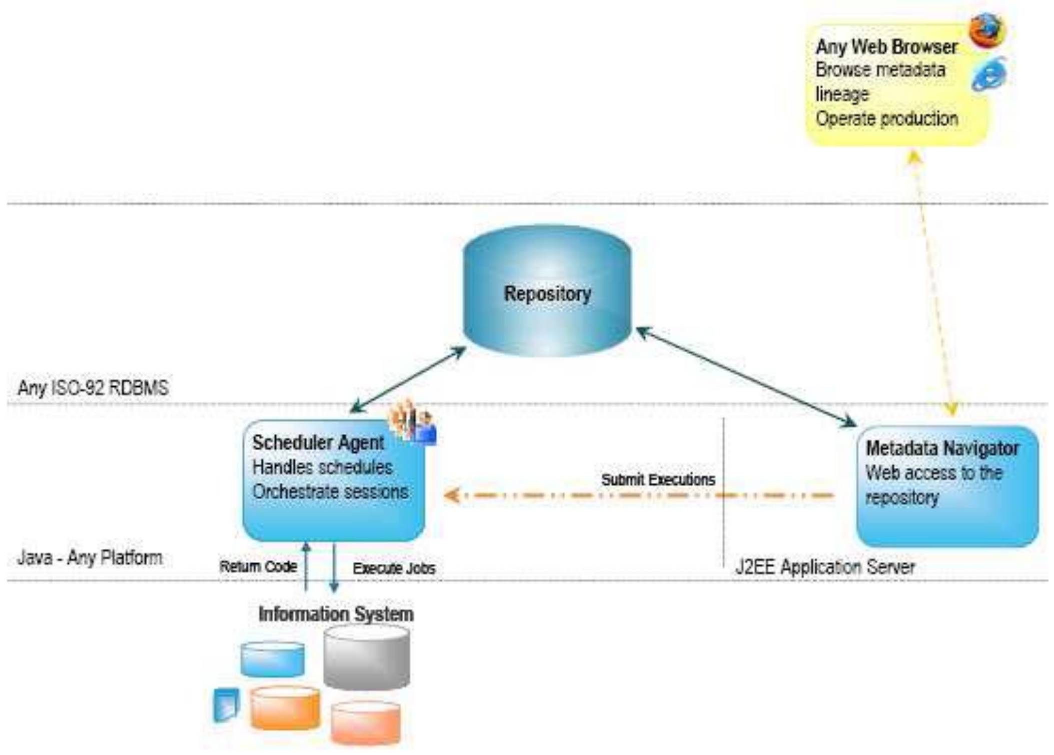

Figure 4. Through Metadata Navigator, users can access and execute metadata from a Web browser.

Other Components and Features

The ODI-EE also includes the following optional components and features:

- Knowledge modules make it possible to quickly and easily integrate technologies, databases, and applications. They exist for a large range of platforms, including Oracle, Teradata, Sybase IQ, Netezza, SAP/R3, Oracle Applications, Siebel, LDAP, and XML.
- The Advanced Parallel Option with Load Balancing feature enables large volumes of data to be processed by automatically balancing the workload between several Agents.
- Advanced version management provides an interface to manage, safeguard, and replicate revisions of units of work, even in the largest development environments.
- The Common Format Designer (CFD) feature lets users quickly design or assemble a data model from other data models, and then automatically generate the processes to load and extract data to and from this model. Users can, for example, use Common Format Designer to create operational datastores, datamarts, or master data canonical format by assembling heterogeneous sources. It can also be used to design a data warehouse model (for example, Star or Snowflake Schema, 3NF).
- The Publish and Subscribe Changed Data Capture (CDC) feature tracks changes in source data and reduces the volume of processed data by extracting only the changed data.
- The Publish and Subscribe Messaging feature provides the ability to use a message-oriented middleware (MOM) to implement an asynchronous, eventdriven integration architecture.

Conclusion

ODI-EE is a lightweight, legacy-free, state-of-the-art integration platform. All components can run independently on any Java-compliant system. Because of this legacy-free architecture, ODI-EE installs within minutes on any platform.

The Oracle Data Integrator Enterprise Edition
Architecture
February 2009

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

0109