

Oracle Warehouse Builder 10g Runtime Environment, an Update

An Oracle White Paper
February 2004

Runtime Environment, an Update

Executive Overview	3
Introduction	3
Runtime in warehouse builder 9.0.3	5
Runtime Overview (version 9.0.3 or earlier).....	5
Mapping Configuration Properties (version 9.0.3 or earlier)	6
Process Flow Execution (version 9.0.3 or earlier)	7
Deployment (version 9.0.3 or earlier)	8
Runtime Audit Viewer (version 9.0.3 or earlier)	9
Conclusion	9
Runtime in warehouse builder 10g.....	11
Installing the runtime components (version 9.0.4 or later)	11
Setting up the users	11
Locations and Connectors (version 9.0.4 or later).....	12
Locations	12
Connectors	14
Advantages	16
Mapping Configuration Properties (version 9.0.4 or later).....	16
Process Flow Execution (version 9.0.4 or later)	16
Deployment (version 9.0.4 or later)	17
Runtime Audit Browser (version 9.0.4 or later)	19
Adapter-based Deployment (version 9.0.4 or later)	21
Conclusion.....	21

Runtime Environment, an Update

EXECUTIVE OVERVIEW

To manage software implementation and maintenance is difficult. To manage a Business Intelligence system that integrates data from multiple sources is even more difficult.

Oracle Warehouse Builder provides the concepts and tools to manage the design and deployment of your Business Intelligence system. On top of that, a powerful architecture provides all you need to fully manage your system.

INTRODUCTION

Oracle Warehouse Builder is the only enterprise business intelligence integration design tool that manages the full life cycle of data and metadata for the Oracle Database. I can use Oracle Warehouse Builder to build relational and dimensional structures as well as data flows and process on top of these. Once I am happy with my definition I can generate the code to implement the structures in an Oracle database. The generator will take advantage from the latest and greatest features in the Oracle database. Also, I can easily share metadata via import and export mechanisms as well as via full published public APIs on top of Warehouse Builder. Generated code is deployed into a runtime environment.

The runtime environment is the area where you manage your day-to-day system operation. Rather than looking into multiple applications, analyzing multiple log files and accessing multiple log tables in order to find out the status of the system today, you want a consolidated view. Warehouse Builder provides that consolidated view. The HTML-based runtime audit browser enables you to view audit details from any PC, but low-level, fully supported, SQL access to the runtime data is available as well.

The Warehouse Builder 10g release features an enterprise-level runtime platform. This paper introduces the Warehouse Builder runtime platform versus the formerly used concept. The formerly used concepts are described first, highlighting those areas where the changes have taken place.

Readers that are new to Warehouse Builder can start with the section about the 10g version¹. To know what used to be provided is less useful for those users.

¹ The concepts described in this paper related to Warehouse Builder 10g also apply to versions 9.0.4 and 9.2.

RUNTIME IN WAREHOUSE BUILDER 9.0.3

The runtime environment in the Warehouse Builder versions 9.0.3 and earlier were very loosely coupled with the design environment. This section describes some of the relevant aspects of the runtime environment in those releases.

Runtime Overview (version 9.0.3 or earlier)

The runtime assistant is part of the Warehouse Builder installation. Before you can start deploying ETL processes to the target environment, you have to run the runtime assistant. The runtime assistant will create a number of relational tables and a number of PL/SQL APIs on top of these tables that will be accessed by the generated code. In general the runtime assistant will run against the eventual target schema, even though that is not a requirement (to install into the same database is a requirement). The ETL processes, mappings, always have to run in the schema that is populated by the runtime assistant. In practice, this means that if you want to target multiple schemas in one database, you may create multiple target schemas as indicated in Figure 1. The architecture has advantages from a separation perspective. However, in order to retrieve audit data you would have to reconnect to different users over and over again.

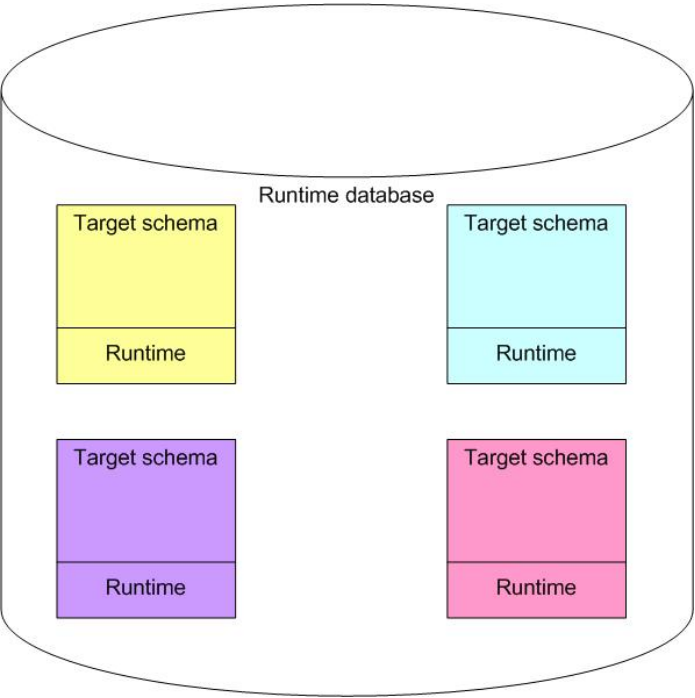


Figure 1. Multiple runtime schemas in a 9.0.3 or earlier release of Warehouse Builder.

Mapping Configuration Properties (version 9.0.3 or earlier)

Every mapping has its own configuration properties indicating physical pointers to sources and targets (see Figure 2). Warehouse Builder will do its best to assist you by defaulting to the use of database links, assuming a database link name the same as the name of the source module.

The default would often be changed, because the default strategy would not be followed.

- Existing database links with different names are used.
- Pointers to local schemas are preferably accessed through a <schema>.<object> notation instead of the default <object>@<db link> notation.
- Synonyms rather than direct pointers are used.
- Public instead of private database links are used.

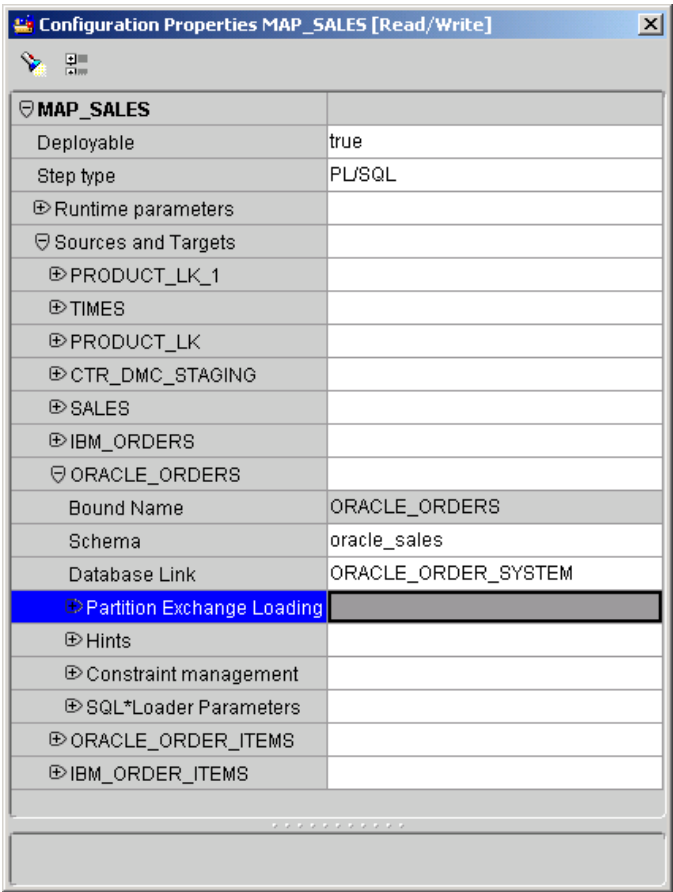


Figure 2. Mapping configuration properties in Warehouse Builder 9.0.3.

This approach is very flexible. Warehouse Builder allows users to point to any objects that had the same definition. However, in some cases you may end up spending a lot of time configuring each individual mapping to specify the access to

the sources and targets. Also, if you would want to deploy to another environment (e.g. you would move the test environment into a production situation) you would have to use exactly the same names for these configuration parameters you entered before. The amount of flexibility almost makes the solution inflexible.

Process Flow Execution (version 9.0.3 or earlier)

In Warehouse Builder 9.0.3 or earlier you can use the Workflow Deployment wizard to create activity (process step) definitions in an Oracle Workflow engine. Every activity is defined as an external job that will be executed by Oracle Enterprise Manager. Use Workflow Builder to define the dependencies and a workflow process is ready to be executed (see Figure 3).

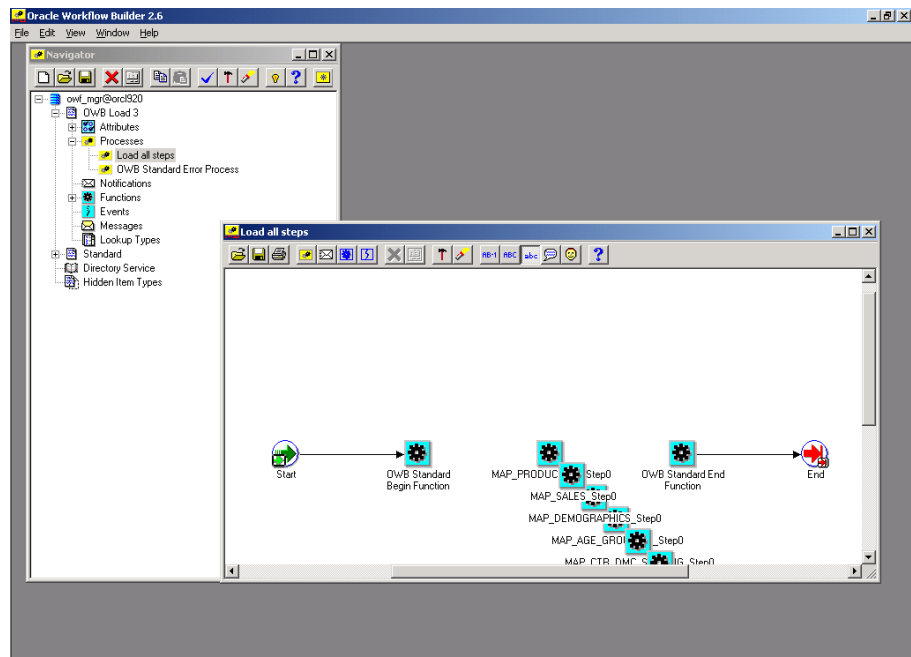


Figure 3. Definitions as these end up in Oracle Workflow.

In order to start the Workflow process you have to have a component called the Workflow Queue listener. This listener is installed as part of the Warehouse Builder design time installation. On the target module in Warehouse Builder you have to configure the path to the workflow queue listener. The workflow queue listener can be started manually or via the job in Oracle Enterprise Manager that can be created as part of the Workflow Deployment wizard.

In either case, the workflow queue listener will walk through the process flow and delegate the execution of every individual job to Oracle Enterprise Manager's Management Server. Management Server must have the individual jobs for the mappings defined in order to be executable (see Figure 4).

Job Library		
Job Name	Owner	
MAP_PRODUCT_LK_job	SYSMAN	
MAP_AGE_GROUP_LK_job	SYSMAN	
MAP_DEMOGRAPHICS_job	SYSMAN	
MAP_CTR_DMC_STAGING_job	SYSMAN	
MAP_CHANNELS_job	SYSMAN	
MAP_GEOGRAPHICS_job	SYSMAN	
MAP_COUNTRY_LK_job	SYSMAN	
MAP_PRODUCTS_job	SYSMAN	
MAP_SALES_job	SYSMAN	

Figure 4. The Enterprise Manager job library has to contain all executable mappings.

Figure 5 provides an overview of the Workflow Queue listener architecture. In the most complex scenario the Workflow server and queue listener would be installed on one machine, Oracle Management Server on another machine and the target database on yet another machine. The configuration is flexible though, which practically means that all components can be installed on the same server.

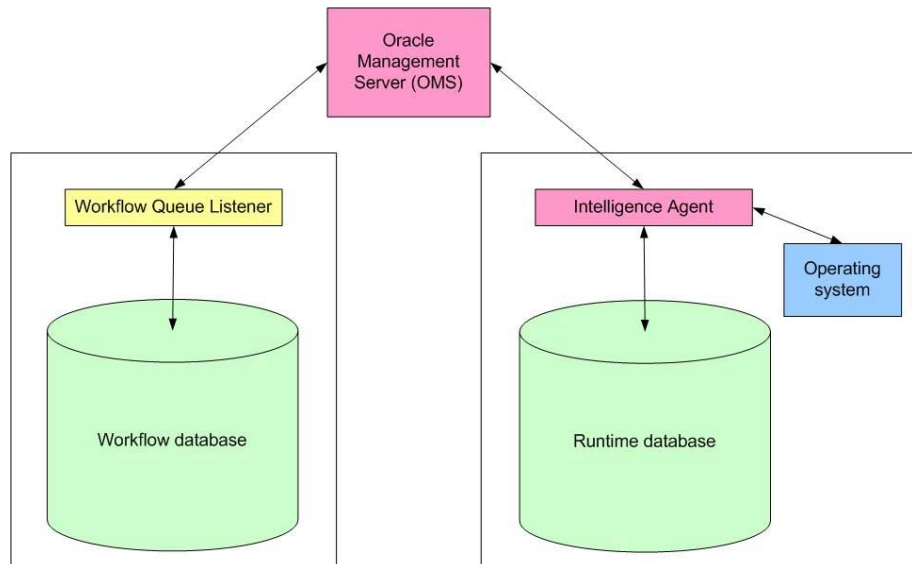


Figure 5. Workflow queue listener architecture.

The entire solution, relying on correct execution of the Workflow Queue listener, Oracle Management Server and one or more Intelligent Agents is quite complex and needed improvement.

Deployment (version 9.0.3 or earlier)

Every time you want to deploy an object you have to generate the code for the object (or multiple objects at the time) and use the deploy button to deploy. For performance reasons Warehouse Builder will reuse generated code knowing the definition that resulted in the code has not changed. Upon deployment a connection dialog will pop up asking you into which schema you want to install the

object. Warehouse Builder will remember the connect information you entered the last time, but you will be required to re-enter the information each time you restart Warehouse Builder.

Multi deployment, because of the larger number of schemas, requires a little more administration:

- What is the connect information for the target schemas (e.g. development, test, QA, production)?
- Keep track of the current status of deployment in each of the environments.
- Make sure to align the configuration between the different environments to make sure that no changes have to be made between the different environments.

Runtime Audit Viewer (version 9.0.3 or earlier)

You use the Runtime Audit Viewer to evaluate the run of the individual mappings. The viewer is a Java client side component that is installed as part of the Warehouse Builder product (see Figure 6). I.e. if you want to view the audit results from a specific computer you have to install the entire product on that computer.

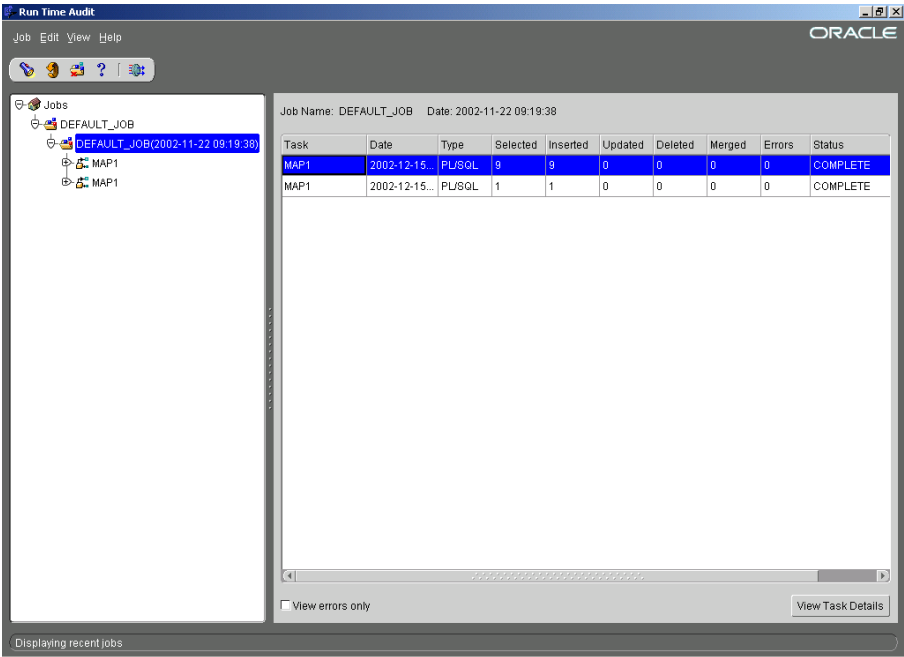


Figure 6. The Runtime Audit Viewer.

Conclusion

From a design perspective, the configuration, generation and deployment model in Warehouse Builder 9.0.3 and earlier is very flexible. However, the flexibility could

lead to difficulties in resolving the exact metadata definitions because of the flexibility.

From a runtime perspective, to store runtime audit data in multiple schemas provides a clear separation, but does require users to reconnect to multiple schemas to get hold of their data.

Both on the design and on the runtime side there was room for improvement. Oracle Warehouse Builder 10g addresses a number of the improvement areas. The improvements were introduced in the 9.0.4 release and existed in the 9.2 release also.

RUNTIME IN WAREHOUSE BUILDER 10G

The 10g version of Oracle Warehouse Builder features new concepts such as so-called locations and connectors, a new deployment framework called the deployment manager and an HTML-based runtime audit browser that replaces the Java client side component. The architecture has changed: deployment actions are stored and the runtime statistics are kept in one location. The new architecture also introduces an adapter-based framework, which is extensible and open to other Oracle tools as well as partner or other vendor solutions.

Installing the runtime components (version 9.0.4 or later)

The runtime platform for Warehouse Builder 10g requires software outside the database. This Warehouse Builder server-side software, has to be installed in a separate Oracle home. The runtime install has to run on the machine that will contain the target database structures.

Setting up the users

Once the server side software has been installed the runtime assistant automatically pops up from the Oracle Universal Installer. You have to go through the first steps of the runtime assistant in order to register the physical location of your runtime server software in the OWBRT_SYS schema. The Warehouse Builder runtime assistant will create this schema. Once the runtime assistant has run from the server once (e.g. after the installation of the server-side software) you can run the assistant subsequently from the client.

The runtime assistant guides you through the creation of:

- The runtime repository owner. The runtime repository owner schema contains all auditing tables and a number of packages that the runtime platform needs in order to manage deployment and execution of objects. The runtime repository owner requires a lot of system privileges in order to install and execute objects in another schema. Typically, not even Warehouse Builder users need to know the password of this user.
- A runtime access user. The runtime access user can provide access to the runtime repository and the services in the runtime repository without having all the system privileges that have been granted to the runtime repository owner. The grants and privileges are managed by database roles. One runtime repository can grant access to multiple runtime access users. To create a runtime access user is optional, but it would be the way to shield the runtime repository password from Warehouse Builder users.
- A target schema. The target schema contains a number of packages and procedures that Warehouse Builder installs out-of-the-box. Also, the target schema is granted access to the runtime repository services needed to log audit messages during execution. Many target schemas can be related to one runtime repository.

In the current 10g release all schemas have to be installed in the same database. Remote target schemas are not supported. Figure 7 shows a schematic overview of the target server.

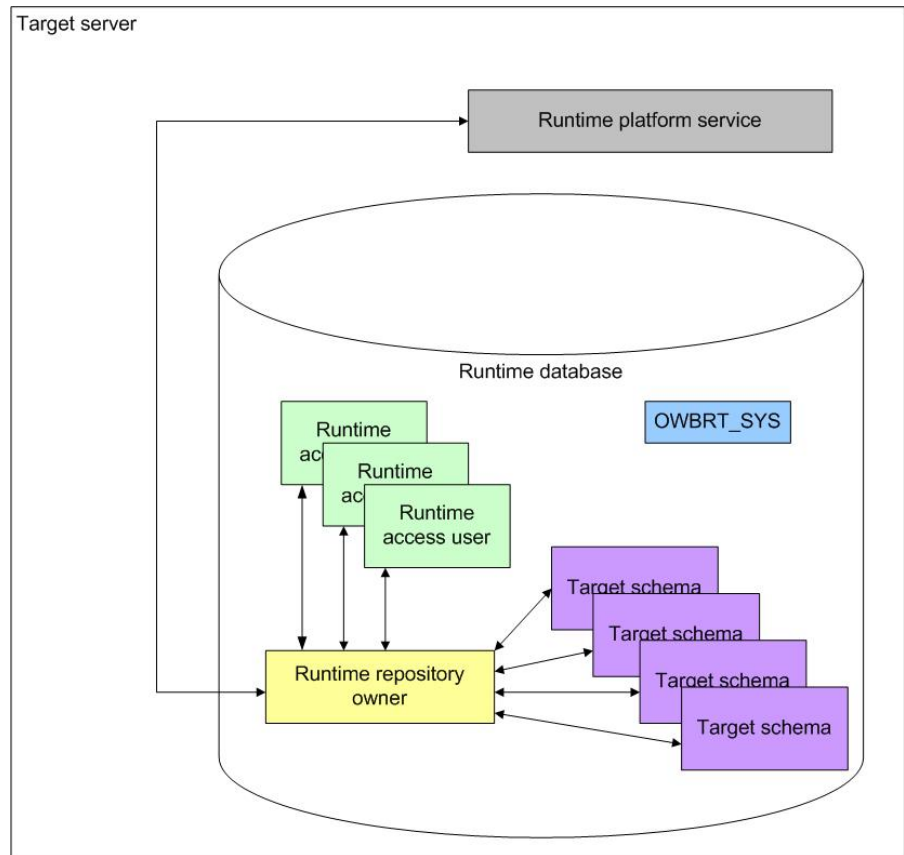


Figure 7. Schematic overview of the target server and runtime platform.

Locations and Connectors (version 9.0.4 or later)

Warehouse Builder 9.0.4 introduces the concept of locations and connectors. This new concept provides insight in the physical deployment and provides flexibility for multi-deployment scenarios.

Locations

A location is created in the design client. Named locations are available as a separate node where you create a module (see Figure 8). For example, a location to represent an Oracle database schema is created under the Databases, Oracle node. Currently there are database locations, flat file locations, SAP application locations and process flow locations.

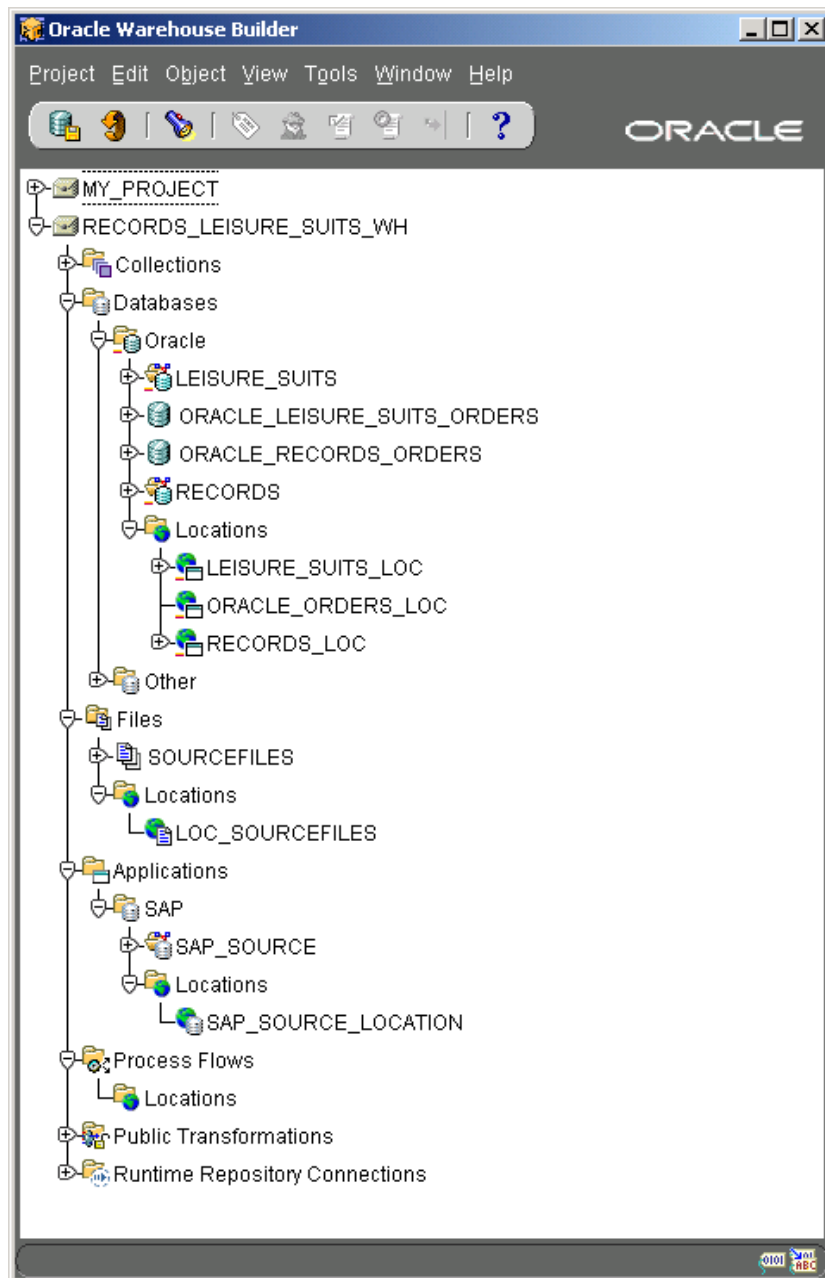
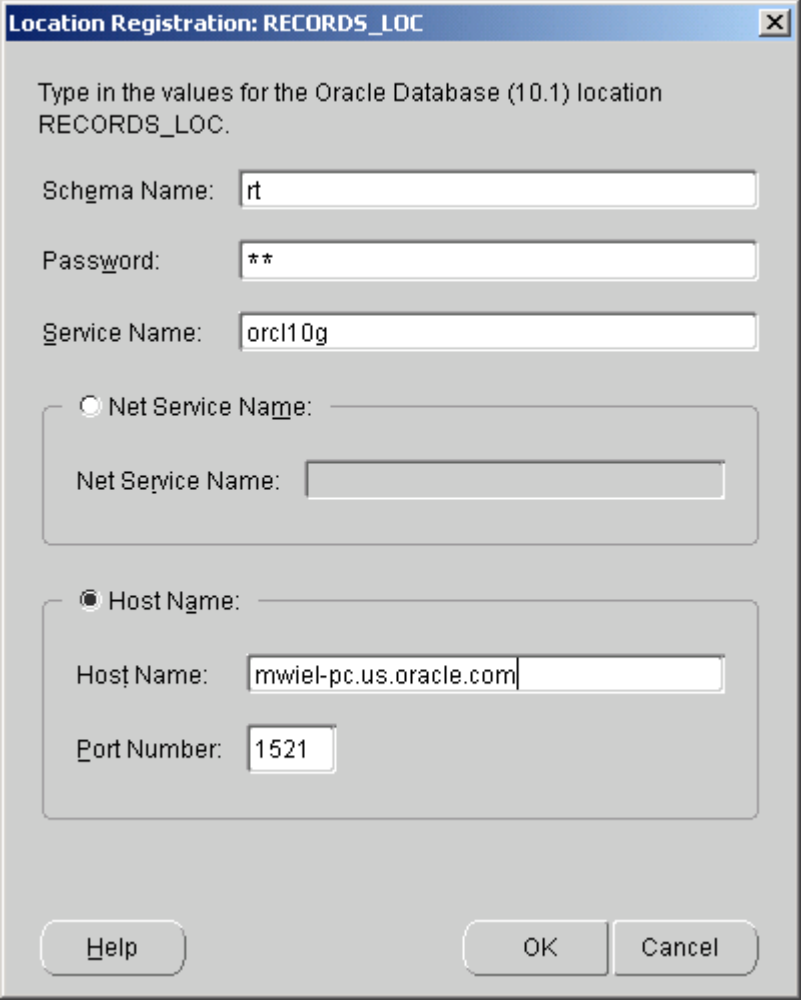


Figure 8. Locations under multiple nodes in the design tree.

In the design client the location is a logical placeholder. The location indicates that upon deployment there is a physical place that is represented by the location in the design client. The physical characteristics are registered in the runtime repository when:

- You first want to deploy an object that uses that location.
- You explicitly register the location in the deployment manager.

Implicit or explicit registration is done using a location registration dialog that requests location specific parameters. Figure 9 shows the location registration for an Oracle database location.



The image shows a Windows-style dialog box titled "Location Registration: RECORDS_LOC". Inside the dialog, there is a text label: "Type in the values for the Oracle Database (10.1) location RECORDS_LOC." Below this, there are several input fields: "Schema Name:" with the value "rt", "Password:" with the value "**", and "Service Name:" with the value "orcl10g". There are two radio button options: "Net Service Name:" (which is unselected) and "Host Name:" (which is selected). Under "Net Service Name:", there is an empty text field. Under "Host Name:", there are two fields: "Host Name:" with the value "mwiel-pc.us.oracle.com" and "Port Number:" with the value "1521". At the bottom of the dialog, there are three buttons: "Help", "OK", and "Cancel".

Figure 9. Location registration for an Oracle database location.

In the design environment a module is related to a location. As a module represents a logical grouping of objects multiple modules can be related to the same location.

Connectors

The existence of a connector indicates that at runtime there is a way (channel) to move data from one location to another location. The connectors are defined in the context of a location (see Figure 10). A connector will eventually be deployed to the target schema. The result of the generation can be:

- No physical object, but generated code would get a schema reference rather than a database link name. Note that the reference would use <schema>.<object> notation, which requires select privileges on the source objects.

- A database link in case the connector points from an Oracle database location to another location that represents a database location.
- A directory object in case the connector points from an Oracle database location to a flat file location. The directory object is necessary in order to be able to create external tables.

Currently connectors can only be created in the context of an Oracle database location. Only one connector can be created to point from one location to a specific other location.

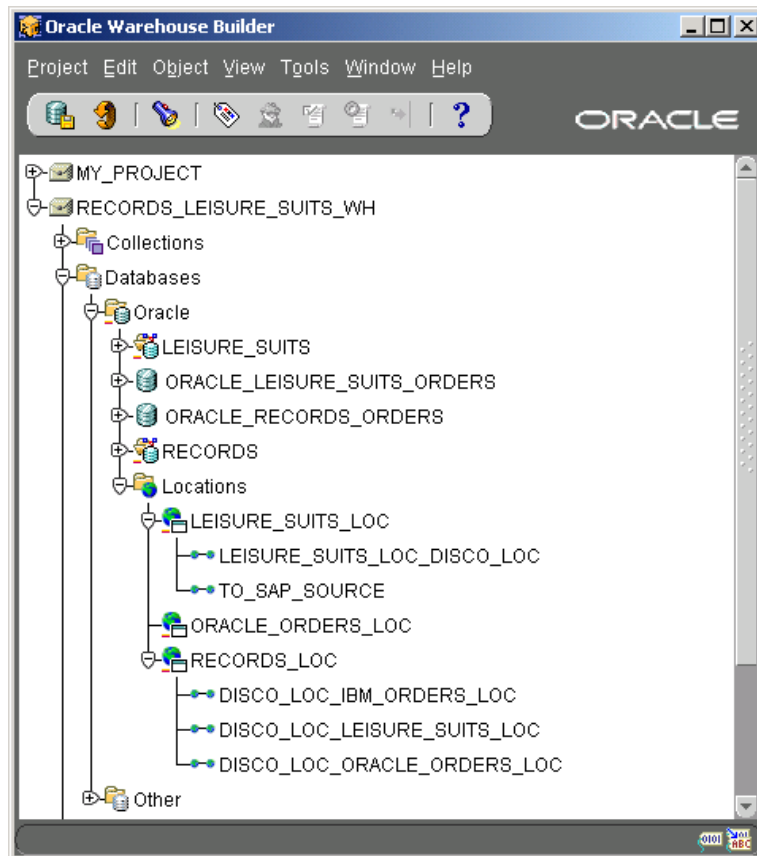


Figure 10. Connectors owned by a location.

Connectors can be created explicitly or implicitly. Warehouse Builder will implicitly create a connector between 2 locations if:

- You create a mapping and use a table from a module that is related to another location than the module that contains the mapping.
- You create an external table. A connector will be created between the flat file module you selected as the default location for the external table data file and the location that is related to the Oracle module that contains the external table definition.

Advantages

The concept of locations and connectors and the way Warehouse Builder treats those provides a number of advantages:

- The concept supports a multi-deployment scenario that allows physical differences without any impact on the design. As the design does not store physical location characteristics locations could be local or remote without any design changes. Also, the mapping configuration does not need to be changed in the multi-deployment scenario.
- You do not have to remember the physical connection details of the target environment each time you deploy to the target schema. The connection details are stored in the runtime repository.

Mapping Configuration Properties (version 9.0.4 or later)

The concept of locations and connectors implies that the physical resolution of the actual objects can be resolved at deployment time. I.e. you do not need to specify any configuration properties anymore to indicate how to access objects in a mapping. The generated code will contain references to the locations that are resolved by the deployment services when you deploy the mapping.

The physical configuration properties database link and schema name that you specified in Warehouse Builder version 9.0.3 and lower are still there for backward compatibility. The Warehouse Builder 10g release is the last version that will support these parameters. Warehouse Builder will display validation warnings if you still specify values for those parameters. The generated code will:

1. Use the configured parameters if values have been specified.
2. Use the location and connector concept if the modules have locations associated.
3. Assume local objects in any other case.

Process Flow Execution (version 9.0.4 or later)

Warehouse Builder 10g also features a process flow editor in the design environment. The editor enables you to fully define a process flow definition in Warehouse Builder and deploy the whole flow into the execution engine (see Figure 11).

The deployment manager provides an interface that shows the physical grouping of objects. I.e. the location is the primary entry point for the tree on the left hand side (see Figure 12).

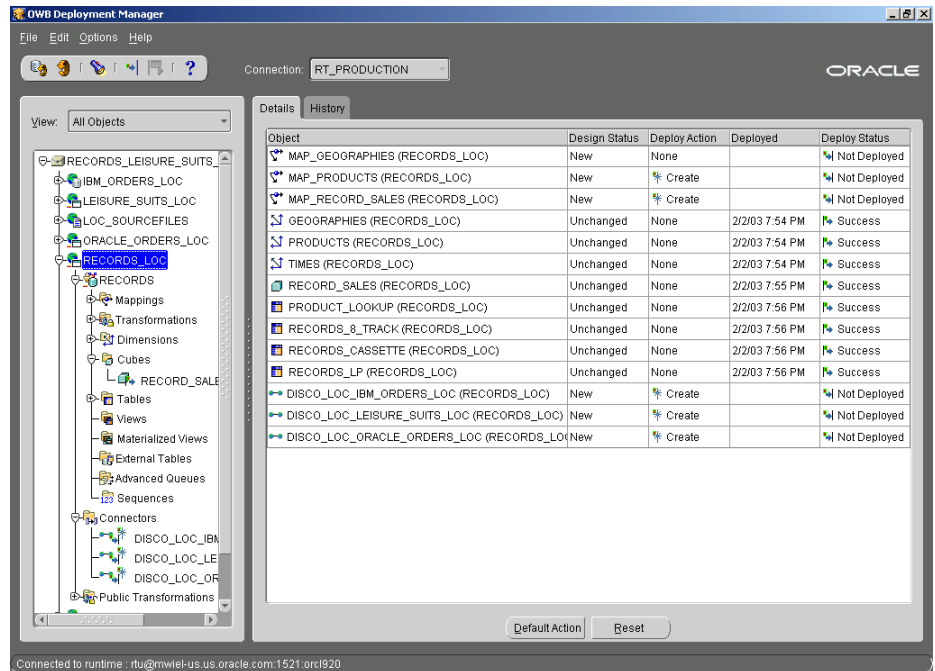


Figure 12. The deployment manager interface.

The right hand side shows the objects you have selected explicitly or implicitly indicating the design status, deploy action, the date/time of the last deployment as well as the deploy status. The Default Action button at the bottom determines the most appropriate deploy action for the object you selected:

- Create if the design status is new.
- Upgrade or replace (dependent on whether upgrade is supported for the object) if the design status is changed.

You can always overwrite or explicitly set the deploy action. Drop is also supported as a deploy action.

Warehouse Builder 10g also keeps track of the deployment audit history. The History tab provides insight in past deployments (unless the deployment audit information was purged). See Figure 13.

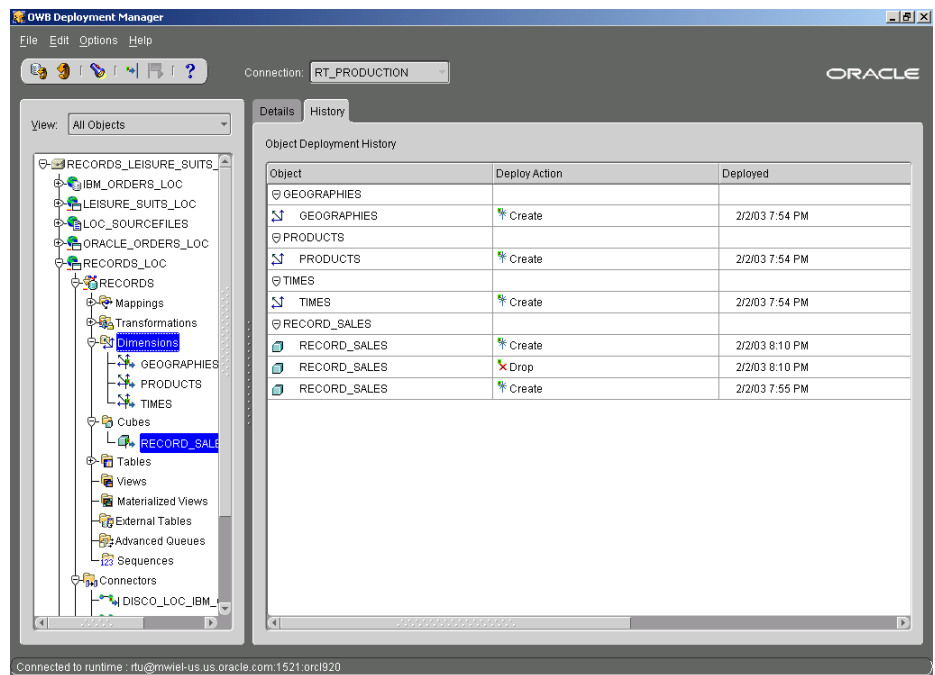


Figure 13. Deployment history in the deployment manager.

The Connection toggle at the top of the screen enables you to switch from one runtime repository connection to another runtime repository connection. This feature enables you to support a multi-deployment scenario and manage multiple runtime platforms that have implemented the same design.

As the deployment history is stored in the runtime repository you can read the exact deployment status based on the connection you selected. This advantage is in line with support for a multi-deployment scenario.

Runtime Audit Browser (version 9.0.4 or later)

A Runtime Audit Browser in Warehouse Builder 10g has replaced the Java Runtime Audit Viewer from version 9.0.3. The HTML-based viewer provides access from any client browser without the need to install the Warehouse Builder product. The runtime audit browser provides access to both the deployment history (see Figure 14) as well as the execution details (see Figure 15). You can purge the audit details via the runtime audit browser interface when you are logged on as a user with 'QA user' privileges. This is also the only default role that provides access to the error records in the runtime audit.

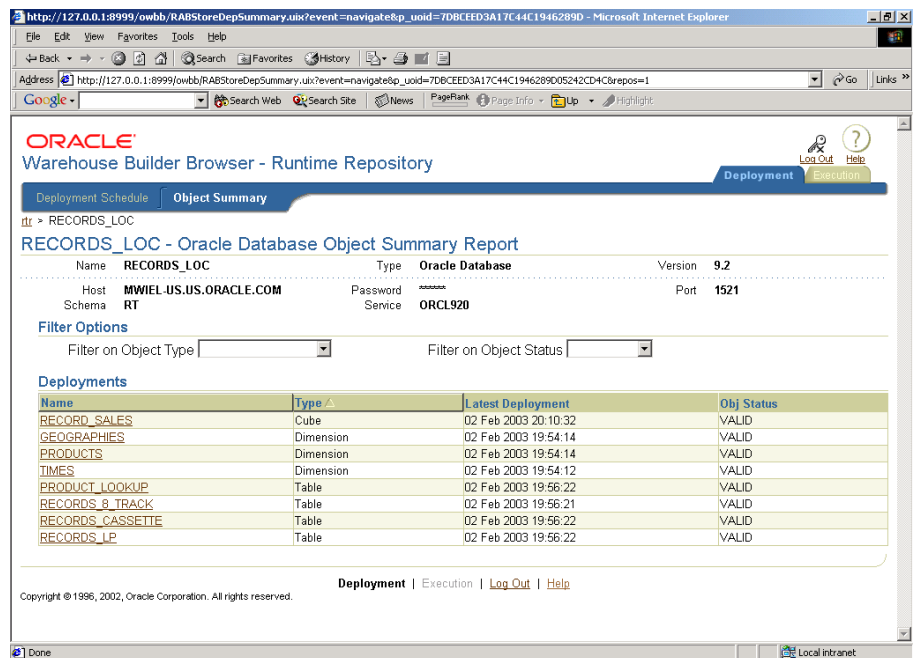


Figure 14. Deployment history in the runtime audit browser.

The runtime audit browser can be installed on an Oracle Application Server instance, in which case the solution is fully compatible with single sign on and the multi user capabilities of Oracle Application Server. You can also run the runtime audit browser from a machine that has the Warehouse Builder client installed. In that case you will not leverage single sign on nor will you get multi user capabilities. You have to run the local OC4J to invoke the runtime audit browser standalone.

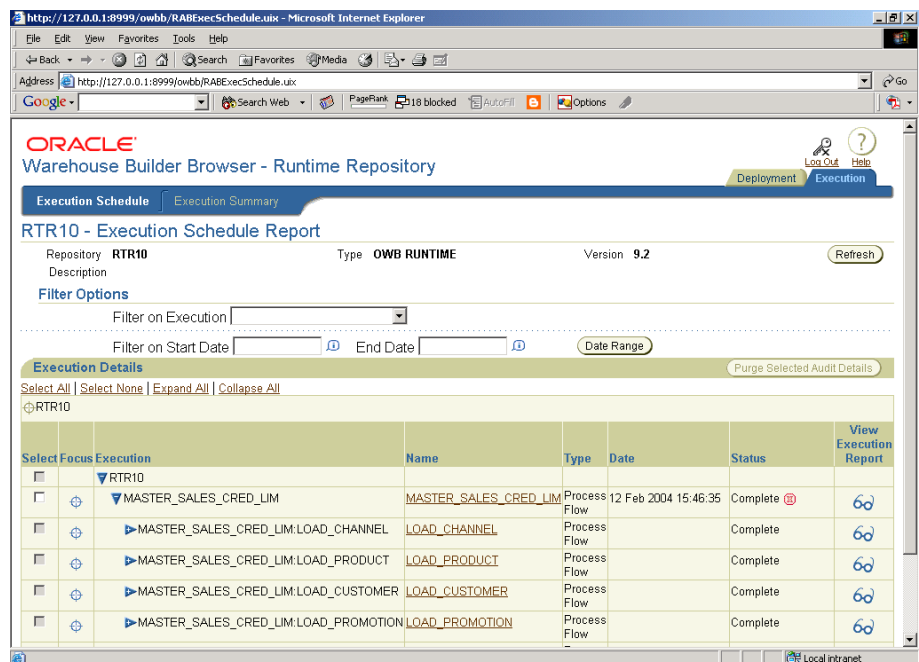


Figure 15. Execution overview in the runtime audit browser.

Adapter-based Deployment (version 9.0.4 or later)

Adapter-based deployment means that Warehouse Builder generates a specific kind of code, preferably industry standard code, that is not necessarily subject to many changes. For every tool implementation (or version) a specific adapter would be written that translates this generic code into code that is understandable by the target tool.

The adapter-based approach in Warehouse Builder 10g provides extensibility for deployment and execution adapters. Expect both partners (not restricted to Oracle) as well as Warehouse Builder development to add adapters in the future to support more deployment targets.

CONCLUSION

Management of a Business Intelligence system that aggregates data from multiple sources is extremely complex. Oracle Warehouse Builder release 10g provides a enterprise level runtime platform that enables you to carefully manage this environment. The changes have resulted in a runtime environment that is:

- Flexible because of the introduction of the location and connector concept.
- Consolidated because of the central Runtime Repository that oversees the entire deployment system.
- Manageable because of the new deployment manager, the extensive auditing capabilities and the HTML-based runtime audit browser.
- Easier to maintain because it does not necessarily require an entire Oracle Enterprise Manager installation and Workflow Queue listener installation/configuration.
- Better performing because of the support for native execution.
- Extensible because of the adapter-based architecture.



Oracle Warehouse Builder 10g Runtime Environment, an Update
February 2004

Author: Mark Van de Wiel

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

Oracle is a registered trademark of Oracle Corporation. Various product and service names referenced herein may be trademarks of Oracle Corporation. All other product and service names mentioned may be trademarks of their respective owners.

Copyright © 2004 Oracle Corporation
All rights reserved.