OWB2ODI Converter

Migrating from OWB to ODI
Step-by-Step Overview

Kobee

Table of contents

Management Summary

In its latest Statement of Direction, Oracle announced that **OWB would not be supported anymore** beyond release 12 of the Oracle database. Therefore, all OWB installations need to be migrated before 2017.

"Future database releases beyond Oracle Database 12c Release 1 will not be certified with OWB 11.2."
[Quote by Oracle]

Oracle Data Integrator (ODI) is Oracle's new strategic product for high-performance, flexible and heterogeneous data integration.

The main benefit of a conversion to ODI, is that the basic OWB functionality remains available in ODI. So, by migrating your OWB project to ODI, you won't lose your previous investments in OWB.

To make sure that this conversion process runs as smoothly as possible, with a minimum of manual intervention, RedBridge offers a complete OWB to ODI conversion service using its intelligently automated OWB2ODI conversion tool.

After an initial analysis of your OWB .mdl export file and a joint assessment meeting, the OWB project, including its metadata, will be converted to ODI, tested at RedBridge's Services Centre, customized where required, tested in your own environment and, finally, successfully rolled out to production.

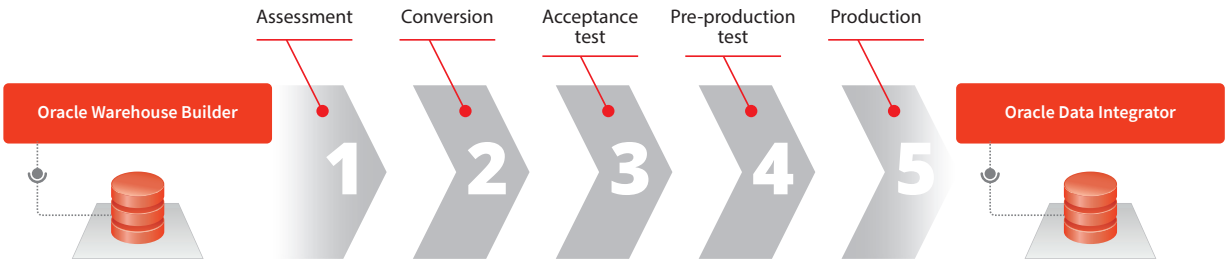
Our service is fast and complete, and works for all OWB and ODI versions as well as for all O/S platforms. It includes the migration, the reworking and the customization of the OWB mappings and process flows in ODI, for a fixed timeframe and a fixed price.

In the remaining of this document, we will describe more in detail the different steps of the conversion process.

WAREHOUSE BUILDER → DATA INTEGRATOR

The Conversion Process in Five Steps

The conversion process is split up into 5 main steps:

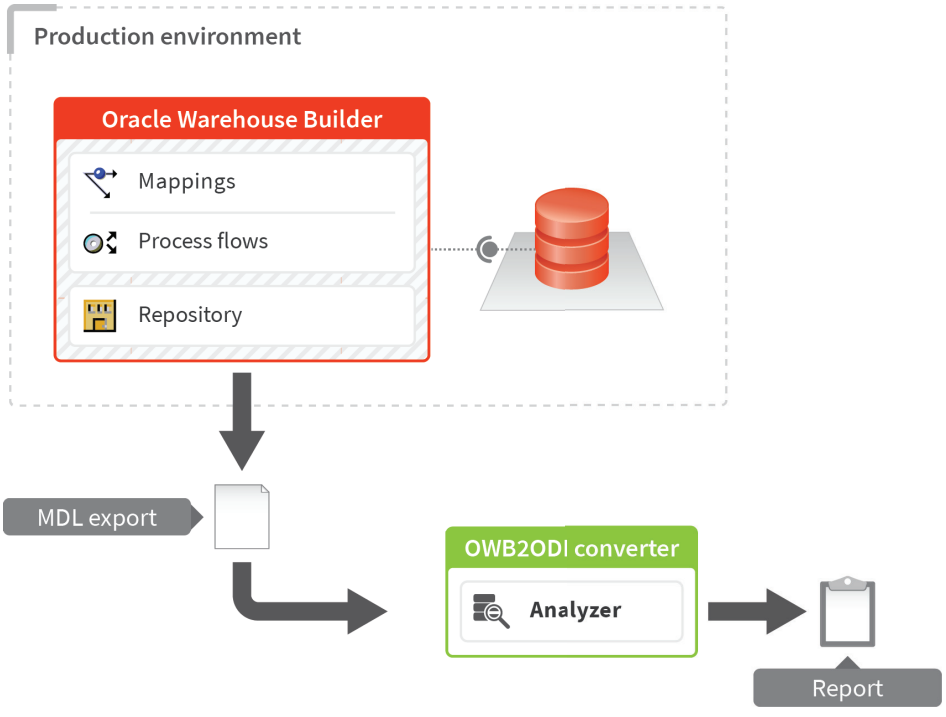


Before we can start we need the client to provide us with an OWB .mdl file export. This .mdl file export must contain all project Locations, the Configuration (if it is different from default), all mappings and process flows in validated state, and all dependencies.

Step 1: Assessment

The objective of this first step is to define the project's framework, to evaluate the consistency of the original OWB project and to define the conversion criteria. This step can be subdivided into the following tasks:

Task 1: Generate a Conversion Assessment and Statistics Report
Task 2: Decide on exceptions handling
Task 3: Define the topology
Task 4: Compare the possible conversion modes
Task 5: Define the Knowledge Modules (KM)
Task 6: Decide on configuration management



Task 1 – Generate a Conversion Assessment and Statistics Report

The Conversion Analyzer will use the provided .mdl file to perform a series of operations.

First of all, it will analyze whether there are any special operators for which essential customizations are required. For example, in ODI 11g consider the case of the splitter operator that allows the flow to direct itself towards several targets in the same mapping. This operator has to be carefully managed because the Oracle Data Integrator does not allow a mapping flow to end in multiple targets. In this case, it generates as many interfaces as the Oracle Warehouse Builder operator targets. Note that ODI 12c does not have this problem.

Next, it will analyze each mapping, starting from the target operator and ending on the leftmost operator, ensuring the entire mapping structure and its related operators remain unchanged. When the analysis is finished, the Conversion Analyzer will generate a report on the OWB process flows and mappings.

ACTIVITY_TYPE	AMOUNT	FIRST_FOLDER_NAME	SECONDARY_FOLDER_NAME	PF_NAME	ACTIVITY_PER_PF	PF_STATE
OR	7.942	Folder_A	Folder_1	Process flow A1.1	10	LOW
TRANSFORMATION	5.995	Folder_A	Folder_1	Process flow A1.2	6	LOW
MAPPING	2.199	Folder_A	Folder_1	Process flow A1.3	35	LOW
SET_STATUS	1.930	Folder_A	Folder_1	Process flow A1.4	24	LOW
SUBPROCESS	1.505	Folder_A	Folder_1	Process flow A1.5	13	LOW
AND	1.217	Folder_A	Folder_1	Process flow A1.6	42	LOW
ASSIGN	332	Folder_A	Folder_1	Process flow A1.7	14	LOW
FORK	252	Folder_A	Folder_1	Process flow A1.8	46	LOW
FILE_EXISTS	92	Folder_A	Folder_1	Process flow A1.9	18	LOW
USER_						LOW
WAIT						LOW
END_L						LOW
FOR_L						LOW
ROUTE						LOW
WHILE_						LOW
						LOW
						LOW
						LOW
						LOW
NI						MEDIUM
						MEDIUM
MA						LOW
						LOW
NUM_P						LOW
NUM_P						LOW
NUM_P						LOW

OPERATOR_TYPE	AMOUNT
TABLE	7.011
EXPRESSION	2.928
FILTER	1.590
JOINER	1.336
CONSTANT	687
INPUT_PARAMETER	630
AGGREGATOR	603
SET_OPERATION	447
SPLITTER	346
PREMAPPING_PROCESS	323
POSTMAPPING_PROCESS	183
SEQUENCE	151
FLAT_FILE	128
VIEW	89
DEDUPLICATOR	52
TRANSFORMATION	51
SORTER	4
TOTAL	16.559
NUM_MAPPINGS	1.000
MAX_NUM_OPERATORS_IN_MAPPING	50
NUM_MAPPINGS_LOW	400
NUM_MAPPINGS_MEDIUM	300
NUM_MAPPINGS_HIGH	300

MODULE_NAME	MAPPING_NAME	OPERATOR_PER_MAPPING	MAPPING_STATE
Module 1	Mapping 1.1	3	LOW
Module 1	Mapping 1.2	12	LOW
Module 1	Mapping 1.3	7	LOW
Module 1	Mapping 1.4	21	LOW
Module 1	Mapping 1.5	13	LOW
Module 1	Mapping 1.6	8	LOW
Module 2	Mapping 2.0	19	LOW
Module 2	Mapping 2.1	21	LOW
Module 2	Mapping 2.2	25	MEDIUM
Module 2	Mapping 2.3	32	MEDIUM
Module 2	Mapping 2.4	14	LOW
Module 2	Mapping 2.5	6	LOW
Module 2	Mapping 2.6	9	LOW
Module 2	Mapping 2.7	12	LOW
Module 2	Mapping 2.8	18	LOW
Module 3	Mapping 3.1	23	MEDIUM
Module 3	Mapping 3.2	34	MEDIUM
Module 3	Mapping 3.3	2	LOW
Module 3	Mapping 3.4	6	LOW
Module 3	Mapping 3.5	7	LOW
Module 3	Mapping 3.6	10	LOW
Module 3	Mapping 3.7	27	MEDIUM
Module 3	Mapping 3.8	16	LOW
Module 4	Mapping 4.1	11	LOW
Module 4	Mapping 4.2	4	LOW
Module 4	Mapping 4.3	4	LOW
Module 4	Mapping 4.4	30	MEDIUM
etc.	etc.		

TOP: Process flow report, **BOTTOM**: Mappings report

After the generation of the report, IKAN will set up a meeting with the client to discuss the generated Conversion Assessment and Statistics Report. Together, they will perform tasks 2 and 6 concerning the procedural and managerial decisions.

During this meeting, the client and IKAN will also compile the "Roles and Responsibilities Matrix" for each task within the conversion process. Tasks 2 to 6 cover the different decisions that need to be taken.

This meeting and the decisions taken are essential for the success of the conversion. They are the foundation upon which the entire conversion project will be built.

Task 2 – Decide on exceptions handling

This task is dedicated to deciding on how to manage (work-around or manual conversion) any possible case that cannot be automatically converted to ODI using the OWB2ODI Converter

For example, a particular OWB operator in a mapping, a particular OWB activity or a particular transition condition in a process flow, or an external workflow layer (not in OWB) used to execute the OWB project's components.

These are extremely rare contingencies, as most OWB components will be automatically converted.

Task 3 – Define the topology

In ODI, the definition of the topology, the logical architecture, and the physical architecture are necessary to indicate where the data are physically located.

The OWB2ODI Converter will analyze the OWB Repository Metadata to collect the following information: the technology, the machine where the data is located, and the database schema and/or file metadata.

ODI was developed to operate with any database technologies available (Oracle, DB2, Teradata, SQL Server, and many others). ODI provides the ability to refer to a database schema/user by its logical form. The logical user or logical schema is an abstract reference to a physical schema, which is defined within ODI as a real user in a specific database technology. Logical schemas and physical schemas are related through contexts. For example, the BUDGET logical schema may be associated with the BUDGET physical schema in the ORCL database through the CTX_BUDGET_ORCL context, etc. All this is configured in ODI's topology section, which contains all the information needed to switch between the logical and physical side.

The OWB2ODI Converter only works on logical schemas because the pointers to physical schemas are configured in the topology, and the context is assigned either at runtime or when "conversions" are executed. The result of that process shows how easy it is to generate a technology-independent code. As long as the right context is set up, everything will work properly.

Note: a logical schema is always associated with its own technology, although it can be easily moved to another one by deleting it from the old platform and building it on the new one.

Task 4 – Compare the possible conversion modes

This task is dedicated to providing the client with a detailed explanation of the conversion mode of every OWB component.

The OWB2ODI Converter can convert the following:

- For ODI 11g: OWB mappings into ODI interfaces and ODI packages
- For ODI 12c: OWB mappings into ODI mappings and ODI packages
- OWB process flows logic into ODI native tools, ODI packages, ODI procedures and ODI load plans (available since ODI v.11.1.1.5).

Task 5 – Define the Knowledge Modules

ODI's capabilities to handle any RDBMSs are represented by its Knowledge Modules (KMs).

The OWB2ODI Converter has a specific console to set parameters and options related to the use of these KMs. **This is a very important task, as the correct definition of these KMs will affect the performance of the ODI project.**

The following example shows how the OWB2ODI Converter can be customized in order to choose the KMs best suited to your technology and needs.

Knowledge Module	Source Tech	Target Tech	Type of loading	Default Operating Mode
Oracle SQLLDR	ORACLE	ORACLE	INSERT	SET BASED
Oracle Incremental Update	ORACLE	ORACLE	UPDATE	SET BASED
IKAN Custom KM	ORACLE	ORACLE	INSERT/UPDATE	SET BASED
DSNTIAUL	DB2	DB2	TRUNCATE/INSERT	SET BASED
Client Custom KM	DB2	DB2	INSERT/UPDATE	SET BASED FAIL OVER ROW BASED
…	…	…	…	…

Example of a customized KMs selection grid

Customized KMs can also manage other aspects such as: ANALYZE on target table, the maximum number of allowable errors on target table, loading HINT, selecting HINT, …

Task 6 – Decide on configuration management

The last subtask of this step, but certainly not the least important one, is to decide on the operative protocol to be applied during the conversion period.

To obtain best results, it would be opportune to freeze any maintenance activity on the OWB project to be converted. The ideal scenario would be to:

- roll out every OWB project undergoing modification into the client's production environment,
- align each of the client's environment (development, test and production), and
- freeze any maintenance activity.

The client has to decide if freezing maintenance activity is possible or not. If not, IKAN needs to know how the client intends to execute configuration management of the OWB project to be converted.

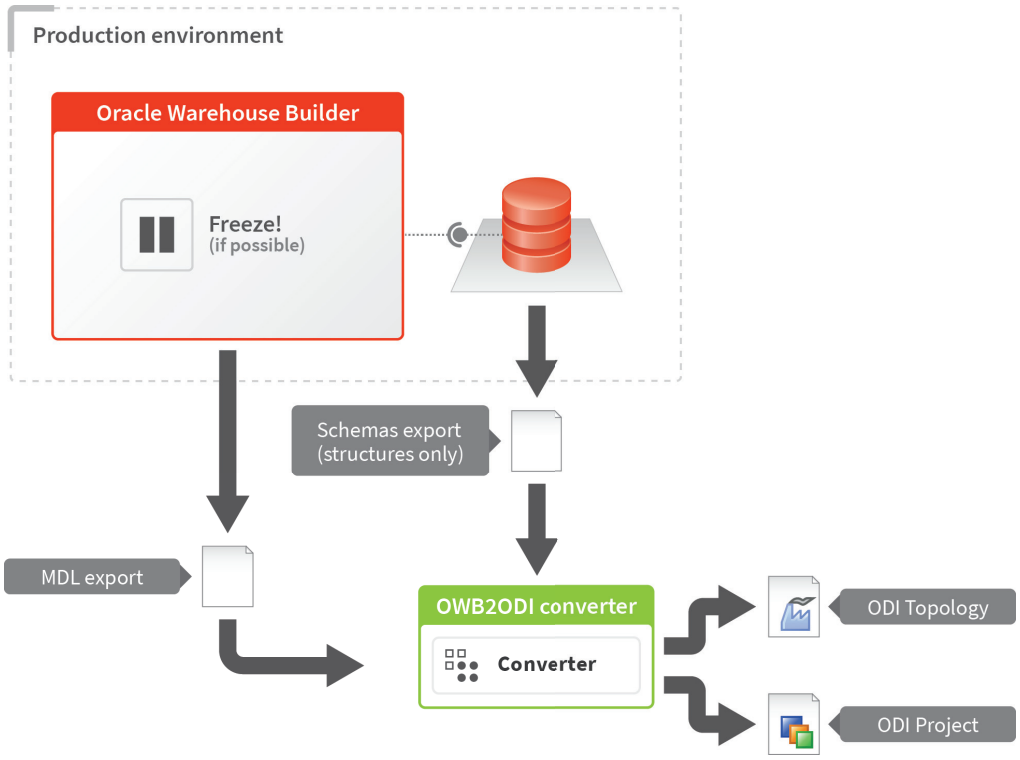
Also during this task, the "fixing protocol", which is the management process (roles, responsibilities, actions, etc.) needed to fix any bugs identified during the acceptance test task, must be defined.

Step 2: Conversion

The second step is the actual conversion step. The objective of this step is to convert the original OWB project into a new ODI project and to generate the new ODI project's metadata to be imported into the client's ODI repository.

This step can be subdivided into the following tasks:

Task 7: Convert the OWB Mappings and OWB Process Flows
Task 8: Execute formal tests
Task 9: Generate the ODI project's metadata



Task 7 – Convert the OWB Mappings and OWB Process Flows

This is the core task of the entire conversion process and it is executed by IKAN's Services Centre.

Before starting the task, the following should be carried out:

- supply a definitive .mdl file aligned with the latest OWB project version,
- supply a DB schema export aligned with the latest OWB project version, and
- freeze any maintenance activity related to the OWB project to be converted.

Based upon the decisions taken concerning the topology (Task 3 "Topology definition"), IKAN's Services Centre will carry out the ODI topology setting in its internal conversion environment.

Next, all mappings will be analyzed and a first transformation will be applied whenever specific operators are found. This is an essential process in order to guarantee the proper functioning of the new Oracle Data Integrator flow, while keeping the semantic flow unchanged.

This task is performed automatically by the OWB2ODI Converter. Carrying out this process manually, would not only be costly and time-consuming, but also the risk of introducing new mistakes due to haste or technical misunderstandings would be very high.

Once the additional mappings are "normalized", recursive techniques are used to generate the operator's tree of each mapping. The tree is redrawn and each operator involved is transformed according to the new Oracle Data Integrator semantic. The correct topological information is maintained, taking into account the possible overruling of the location using a database of links or different schemas.

From a conceptual point of view, OWB and ODI are similar, but they do not have equivalent features and have profound and significant differences. Therefore, the OWB2ODI Converter does not handle some of the OWB components because they are rarely used, because they do not have a corresponding ODI function or because the conversion would be too complex, ineffective or inefficient.

The following table indicates which OWB components are automatically converted by the OWB2ODI Converter and which ones need to be converted manually.

OWB mapping: converted operators		
Aggregator	Joiner	Sorter
Constant	Mapping input/output parameter Mapping	Splitter
Deduplicator (Distinct)	Match-Merge	Table
Expression	Materialized View	Transformation
External Table	Pivot/Unpivot	View
Filter	Pre/Post mapping process	Anydata Cast (since 11.1)
Flat File (File multi-record)	Sequence	
Key Lookup	Set Operation	Table Function

OWB mapping: unhandled operators		
Dimension	Expand object	Pluggable mapping
Cube	Varray iterator	Queue (11.2)
Construct	Name and address	Subquery filter (11.2)
Data generator		LCR cast/splitter (11.2)
OWB process flow: converted activities		
Data auditor	FTP	And / Or
Mapping	Manual	End
Subprocess	Notification	End Loop
Transform	Set status	For Loop
Assign	SQLplus	While Loop
Email	User-defined	Fork
File Exists	Wait	Route
OWB process flow: converted transition conditions		
Success	Error	Complex
OWB process flow: unhandled activities		
Web services (11.2)	EJB / Java class (11.2)	OMB plus (11.2)
OWB process flow: unhandled transition conditions		
Warning*	Extended	

Since ODI does not have a “warning status”, the OWB “warning transition condition” is handled depending on each client’s specific needs. This is a possible topic to be covered in the assessment meeting.

The OWB “Fork activity” is converted by using the ODI “Load plan” feature that is only available from ODI v.11.1.1.5 onwards. Therefore, in order to manage and convert all of the most used and useful activities in the Process Flow, we need an ODI version that is equal or greater than ODI v.11.1.1.5.

Task 8 – Execute formal tests

When the conversion is done, the first formal non-regression tests are executed at IKAN's Services Centre on empty data structures. At this stage, the test only concerns the format correctness of the new ODI project, as no data are available for a real test run.

Task 9 – Generate the ODI project's metadata

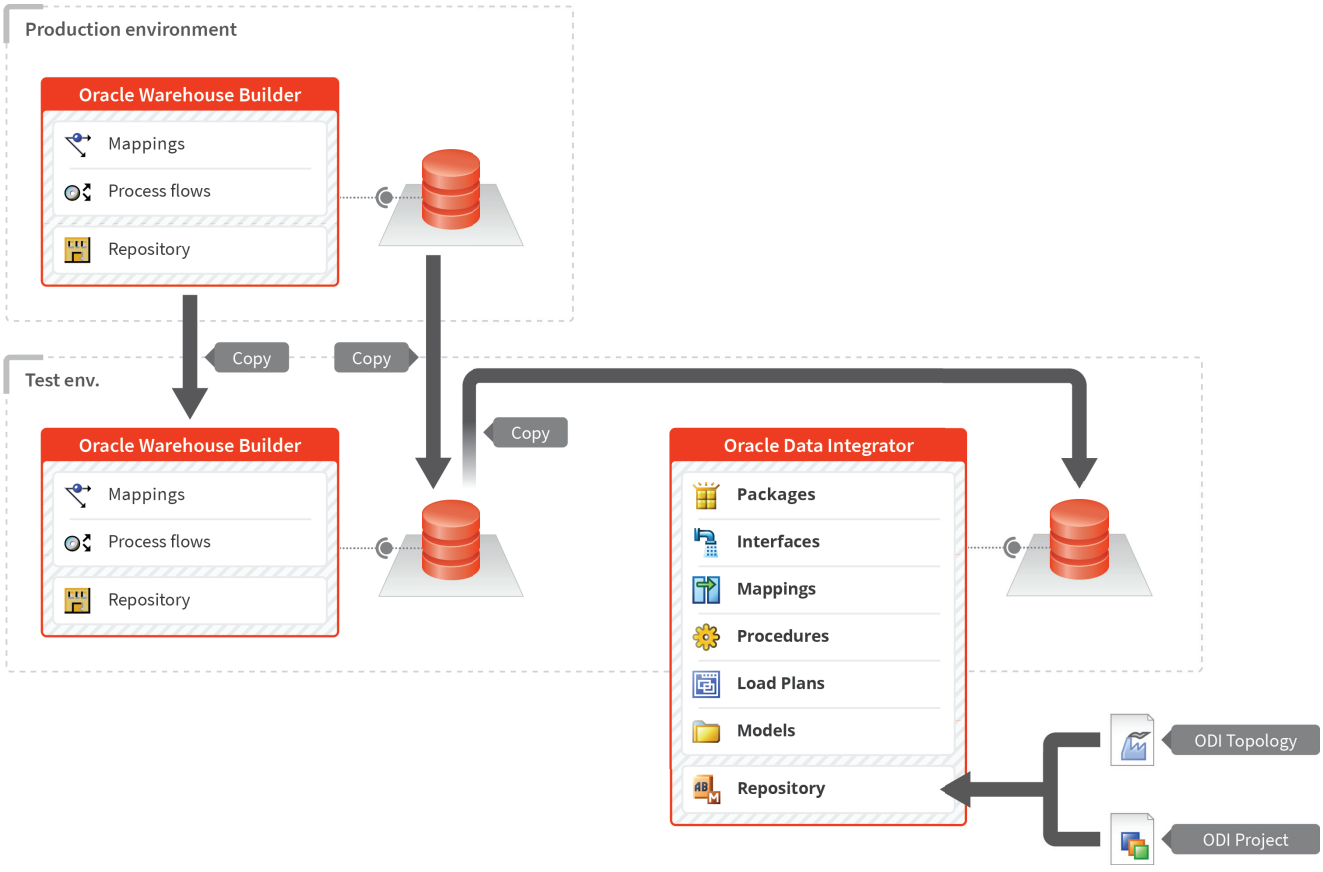
The last task of the conversion step is dedicated to the generation of the new ODI project's metadata. These metadata will be included in the .xml files that will be delivered to the client.

Step 3: Acceptance Test

Once everything has been tested successfully at IKAN, it is time to test the ODI mappings and load plan in the client's Test Environment. The objective of this step is to compare the results of the new ODI project results (in the Test Environment) with the results of the current OWB project (in the Production Environment) and to fine-tune the new ODI project's performances.

This step can be subdivided into the following tasks:

Task 10: Set up the ODI Test Environment
Task 11: Execute acceptance tests
Task 12: Tune the new ODI project's performance



Task 10 – Set up the ODI Test Environment

First the Test Environment needs to be set up. This includes:

- installing the OWB product,
- installing the ODI product,
- copying the OWB project's components,
- importing the .xml files supplied by IKAN's Services Centre into the ODI repository, and
- copying the production databases twice: one copy for the OWB project and another copy for the ODI project.

Task 11 – Execute acceptance tests

Next, the ODI project will be tested to ensure that there are no regression issues.

Concretely, a complete parallel run of the two projects will be executed and, next, the respective results will be compared to verify their matching. Each incorrect result will be investigated to determine the relative causes.

The "fixing protocol" established during the initial assessment meeting (Task 6 of the Assessment step) will be applied for each problem detected.

Task 12 – Tune the new ODI project's performance

The last task of this Acceptance Test step consists in fine-tuning the ODI project's performances in collaboration with the client's Database Administrator.

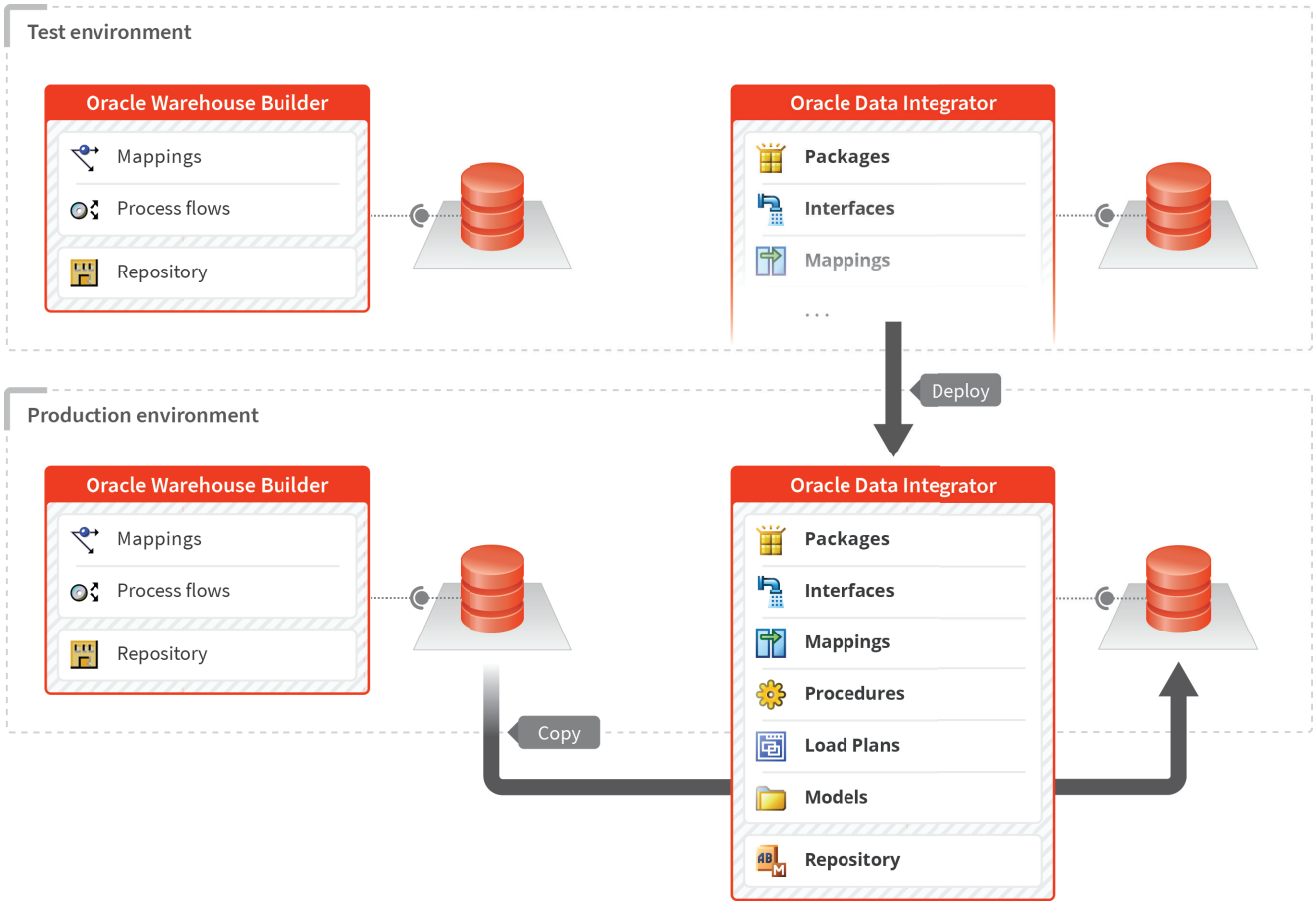
Step 4: Pre-production Test

If the tests in the Test Environment are successful, we will start the tests in a Pre-Production Environment.
The objective of this step is to ensure the same level of correctness in the Production Environment as in the Test Environment.

Also, the ODI project performances (in the Pre-production Environment) will be once again compared with the original OWB project (in Production environment). If required, additional fine-tuning will be done.

This step can be subdivided into the following tasks:

Task 13: Set up the (parallel) ODI Production Environment
Task 14: Execute additional verification tests
Task 15: Verify the performance



Task 13 – Set up the (parallel) ODI Production Environment

During this task, a parallel Pre-production Environment is set up.

Task 14 – Execute additional verification tests

Theoretically, this is a redundant step. However, experience teaches us that a parallel production run must be carried out in order to locate any possible problems that may affect the production environment (configuration, privileges, missing patches, etc.).

The outcome results of the two projects are compared once again.

Task 15 – Verify the performance

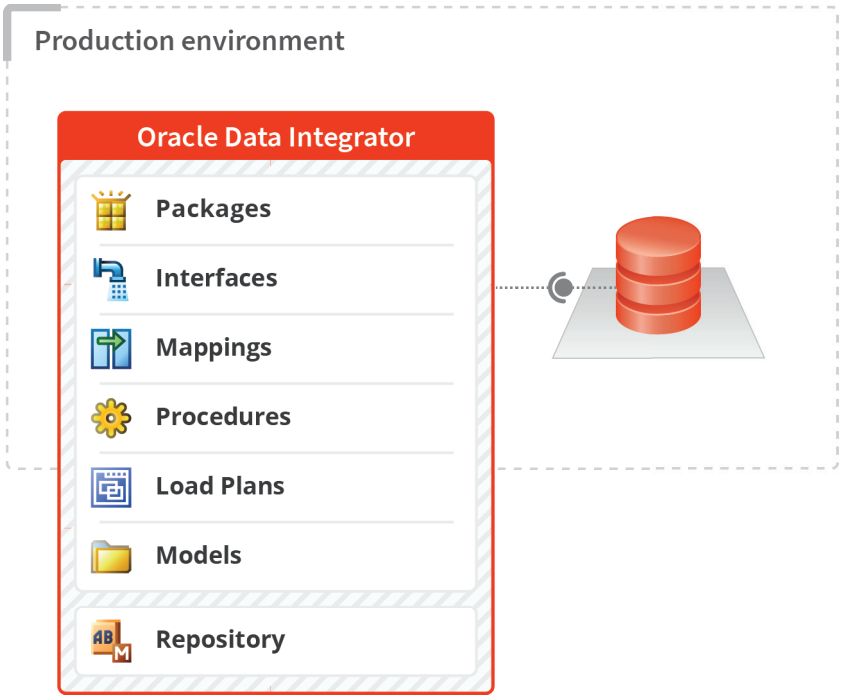
Theoretically, this is also a redundant task. But here again, experience teaches us that it is a good idea to also compare once again the performances.

Step 5: Production

In this final step, the original OWB project and its dependencies (OWB installation, DB schema, etc.) will be removed from the Production environment, and the scheduling tool will be switched from the original OWB project to the new ODI project.

This step can be subdivided into the following tasks:

Task 16: Clean the production environment
Task 17: Switch from OWB to ODI



Task 16 – Clean up the Production Environment

The cleaning consists of deinstalling OWB, deleting the DB schema and removing any other software components that are unnecessary to run the new ODI project.

Task 17 – Switch from OWB to ODI

Finally, the last task in the conversion process consists of physically switching from the old OWB project to the new ODI project.

Summary

As OWB is at the end of its lifecycle, a migration towards an alternative environment is required. To prevent losing your previous investments made in OWB, converting your existing OWB projects to ODI is an excellent choice.

Our full OWB2ODI conversion service ensures a smooth and accurate conversion. It works for all OWB and ODI releases, is customizable to your needs and covers the complete migration. On top of that it is fast (typically a few weeks) and it is done within a fixed timeframe and for a fixed price.

For more detailed information on the different steps of the Conversion process, feel free to:

- download the white paper from our website:
- send us your OWB mdl export file to receive a price and time quote or
- request a proof of concept based on a selected set of mappings and process flows.

For more information

https://www.kobee.io/company/contact
Call us: +32 15 797306

IKAN Development
Kardinaal Mercierplein 2
2800 Mechelen, Belgium
Tel. +32 15 797306

info@kobee.io
www.kobee.io