

# VCR4ODI User Guide

Release 4.0  
February 2019



# RedBridge

RedBridge Software BVBA  
Kardinaal Mercierplein 2  
2800 Mechelen  
BELGIUM

Copyright © 2019 RedBridge Software BVBA

No part of this document may be reproduced or transmitted in any form or by any means, electronically or mechanically, for any purpose, without the express written permission of RedBridge Software BVBA

---

# Table of Contents

## PART I - GETTING STARTED

<b>Chapter 1 - Introduction .....</b>	<b>1</b>
<b>Chapter 2 - Starting the VCR4ODI connector.....</b>	<b>2</b>
2.1. Oracle Data Integrator Tab Page .....	4
2.2. Version Control Repository Tab Page .....	5
2.3. Additional Buttons.....	8
<b>Chapter 3 - VCR4ODI Profiles.....</b>	<b>9</b>
3.1. Creating a New Profile .....	10
3.2. Editing an Existing Profile.....	12
3.3. Copying an Existing Profile .....	12
3.4. Deleting an Existing Profile .....	13

## PART II - FUNCTIONALITIES

Chapter 4 - Functionalities.....	15
Chapter 5 - Committing Elements .....	16
Chapter 6 - Restoring Elements.....	21
Chapter 7 - Locking/Unlocking Elements.....	26

## PART III - PROBLEM SOLVING

Chapter 8 - Debugging and Logging.....	30
Chapter 9 - Managing Conflicts .....	31
9.1. Commit Constraints.....	31
9.2. Restore Constraints .....	31

---

# GETTING STARTED

---

# Introduction

The VCR4ODI connector is a component connecting ODI repositories with Subversion. On the one hand it provides a view on your selected ODI Repository and on the other hand a view on the selected Subversion trunk or branch.

The VCR4ODI connector allows to

- work at the individual object level,
- lock objects to prevent others from working on the same objects you are working on,
- commit objects from the selected ODI repository to the selected Subversion trunk or branch,
- restore objects from a previous version.

On top of that, VCR4ODI is able to calculate all dependencies between objects, and, when committing, it will automatically create a new revision for Projects, Models, Scenarios and the global objects that are used.

This guide describes the usage of the VCR4ODI connector.

Other related documentation include the *VCR4ODI Installation Guide* and the *LCM4ODI Getting Started Guide*.

# Starting the VCR4ODI connector

1. Start the VCR4ODI connector by executing the `vcr4odi.bat` file.  
The VCR4ODI startup screen is displayed.



2. From the drop-down list, select the profile definition you want to use for the connection.  
Before starting the VCR4ODI connector, you can verify the settings of a specific Profile by clicking the *Edit Profile* button.

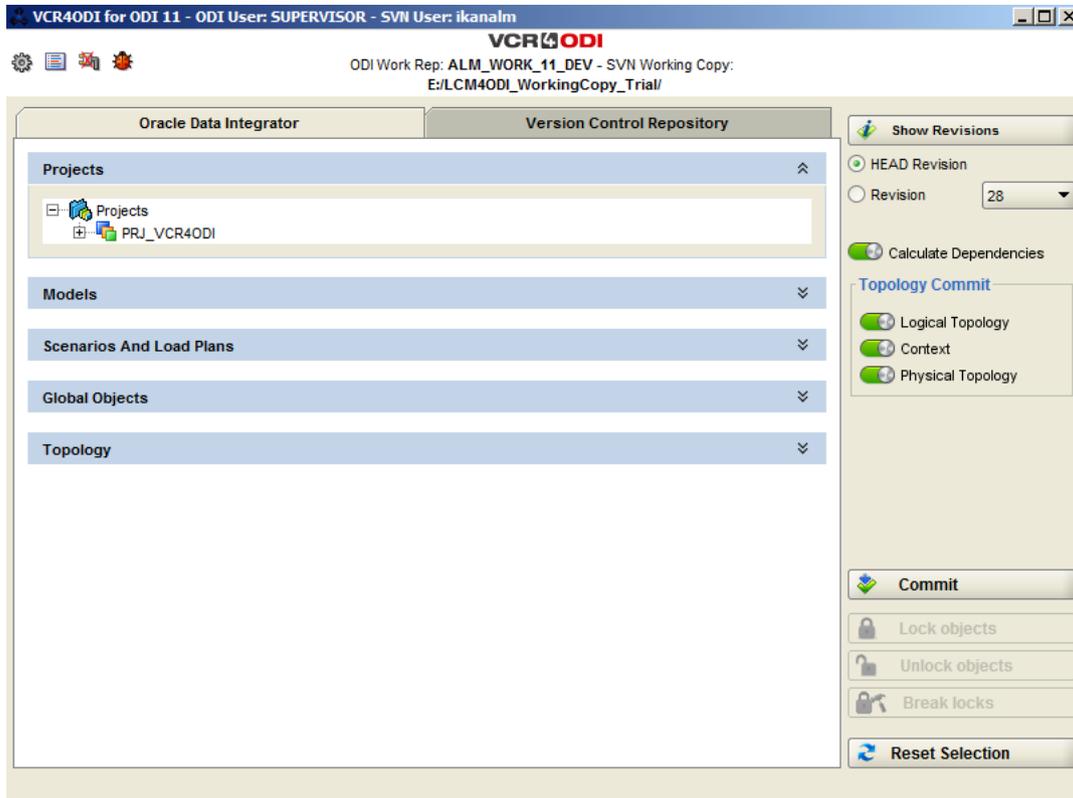
---

**Note:** If this is the first time you use the VCR4ODI Connector, you will first need to create a profile.

For more information on managing profiles, refer to the chapter [VCR4ODI Profiles](#) (page 9).

3. Click the *Start* button.

The following screen is displayed.



On this screen, two tab pages are available:

- *Oracle Data Integrator*

The *Oracle Data Integrator* tab is used to select the ODI objects you want to commit to Subversion.

For more information, refer to the section [Oracle Data Integrator Tab Page](#) (page 4)

- *Version Control Repository*

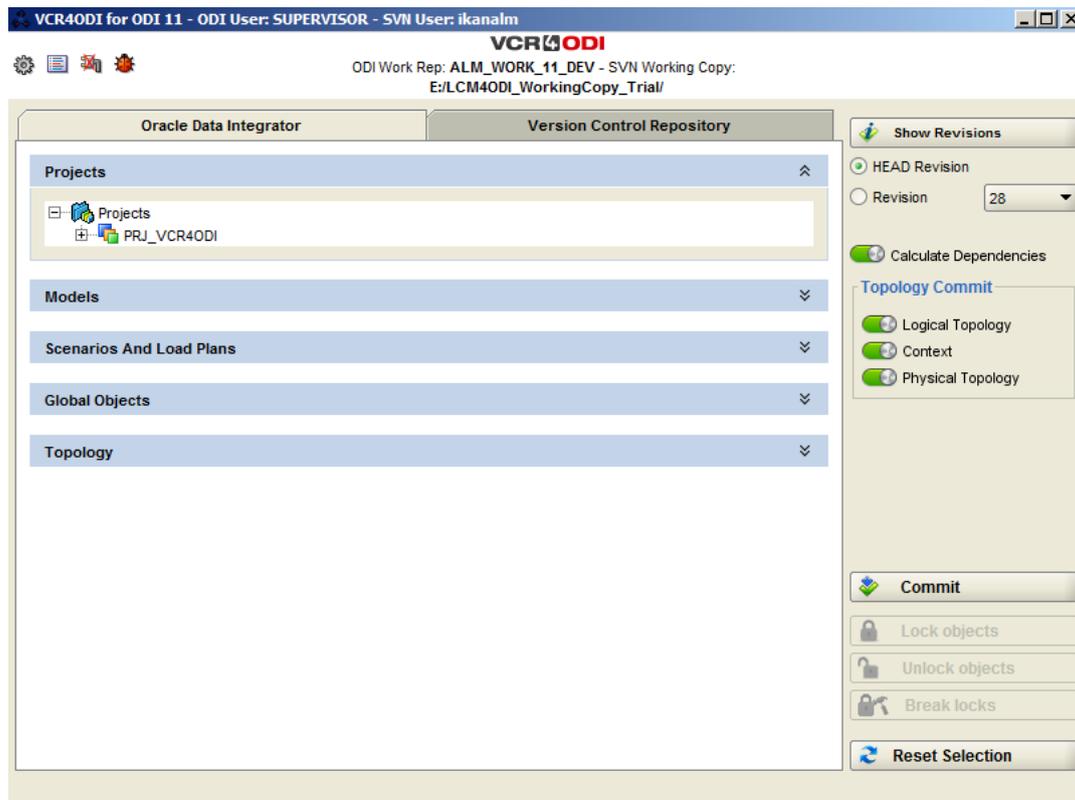
The *Version Control Repository* tab is used to select the objects you want to restore from Subversion to ODI.

For more information, refer to the section [Version Control Repository Tab Page](#) (page 5)

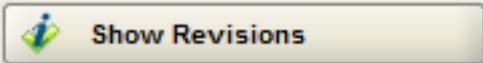
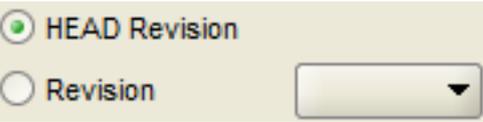
## 2.1. Oracle Data Integrator Tab Page

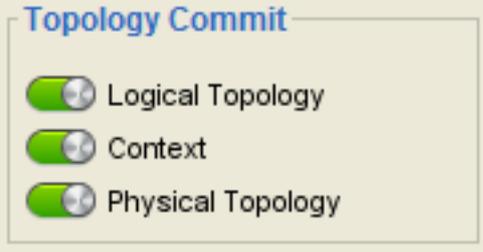
On the *Oracle Data Integrator* tab, all objects stored in the ODI repository are displayed the same way as in ODI Studio.

This tab is used to select the objects you want to commit to the VCR.



The following buttons are available:

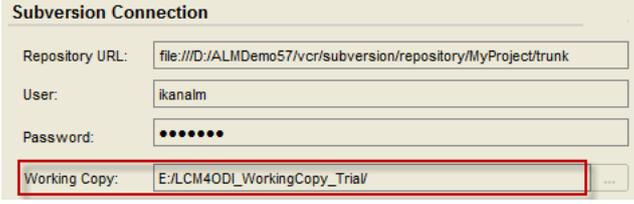
Button	Meaning
	This option is used to display the different revisions of a selected object in the Subversion repository. <b>Note:</b> To be able to use this functionality, a Tortoise client needs to be installed.
	These options are used to select the revision you want to use for the commit operation, i.e., the HEAD revision or a specific revision.
	When activating this option, VCR4ODI will also automatically calculate and commit all dependencies of the selected ODI objects. When deactivated, only the selected objects will be committed, without their dependencies. This can be helpful if you only want to commit a single set of objects.

Button	Meaning
 <p>The dialog box titled "Topology Commit" contains three toggle switches, all of which are turned on. The switches are labeled "Logical Topology", "Context", and "Physical Topology".</p>	<p>This option is used to specify which parts of the topology you want to include or exclude when committing objects: the Logical Topology, the Context and/or the Physical Topology.</p>
 <p>A rectangular button with a green checkmark icon on the left and the text "Commit" in bold black font.</p>	<p>This button is used to commit the selected ODI objects to the Version Control Repository.</p>
 <p>A rectangular button with a blue circular arrow icon on the left and the text "Reset Selection" in bold black font.</p>	<p>This button is used to deselect the objects selected in the tree structure.</p>

For more information, refer to the section [Committing Elements](#) (page 16).

## 2.2. Version Control Repository Tab Page

This tab page points to the *Working Copy Location* specified in the Profile definition you used to start up the VCR4ODI connector.

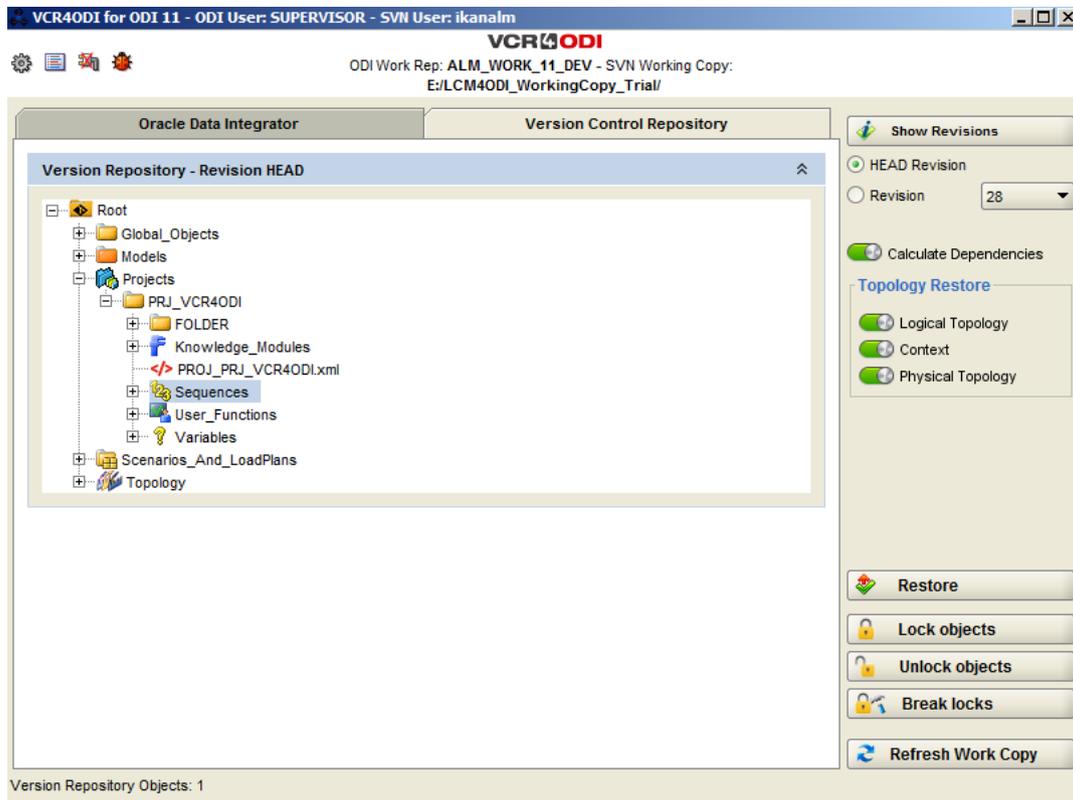


The "Subversion Connection" dialog box contains the following fields:

- Repository URL: file:///D:/ALMDemo57/vcr/subversion/repository/MyProject/trunk
- User: ikanalm
- Password: ••••••
- Working Copy: E:/LCM4ODI\_WorkingCopy\_Trial/ (highlighted with a red border)

For more information on defining profiles, refer to the chapter [VCR4ODI Profiles](#) (page 9).

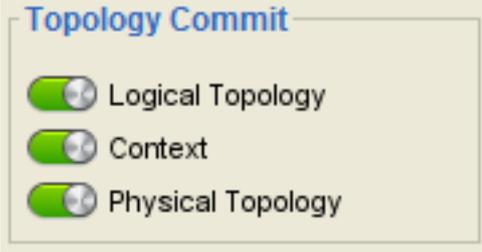
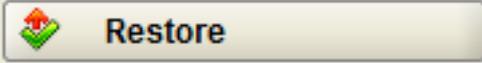
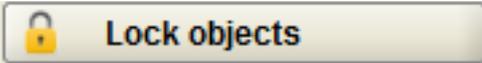
The *Version Control Repository* tab is used to view the objects which have already been committed to the VCR and which can be restored.



The following buttons will be available:

**Note:** If nothing is displayed in the tree, you have to refresh you Work Copy using the button in the bottom right corner.

Icon	Meaning
 <b>Show Revisions</b>	This option is used to display the different revisions of a selected object in the Subversion repository. <b>Note:</b> To be able to use this functionality, a Tortoise client needs to be installed.
<input checked="" type="radio"/> HEAD Revision <input type="radio"/> Revision	These options are used to select the revision you want to use for the restore operation, i.e., the HEAD revision or a specific revision.
 <b>Calculate Dependencies</b>	When activating this option, VCR4ODI will also automatically restore all objects and dependencies of the selected objects. When deactivated, only the selected objects will be restored, without their dependencies. This can be helpful if you only want to restore a single set of objects.

Icon	Meaning
	<p>This option is used to specify which parts of the topology you want to include or exclude when restoring objects: the Logical Topology, the Context and/or the Physical Topology.</p>
	<p>This button is used to restore the selected objects from the Version Control Repository to the ODI repository defined in your Profile. <b>Note:</b> You cannot restore inconsistent objects or objects that are locked by another user.</p>
	<p>This button is used to lock objects in the Version Control Repository.</p>
	<p>This button is used to unlock objects in the Version Control Repository.</p>
	<p>This button is used to break locks on objects in the Version Control Repository. <b>Important:</b> This button should be used with caution.</p>
	<p>This button is used to clean up the working copy and synchronize the tree view with the current situation in Subversion. It is recommended to refresh your local working copy before executing a restore operation.</p>
	<p>In case the revision in your local working copy is not in sync with the revision in the Subversion repository, a warning exclamation mark is displayed in the title bar of the Version Control Repository tab page.</p>

For more information, refer to the following sections:

- [Restoring Elements](#) (page 21)
- [Locking/Unlocking Elements](#) (page 26)

## 2.3. Additional Buttons

Apart from the tab-specific buttons in the outer right column, the following buttons are always available at the top of the window:

Button	Meaning
	This button is used to display the properties of the current VCR4ODI profile. Refer to the <i>VCR4ODI Installation Guide</i> for more detailed information.
	This button is used to display the log file. <b>Note:</b> the log is only available if the <i>Debug</i> option has been activated.
	This button is used to quit the VCR4ODI connector.
	These buttons are used to activate/deactivate debugging. If deactivated, only very few information and errors will be traced.

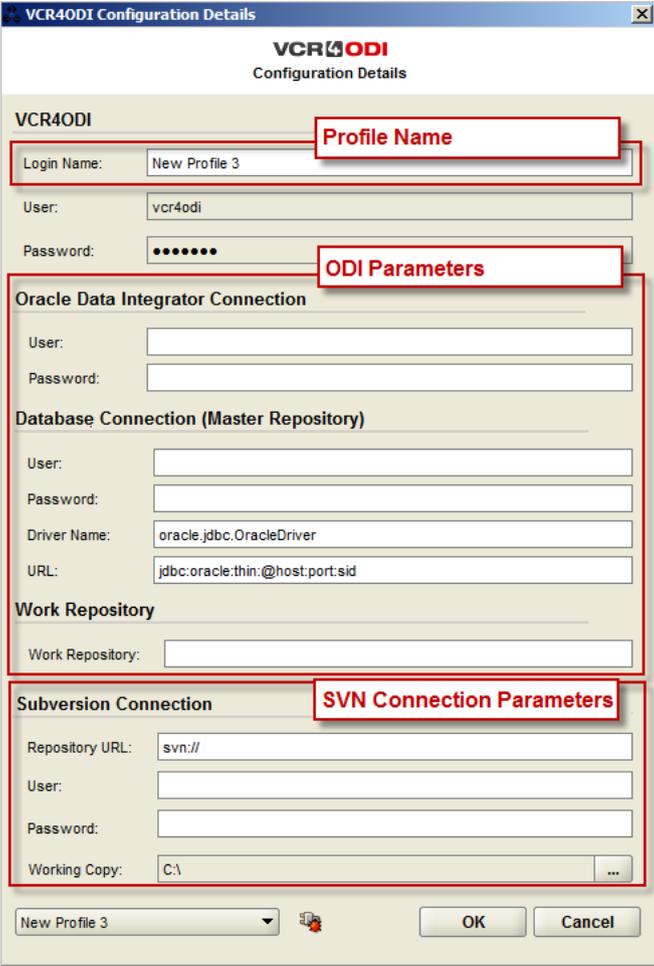
# VCR4ODI Profiles

Before you can start working with *VCR4ODI*, you must specify the configuration properties. For more detailed information, refer to the following sections:

- [Creating a New Profile](#) (page 10)
- [Copying an Existing Profile](#) (page 12)
- [Editing an Existing Profile](#) (page 12)
- [Deleting an Existing Profile](#) (page 13)

## 3.1. Creating a New Profile

1. On the VCR4ODI startup window, click the  *Add New Profile* button. The following screen is displayed:



The screen is divided into three panels:

- the VCR4ODI profile parameters  
**Note:** The fields *User* and *Password* are for future use.
- the ODI parameters: all parameter necessary to connect to the master and to the repository,
- the Subversion connection information and your working copy location

2. Provide a name for the new Profile.
3. Update the ODI parameters as required.

### Oracle Data Integrator Connection

Field	Meaning
User	The name of the ODI Supervisor User.
Password	The password of the ODI Supervisor User.

## Database Connection (Master Repository)

Field	Meaning
User	The Schema name of the Master Repository.
Password	The Schema password of the Master Repository.
Driver Name	The class name of the jdbc driver used to connect to the ODI repository.
URL	The URL path of the Master Repository.

## Work Repository

Field	Meaning
Work Repository	The name of the ODI Work Repository you want to connect to.

**Note:** All passwords will be encoded using the ODI *encode.bat* file that you can find under *%ODI\_HOME%/oracledi/agent/bin*.

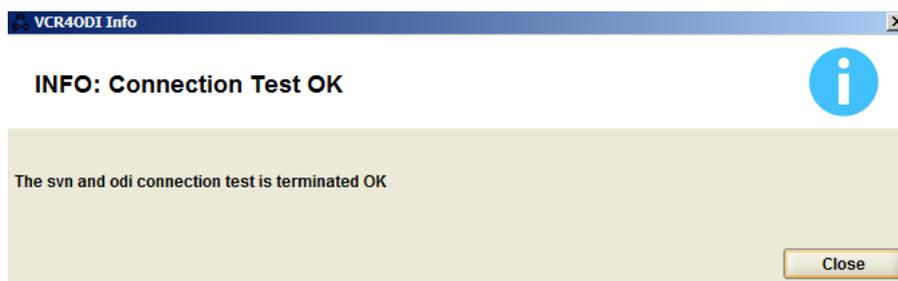
### 4. Update the Subversion Connection parameters as required.

Field	Meaning
Repository URL	The URL path of the Subversion Repository.
User ID	The name of the Subversion User.
Password	The password of the Subversion User.
Working Copy	The path to the local working copy. This folder will be the starting point for checking out objects from the repository and for managing the comparison between ODI and Subversion. <b>Important:</b> Each user should have his/her own local working repository.

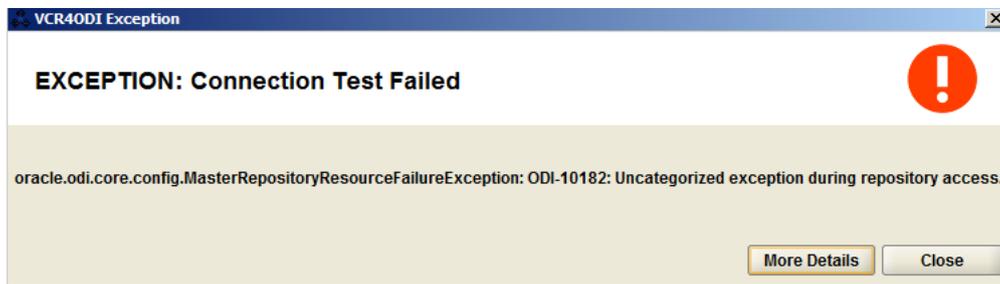
### 5. Test the connection.

You can test whether the connection is working by clicking the  button.

If the test succeeds, the following message will be displayed.



If the test failed, an error message similar to the one below will be displayed.



You can click the *More Details* button for more information concerning the connection problem.

6. If all properties are set correctly, click the *OK* button to return to the startup window. The new profile will be automatically selected.
7. Click the *Start* button to start using the VCR4ODI connector.

## 3.2. Editing an Existing Profile

1. On the VCR4ODI startup window, click the  *Edit Profile* button. The *VCR4ODI Configuration Details* screen is displayed.
2. Modify the fields as required.  
For a detailed description of the different fields, refer to the section [Creating a New Profile](#) (page 10).
3. If all properties are set correctly, click the *OK* button to return to the startup window.
4. Click the *Start* button to start the VCR4ODI connector using the modified profile.

## 3.3. Copying an Existing Profile

1. On the VCR4ODI startup window, click the  *Copy Profile* button. The *VCR4ODI Configuration Details* screen is displayed.
2. Modify the profile name and the other fields as required.  
For a detailed description of the different fields, refer to the section [Creating a New Profile](#) (page 10).

3. If all properties are set correctly, click the *OK* button to return to the startup window.
4. Click the *Start* button to start the VCR4ODI connector using the new profile.

### 3.4. Deleting an Existing Profile

1. On the VCR4ODI startup window, click the  *Delete Profile* button.  
The following window is displayed:



2. Confirm the deletion of the Profile.

---

# FUNCTIONALITIES

---

# Functionalities

When you start the VCR4ODI connector, the list of ODI objects is displayed on the *Oracle Data Integrator* tab and the list of the versioned ODI objects is displayed on the *Version Control Repository* tab.

Depending on the selected tab page, the following basic functions will be available:

- On the *Oracle Data Integrator* tab:
  - [Committing Elements](#)
- On the *Version Control Repository* tab:
  - [Restoring Elements](#)
  - [Locking/Unlocking Elements](#)

Those functions allow centralized and accurate management of the ODI elements (Tables, Package, Interface, Flat Files, ...).

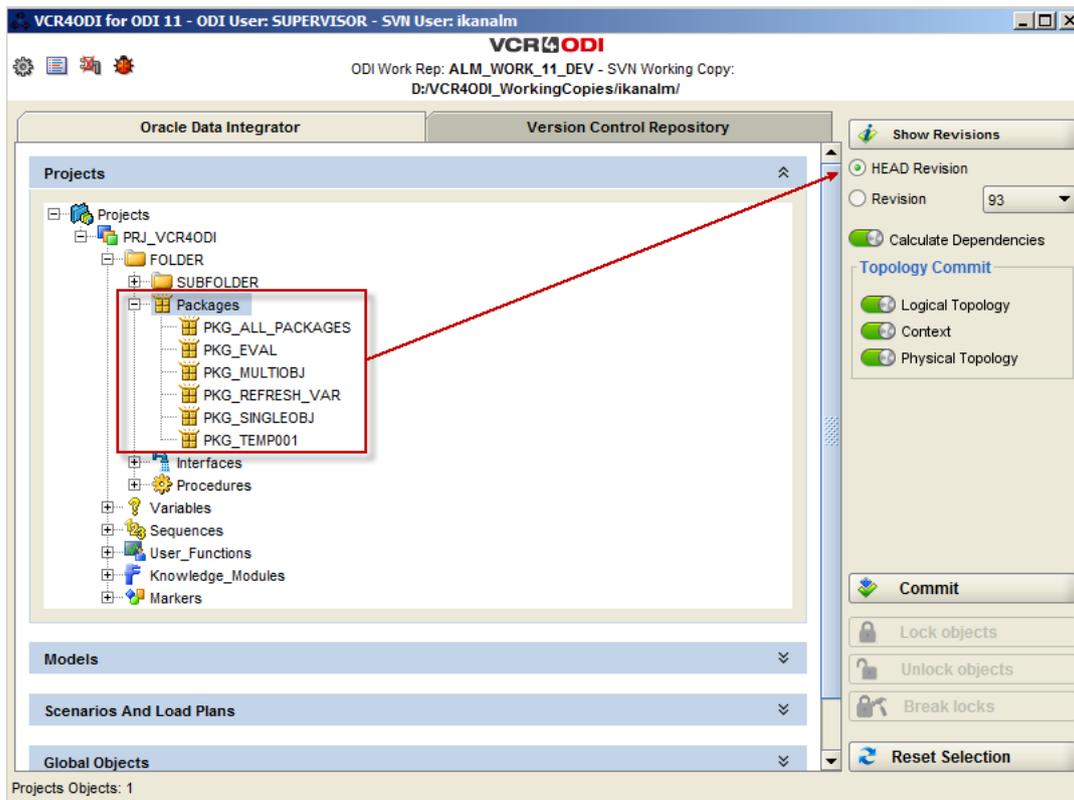
# Committing Elements

Once the development in ODI is finished, the modified objects are ready to be committed to the Version Control Repository. The commit process will first copy the modified ODI objects to the local Working Copy and, next, execute the actual commit operation to the Subversion repository.

**Note:** The first time you execute a commit operation, all ODI objects are copied to the empty local Working Copy. This might take a while depending on the number of objects.

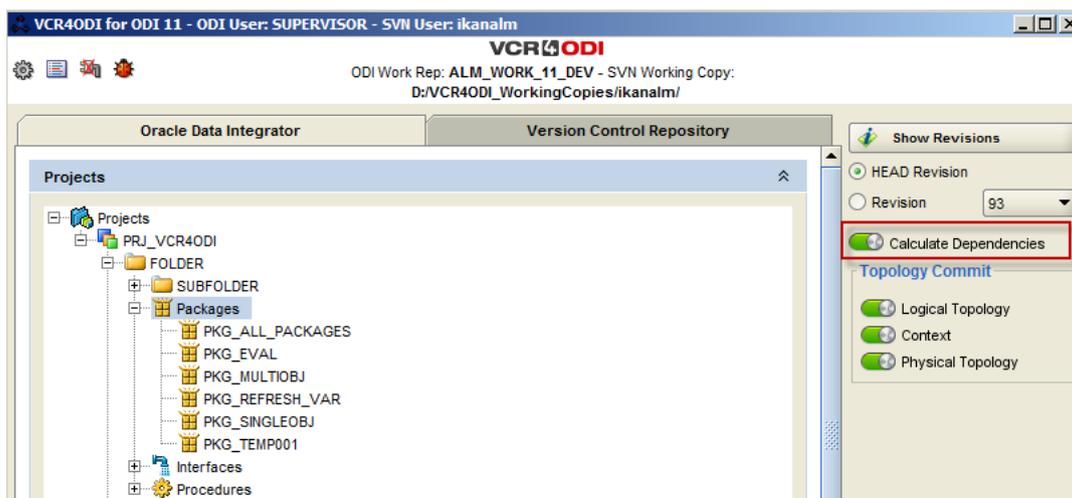
1. Start the *VCR4ODI* connector and select the *Oracle Data Integrator* tab to display the objects from the selected ODI Repository.
2. Select the object(s) you want to commit.
3. Select the object(s) you want to commit to the HEAD Revision.

**Note:** You cannot commit objects to a specific Revision.



4. Select whether you want to include the dependencies.

When activated, VCR4ODI will automatically calculate the objects' dependencies and include them when executing the commit operation.



**Note:** If you deactivate the calculation of the dependencies, topologies can also not be included.

5. Select the topology elements you want to include.



**Note:** The logical topology is needed to be able to include the context and physical topology.

6. Click the *Commit* button.

The operation of controlling the status and locking the objects starts. This may take a while depending on the number of objects.

The list of all the selected objects, and, if required, the dependencies that were calculated by the commit check-in operation, will be displayed in a new window.

For each of the objects, the change and lock status is indicated.

**VCR4ODI Commit**

**Selection buttons**

Check: **All / None Added Changed Inconsistent** Items per page: 50 ok

	Object	Type	Change Status	Lock Status	Path
<input checked="" type="checkbox"/>	PKG_MODULE	SCENARIO			Scenarios_And_LoadPlans...
<input checked="" type="checkbox"/>	MAP_VAR_REFRESH	SCENARIO	<b>Changed object</b>		Scenarios_And_LoadPlans...
<input checked="" type="checkbox"/>	PKG_ALL_PACKAGES	SCENARIO			Scenarios_And_LoadPlans...
<input checked="" type="checkbox"/>	PKG_MULTIOBJ	SCENARIO			Scenarios_And_LoadPlans...
<input checked="" type="checkbox"/>	MAP_VAR_REFRESH	INTERFACE	<b>Object changed by another user</b>	<b>Object locked by another user</b>	Projects/PRJ_VCR4ODI/FO...
<input checked="" type="checkbox"/>	PKG_ALL_PACKAGES	PACKAGE			Projects/PRJ_VCR4ODI/FO...
<input checked="" type="checkbox"/>	PKG_MULTIOBJ	PACKAGE			Projects/PRJ_VCR4ODI/FO...
<input checked="" type="checkbox"/>	PKG_EVAL	PACKAGE	<b>Inconsistent object</b>		Projects/PRJ_VCR4ODI/FO...
<input checked="" type="checkbox"/>	PKG_TEMP001	PACKAGE	<b>Added object</b>		Projects/PRJ_VCR4ODI/FO...
<input type="checkbox"/>	REFRESH_VAR	VARIABLE			Projects/PRJ_VCR4ODI/Var...
<input type="checkbox"/>	MAP_SET_OP	SCENARIO	<b>Unchanged object</b>		Scenarios_And_LoadPlans...

Page 1 of 2  
51 Total Objects - 9 Selected Objects

**Commit**

**Note:** By default, the objects are sorted by "Change Status". If required, you can order them differently by clicking the column headings.

The possible statuses are:

Status	Description
	The object has not been changed.
	The object has been changed by the current user.
	The object has been changed by another user. <b>Note:</b> If objects have been modified by another user, you will not be able to commit.
	The object is new.
	The object is inconsistent.
	The object is not locked.
	The object is locked by the current user.
	The object is locked by another user. <b>Note:</b> If the object has been modified by that other user, you will not be able to commit.

**Note:** You will not be able to commit inconsistent objects or objects that are locked by other users.

- If required, you can still refine the selection of objects that will be committed using the selection options above the displayed list.

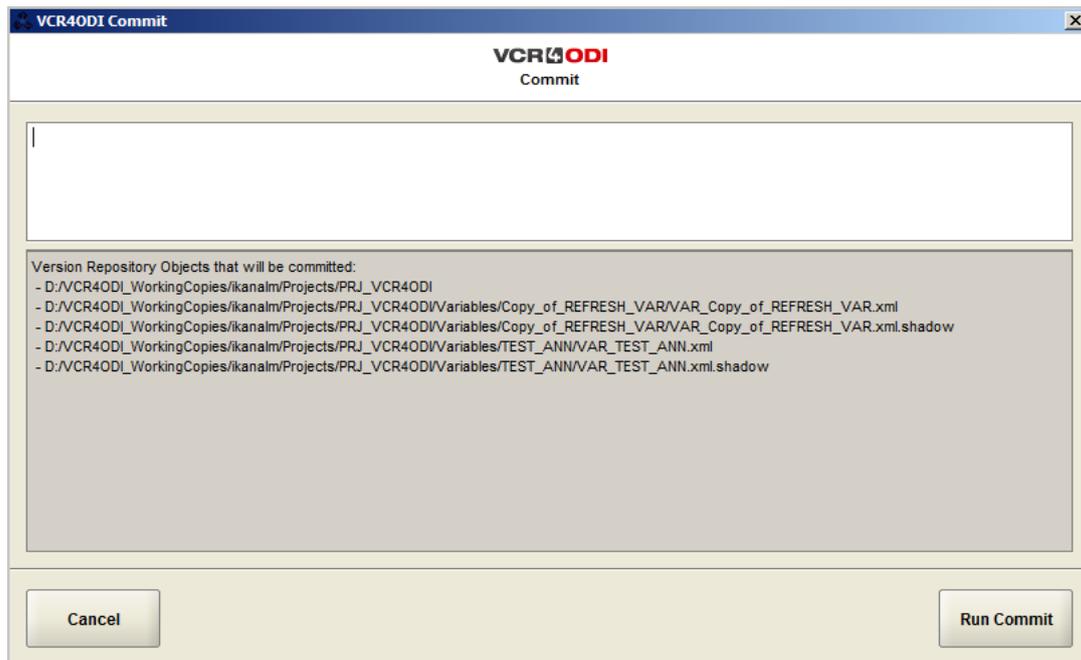
Several additional selection options are available:

All/None	Clicking the <i>All/None</i> option will select/deselect all objects.
Added	Clicking the <i>Added</i> option will select/deselect the new objects.
Changed	Clicking the <i>Changed</i> option will select/deselect the modified objects.
Inconsistent	Clicking the <i>Inconsistent</i> option will select/deselect the inconsistent objects.

**Note:** You can use the navigation arrows to go to the next/previous page.

8. Confirm the selection by clicking the *Commit* button.
9. Next, you have to enter a comment.  
This comment will be stored in Subversion.

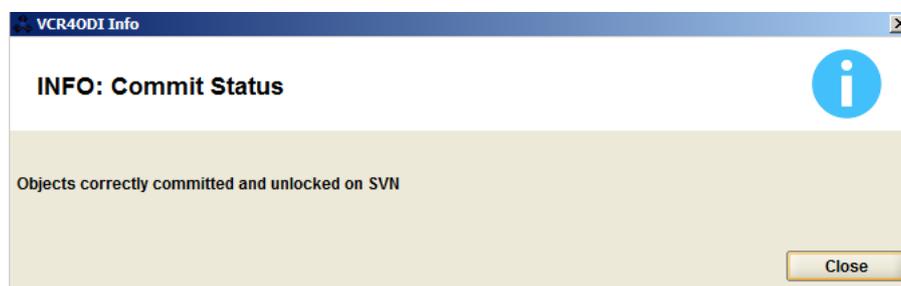
**Note:** If you use IKAN ALM to build your application, combined with a Jira or HP Quality Center Issue Tracking System, you could enter the issue number(s) in this field. IKAN ALM will automatically update Jira/HP Quality Center and create a direct link to the issue in the IKAN ALM interface.



10. Click the *Run Commit* button to finalize the commit procedure.  
The selected objects are committed to the Subversion repository.  
You can also click *Cancel* to stop the commit operation and return to the *Oracle Data Integrator* tab.

**Note:** If one or more constraints are violated, you will not be able to execute the commit operation. Refer to the displayed error message, for more information about the constraint violation problem.

11. If the commit procedure executes correctly, the following message will be displayed:



Click the *Close* button to return to the *Oracle Data Integrator* tab.

---

# Restoring Elements

Before starting any new development on objects which have already been versioned, a developer will have to restore all objects concerned from the Version Control Repository.

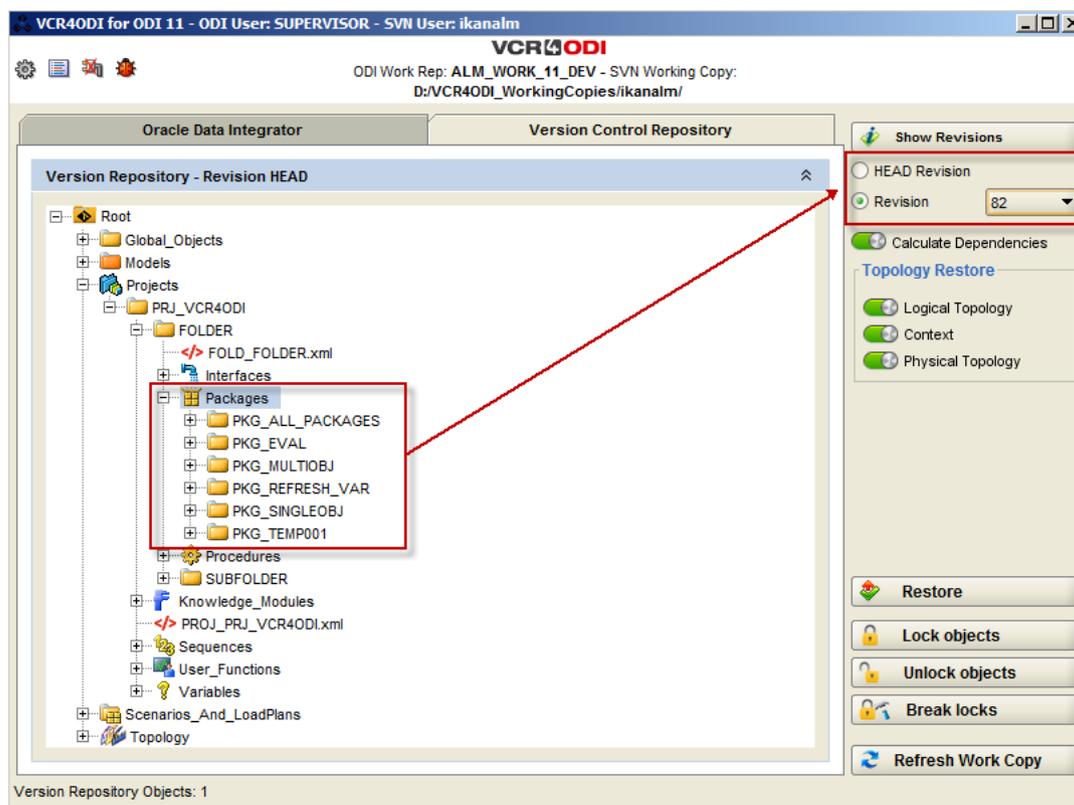
Restoring the latest version from the VCR is necessary to start your development on a consistent set of objects as some objects in the ODI Repository may have been changed or may have become inconsistent.

During the restore operation, the objects are checked out from the VCR server to the User's local working directory. Next, when confirming the restore, the objects will be restored (imported) into the ODI central server.

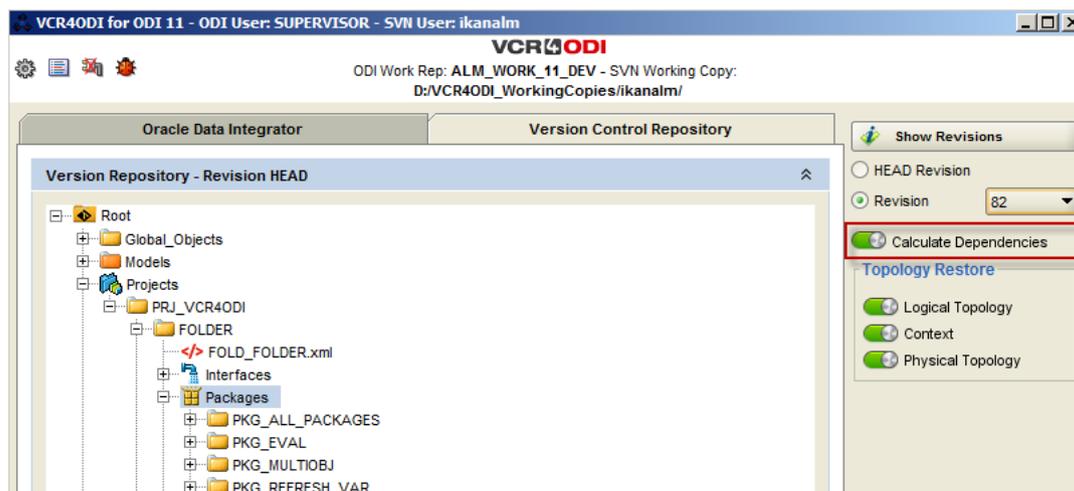
After a successful restore, the imported objects in the ODI Repository will be updated to the latest consistent version and locked by the user who performed the operation, in order to manage the concurrency in a centralized environment.

1. Start the *VCR4ODI* connector and select the *Version Control Repository* tab to display the objects in the Subversion Repository.
2. Select the object(s) you want restore.
3. Select whether you want to restore the object(s) from the HEAD Revision or from a specific Revision.

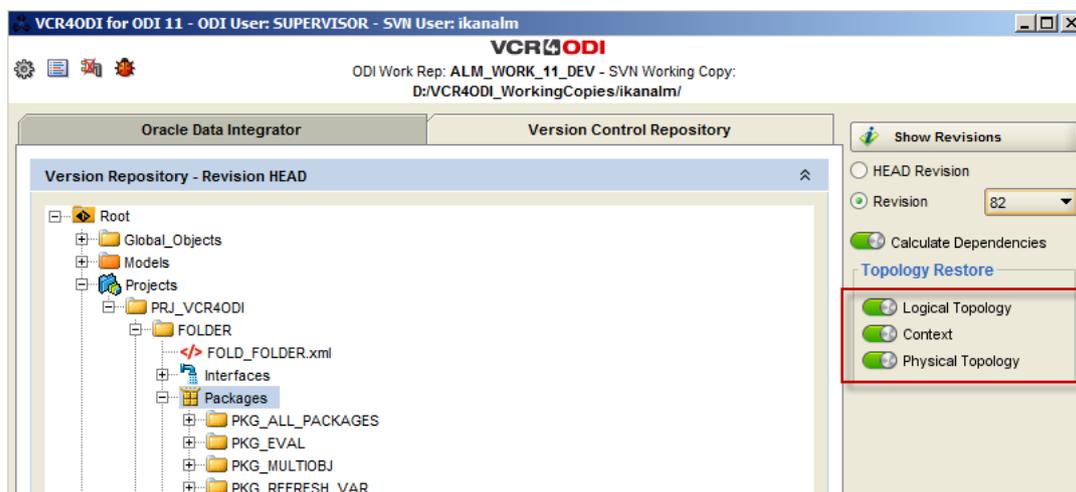
In case you want to restore from a specific Revision, you must select the required Revision from the drop-down list next to the Revision option.



4. Select whether you want to include the dependencies.  
When activated, VCR4ODI will automatically restore the objects' dependencies and include them when executing the restore operation.



- Select the topology elements you want to include.

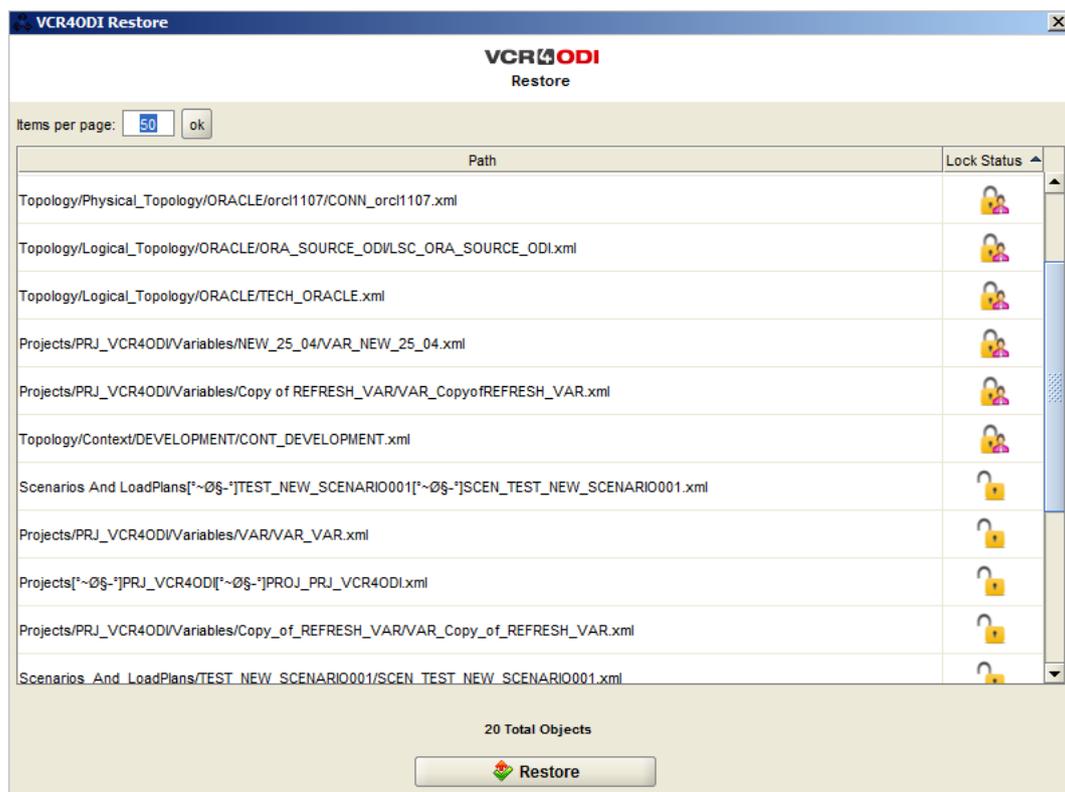


- Click the *Restore* button

The operation of controlling the status and locking the objects starts. This may take a while depending on the number of selected objects (and dependencies).

The list of all the selected objects, and, if selected, the dependencies that were automatically calculated, will be displayed in a new window.

For each of the objects, the lock status is indicated.



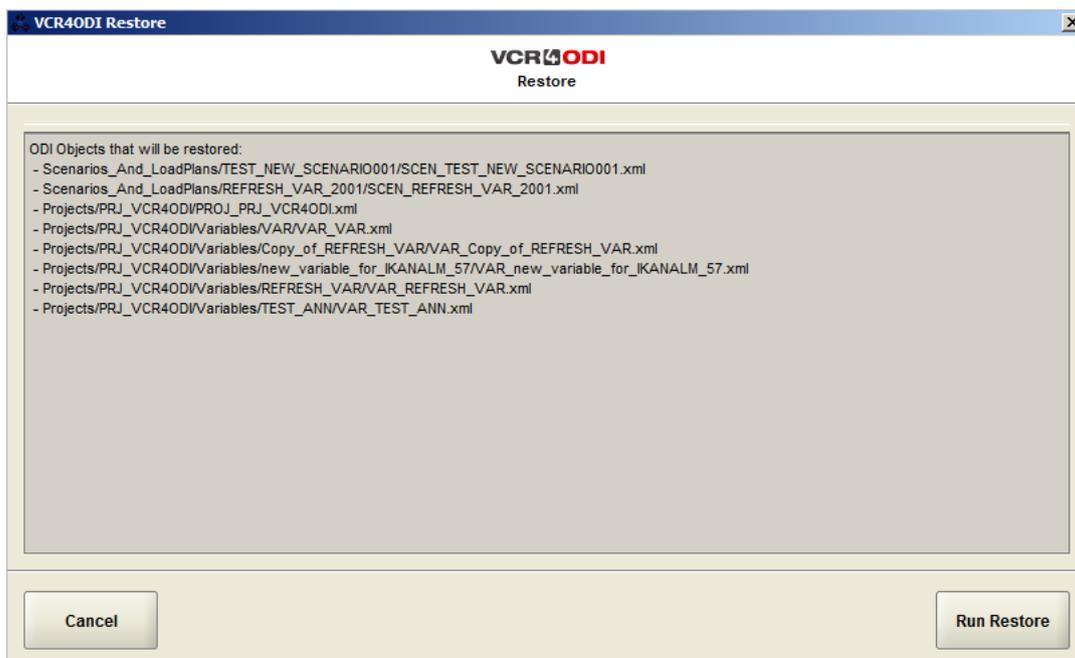
The possible statuses are:

Status	Description
	The object is not locked.
	The object is locked by the current user.
	The object is locked by another user.

**Note:** You will not be able to restore inconsistent objects or objects that are locked by other users.

- Click the *Restore* button.

The list of objects that will be restored is displayed.



- Click the *Run Restore* button to finalize the restore procedure.

You can also click *Cancel* to stop the restore procedure and return to the *Version Control Repository* tab.

9. If the restore procedure executes correctly, the following message will be displayed:



Click the *Close* button to return to the *Version Control Repository* tab.

10. **IMPORTANT:** after a restore operation, the objects are locked by the current user. In order to make them available for other users, you need to unlock the restored objects. For more information, refer to the section [Locking/Unlocking Elements](#) (page 26).

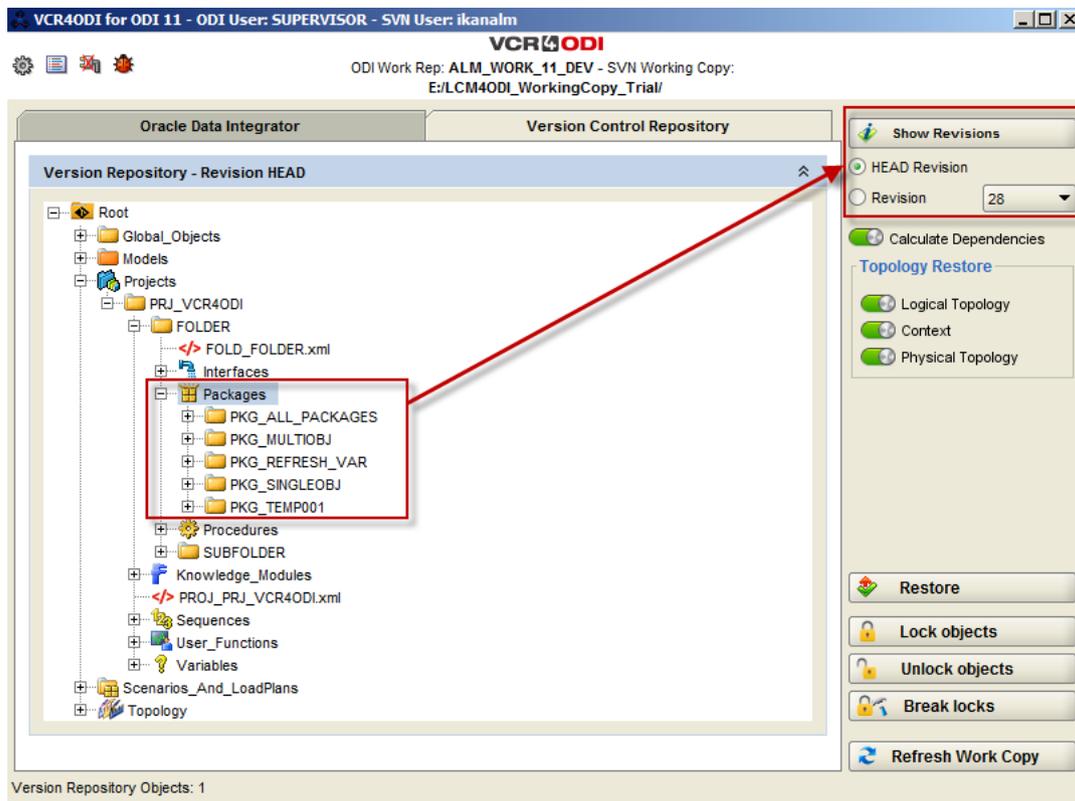
# Locking/Unlocking Elements

On the *Version Control Repository* tab, where all versioned objects in the VCR project are displayed, you can manage the locking status of each of the objects.

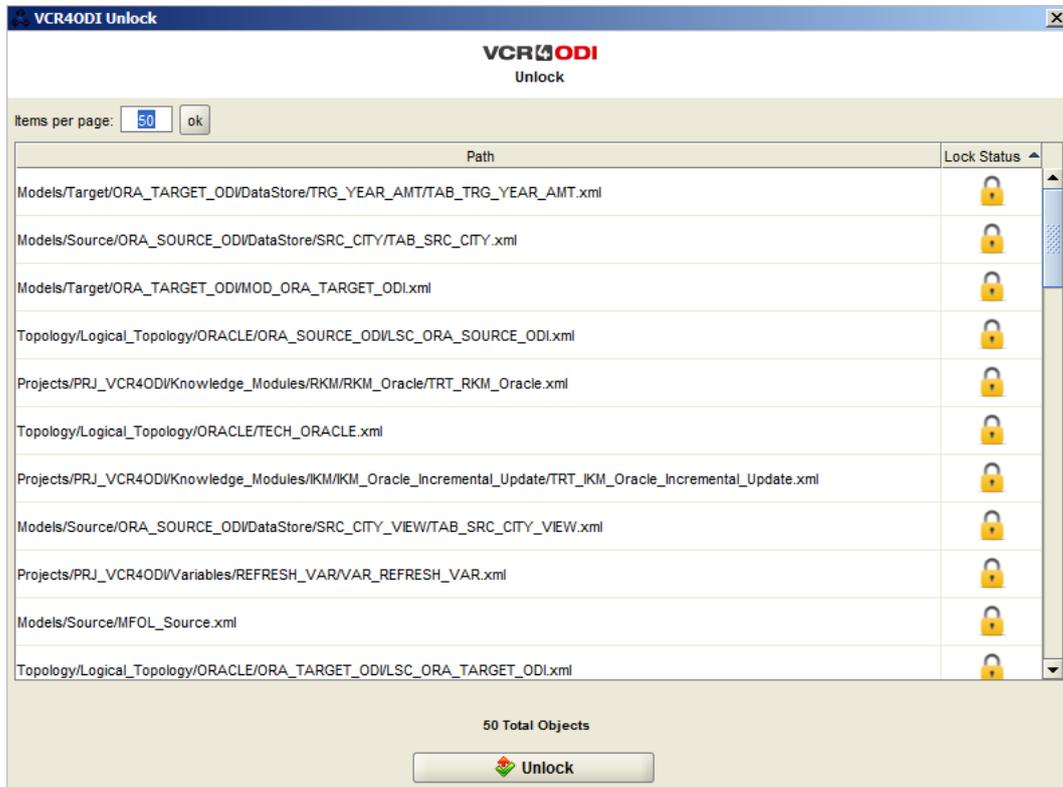
The following buttons are available to modify the lock status of the objects:

Button	Explanation
 <b>Lock objects</b>	Other Users will not be able to work on the objects.
 <b>Unlock objects</b>	This function releases objects locked by the current user.
 <b>Break locks</b>	Breaking a lock forces the unlocking of an object locked by another user. This function should be used with care.

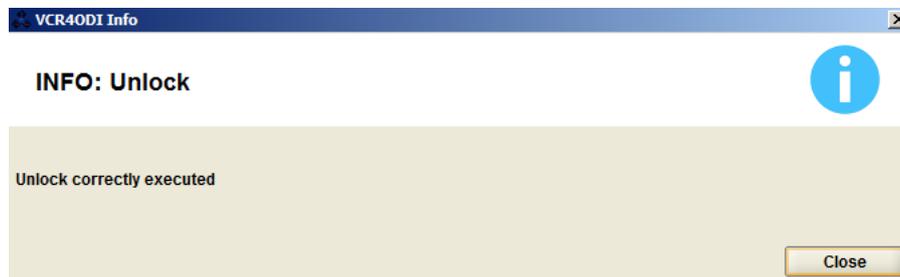
1. Start the *VCR4ODI* connector and select the *Version Control Repository* tab to display the objects in the ODI Repository.
2. Select the elements you want to change the lock status for.
3. Select whether you want to change the lock status of the object(s) for the HEAD Revision or for a specific Revision.



4. Depending on what you want to change, click the *Lock*, *Unlock* or the *Break locks* button.



5. If the procedure executes correctly, a message similar to this one will be displayed:



Click the *Close* button to return to the *Version Control Repository* tab.

---

# PROBLEM SOLVING

---

# Debugging and Logging

Click the  *View Log* button on the main window to display the connector's log file.



```
VCR4ODI Log

VCR4ODI
Log

914: DEBUG 2016-05-03T10:06:21,549 (OdiConnectionManager.java:288) [AWT-EventQueue-0][[]] com.odialm.connect
ion.OdiConnectionManager.class - Committing Transaction oracle.odi.core.persistence.transaction.support.Trans
actionManagerSpringAdapter$TransactionStatusSpringAdapter@ed66c05
913: DEBUG 2016-05-03T10:06:20,800 (ImportManager.java:113) [AWT-EventQueue-0][[]] com.odialm.imp.ImportMana
ger.class - Importing odi object Projects/PRJ_VCR4ODI/FOLDER/Packages/PKG_EVAL/PACK_PKG_EVAL.xml
912: DEBUG 2016-05-03T10:06:20,799 (OdiConnectionManager.java:204) [AWT-EventQueue-0][[]] com.odialm.connect
ion.OdiConnectionManager.class - Setting Thread Authorization...
911: DEBUG 2016-05-03T10:06:20,799 (OdiConnectionManager.java:149) [AWT-EventQueue-0][[]] com.odialm.connect
ion.OdiConnectionManager.class - NEW TRANSACTION STATUS oracle.odi.core.persistence.transaction.support.Trans
actionManagerSpringAdapter$TransactionStatusSpringAdapter@ed66c05
910: DEBUG 2016-05-03T10:06:20,796 (OdiConnectionManager.java:147) [AWT-EventQueue-0][[]] com.odialm.connect
ion.OdiConnectionManager.class - Getting Transaction...
909: DEBUG 2016-05-03T10:06:20,796 (OdiConnectionManager.java:145) [AWT-EventQueue-0][[]] com.odialm.connect
ion.OdiConnectionManager.class - Creating Transaction Manager...
908: DEBUG 2016-05-03T10:06:20,795 (OdiConnectionManager.java:142) [AWT-EventQueue-0][[]] com.odialm.connect
ion.OdiConnectionManager.class - Creating DefaultTransactionDefiniton as PROPAGATION_REQUIRES_NEW...
907: DEBUG 2016-05-03T10:06:20,793 (OdiConnectionManager.java:233) [AWT-EventQueue-0][[]] com.odialm.connect
ion.OdiConnectionManager.class - Creating Transaction And Authentication...
906: DEBUG 2016-05-03T10:06:20,778 (OdiConnectionManager.java:288) [AWT-EventQueue-0][[]] com.odialm.connect
ion.OdiConnectionManager.class - Committing Transaction oracle.odi.core.persistence.transaction.support.Trans
actionManagerSpringAdapter$TransactionStatusSpringAdapter@3b279c0f
905: DEBUG 2016-05-03T10:06:19,849 (ImportManager.java:113) [AWT-EventQueue-0][[]] com.odialm.imp.ImportMana
ger.class - Importing odi object Projects/PRJ_VCR4ODI/Knowledge_Modules/SKM/SKM_Oracle/TRT_SKM_Oracle.xml
```

---

# Managing Conflicts

As mentioned earlier, the *VCR4ODI* connector manages the concurrency between different users and ensures the consistency of the ODI Repository environment for each revision of a VCR project.

To ensure concurrency and consistency, working with two centralized systems, which communicate with each other, whereby multiple users can develop simultaneously, VCR4ODI uses the native VCR locking functionality and every commit or restore operation applies some internally defined constraints. To be able to execute any operation successfully, those constraints must not be violated.

Below you will find the explanation of all mandatory constraints for each operation between ODI and the VCR. Based on those constraints, VCR4ODI is able to maintain consistency in the ODI Repository and manage the concurrency during the parallel development of multiple users.

## 9.1. Commit Constraints

All ODI objects that have to be committed:

- may not be locked by another user
- must be consistent in the ODI environment

## 9.2. Restore Constraints

All ODI objects that have to be restored:

- may not be locked by another user
- must be in XML format

## B

Breaking Locks 26

## C

Changing Releases 15  
Committing Elements 15, 16  
Copying Elements 15

## D

Debugging 30

## E

Elements  
    Committing 15  
    Copying 15  
    Locking 15  
    Removing 15  
    Unlocking 15  
    Updating 15

## F

Functions 15  
    Breaking Locks 26  
    Changing Releases 15  
    Committing Elements 15, 16  
    Copying Elements 15  
    Locking Elements 15, 26  
    Removing Elements 15  
    Restoring Elements 21  
    Unlocking Elements 15, 26  
    Updating Elements 15

## L

Locking Elements 15, 26  
Logging 30

## M

Managing Conflicts 31

## R

Removing Elements 15  
Restoring Elements 21

## U

Unlocking Elements 15, 26  
Updating Elements 15  
Using the VCR4ODI Connector 2

## V

VCR4ODI Connector  
    Using 2

IKAN ALM is developed by IKAN Development N.V., Belgium © Copyright 2019  
The IKAN Development and IKAN ALM logos and names and all other IKAN product  
or service names are trademarks of IKAN Development N.V.

Contact Information:  
IKAN Development N.V.  
Kardinaal Mercierplein 2  
2800 Mechelen  
BELGIUM  
<http://www.ikanalm.com>  
[info@ikan.be](mailto:info@ikan.be)

VCR4ODI is developed by D&T srl, Italy © Copyright 2019  
The D&T logos and names and all other D&T product or service names are trademarks  
of D&T srl.

Contact Information:  
Database & Technology srl  
Largo Promessi Sposi, 4  
20142 Milano  
Italy  
<http://www.databtech.com>  
[dt.marketing@databtech.net](mailto:dt.marketing@databtech.net)

All other trademarks are property of their respective owners.