

Life Cycle Management for ODI Getting Started Guide

Release 4.3

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 - GENERAL INFORMATION

Chapter 1 - Introduction	1
1.1. System Setup for this Getting Started Guide.....	1
1.2. Files Needed.....	2
1.3. Targeted Audience.....	2
1.4. What Will You Learn?	2
Chapter 2 - About this Guide	3
2.1. Installing the Different Components	3
2.2. Demonstration of the Life Cycle Management Functionalities for ODI.....	4
2.3. Quick Start.....	4
Chapter 3 - Required Information	5
3.1. Required Information for Configuring IKAN ALM.....	5
3.2. Required Information for Configuring VCR4ODI	6
3.3. Required Information for Configuring the ODI Solution Phases	8
3.4. Other Oracle Initialization Values	9
Chapter 4 - Getting Started	10
4.1. Downloading the zip files from our website.....	10
4.2. Downloading the “LCM4ODI_trial” folder from the FTP Server	11
4.3. Let’s get started	11

PART II - INSTALLING IKANALM

Chapter 5 - Installing the IKAN ALM Demo Version	13
5.1. Installing the Software	13
5.2. Install the IKAN ALM License	19
5.3. Start	20
5.4. Open the IKAN ALM Web Application	21
5.5. Log on to the IKAN ALM Web Application.....	22

PART III - SETTING UP SUBVERSION

Chapter 6 - Setting Up Subversion	24
6.1. Setting up the Subversion Versioning System.....	24

PART IV - INSTALLING THE VCR4ODI CONNECTOR

Chapter 7 - Installing VCR4ODI	27
Chapter 8 - Creating a VCR4ODI Profile.....	30
Chapter 9 - Starting VCR4ODI.....	34

PART V - USING THE LCM4ODI SOLUTION

Chapter 10 - Creating a Project.....	36
10.1. Subversion Definition	36
10.2. Create your First Project.....	38
Chapter 11 - The BASE Life Cycle.....	39
Chapter 12 - Setting up the Build and Deploy Levels.....	41
Chapter 13 - Creating the Build and Deploy Environments.....	43
Chapter 14 - Using Phases to define the Build and Deploy processes	45
14.1. Core IKAN ALM Phases.....	45
14.2. Specific Solution Phases	46
Chapter 15 - ODI Solution Phases	47
15.1. Copying the ODI Solution Phases	47
15.2. Implementing the Phases in IKAN ALM	48

Chapter 16 - Auditing the Project	60
Chapter 17 - Committing a Change Using VCR4ODI.....	61
Chapter 18 - Creating the Build Level Request	65
Chapter 19 - Creating the Deliver Level Request(s).....	71

GENERAL INFORMATION

CHAPTER 1

Introduction

IKAN ALM provides a Life Cycle Management Solution for ODI using specific Solution Phases. Our LCM4ODI solution needs the following working installations:

- One of the following versions of ODI: ODI 11gR1, ODI 12cR1 or ODI 12cR2
- The corresponding version of ODI Studio
- The IKAN ALM Demo Version

Note: the IKAN ALM Demo version used for this *LCM for ODI Trial* comes with a pre-installed file-based Subversion repository.

- The VCR4ODI Connector (for ODI 11gR1, ODI 12cR1 or ODI 12cR2)

All these components can run on one and the same machine, or on different machines.

For example:

You could have your ODI Repositories on a Linux machine and IKAN ALM running on the same or on another Linux machine, or on a Windows system. The VCR4ODI connector together with ODI Studio could be installed on yet another machine (usually a Windows workstation) and Subversion could run on the machine where ODI or IKANALM or VCR4ODI runs or on another Linux or Windows machine.?

The only requirement is that ODI Studio is installed on the same machine as IKAN ALM and on the same machine as VCR4ODI. So if IKAN ALM and VCR4ODI are not installed on the same machine, you will need two installations of ODI Studio.

1.1. System Setup for this Getting Started Guide

For this *Getting Started Guide*, we have chosen a simple setup whereby:

- Windows is used as main environment.
- On one and the same Windows machine, the following components must be installed: IKAN ALM Demo (containing Subversion), VCR4ODI and ODI Studio.
- ODI can be installed on that same Windows machine or on any other machine.

Note: If the Oracle DB that is being used for the ODI repositories requires connections via TNS, Oracle client software must be installed on the developer machine running VCR4ODI. See also [Installing VCR4ODI](#) (page 27).

1.2. Files Needed

All necessary files are available on our website <https://www.redbridgesoftware.com/download.html>.

You can also download *LCM4ODI_trial* folder from our FTP Server. In that case, you will find the necessary FTP details (user name and password) in the mail you received from our Support Team.

1.3. Targeted Audience

The target audience of this guide are engineers who want to know how to fully automate the Life Cycle Management process using ODI and ODI Studio in combination with the VCR4ODI connector, IKAN ALM and its dedicated ODI Solution Phases.

1.4. What Will You Learn?

After you followed this tutorial, you will be able to:

- set up a Project,
- implement you own life cycle in IKAN ALM and create the necessary Levels and Environments,
- install the ODI Solution Phases and adapt them to your requirements, and
- understand how the full process of versioning, managing life cycles, creating archives and restoring objects using the ODI Solution phases works.

About this Guide

This *Getting Started Guide* describes the complete procedure to install all the different components and set up and adapt the ODI Solution Phases using a simple example.

For the examples in this guide, we use Windows in combination with ODI 11gR1, IKAN ALM 5.7 and the corresponding ODI Solution Phases. If you are using ODI cR1 or cR2, you can use this guide as well. The only difference resides in the parameters for the ODI Solution Phases.

IMPORTANT!

For the purpose of this *Getting Started Guide*, all components, except for ODI, must be installed on the same machine! See also [System Setup for this Getting Started Guide](#) (page 1).

2.1. Installing the Different Components

The first parts in this *Getting Started Guide* will guide you through the different steps for installing the following components.

1. *[A working installation of ODI must be available (may be installed on a different machine).]*
2. Downloading the required files ([Getting Started](#) (page 10))
3. Installing the IKAN ALM Demo version ([Installing the IKAN ALM Demo Version](#) (page 13))

The IKAN ALM Demo installation will fulfill the following requirements: the IKAN ALM Server and Agent will be installed and the Ant scripting tool and Subversion repository will be installed and configured. It also includes the Subversion server.

4. Setting up Subversion ([Setting Up Subversion](#) (page 24))
5. Installing the VCR4ODI connector ([Installing VCR4ODI](#) (page 27))

The VCR4ODI connector enables the interaction between the ODI Repositories and the Version Control Repository (Subversion). It is used to commit ODI objects to the VCR and, vice versa, to restore objects from the VCR into ODI.

Note that the full versions of the different installation guides can be found on our website.

2.2. Demonstration of the Life Cycle Management Functionalities for ODI

Next, all components being installed and configured, the part *Using the LCM4ODI Solution* explains how to achieve full Life Cycle Management for ODI using a simple example ([ODI Solution Phases](#) (page 47)).

The different steps include:

- creating an IKAN ALM project,
- creating the required Life Cycle, Levels and Environments,
- importing the ODI Solution Phases and adapt them to your requirements,
- auditing the project,
- committing a change using VCR4ODI,
- creating a Build Level Request using the Sources in the Subversion Repository,
- creating a Deploy Level Request to deliver the Build result to the Test Environment.

If you want to follow this step-by-step *Getting Started Guide*, continue to the first step: downloading the required files ([Downloading the “LCM4ODI trial” folder from the FTP Server](#) (page 11)).

2.3. Quick Start

You might also download, install and configure the different components separately without following this *Getting Started Guide* step-by-step.

The installation of the components is straight-forward and consists of the following actions:

1. Install IKAN ALM Demo

If you downloaded the files from our FTP Server: extract the *ikanalm_trial_x64.exe* file and execute the *almdemo_install_5-7_x64.exe* file.

If you downloaded the files from our website, you can immediately execute the installer.

2. Set up Subversion for ODI

See also [Setting Up Subversion](#) (page 24).

3. Install and configure the VCR4ODI connector

Install the VCR4ODI connector for the ODI Version you are using.

See also [Installing VCR4ODI](#) (page 27).

4. Install and configure the ODI Solution Phases

See also [ODI Solution Phases](#) (page 47).

Required Information

To follow the procedure described in this *Getting Started Guide*, you need the following information to configure the connections between ODI, IKAN ALM and VCR4ODI.

3.1. Required Information for Configuring IKAN ALM

The following information is needed when creating the Subversion definition in IKAN ALM.

Required Information	Your Value
The name of the VCR Project you will be using. See also Setting Up Subversion (page 24).	

3.2. Required Information for Configuring VCR4ODI

The following information is needed when creating the VCR4ODI User Profile.

Required Information	Your Value
The name of the ODI User.	
The password of the ODI User.	
The Schema name of the Master Repository.	
The Schema password of the Master Repository.	
The class name of the jdbc driver used to connect to the ODI repository.	
The URL path of the Master Repository.	
The name of the ODI Work Repository you want to connect to.	

Required Information

Your Value

The URL path of the Subversion Repository.

This path is composed as follows:

```
file:///<ALM_HOME>/vcr/subversion/repository/  
<MyProject>/trunk
```

In which <ALM_HOME> stands for the IKAN ALM demo installation location) and <MyProject> for the VCR Project name you created earlier ([Setting Up Subversion](#) (page 24))?

The path to the repository can also be found in the Version Control Repository definition in the Global Administration section in IKAN ALM ([Subversion Definition](#) (page 36)).

The path to a 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.

Important: each user should have his/her own local working repository.

3.3. Required Information for Configuring the ODI Solution Phases

The following information is needed when defining the parameters used in the ODI Solution Phases.

Phase Parameters						
Name	Default value	Description	Mandatory	Secure	Integration Type	
alm.phase.builder			✓		ANT	
alm.phase.extractBundle	true		✓		None	
alm.phase.mainScript	restoreODI11.xml	Main Script	✓		None	
odi.action	restoreAllObjects	ODI Restore Action. Valid values are : restoreAllObjects, restoreFromProjectList, restoreAllScenarios	✓		None	
odi.action.loglevel	info	Log Level of the ODI Restore Action. Valid values are : error, warning, info, debug	✓		None	
odi.export.dir	\${source}/odi	Folder that contains the exported ODI files that will be restored	✓		None	
odi.home	C:\oracle\product\ODI_11117	ODI_HOME Location	✓		None	
odi.home.classpath.includes	oracledi.sdk/lib/*.jar	List of paths to jars to include on the classpath of the phase. Paths should be separated by a ", " and are relative to the ODI_HOME	✓		None	
odi.home.jps-config	oracledi/client/odi/bin/jps-config.xml	Path to ODI jps-config.xml file, relative to ODI_HOME	✓		None	
odi.user.name	SUPERVISOR	ODI User Name	✓		None	
odi.user.passwd	*****	ODI User password	✓	✓	None	
odi.workRepository.name		ODI Work Repository Name	✓		None	
oracle.home		ORACLE_HOME Location. Required when rdbms.master.tns.connect.descriptor has been set			None	
project.codes		ODI Project Name(s) to Restore. Names must be separated by a comma (,).			None	
rdbms.master.host	localhost	Hostname of the ODI Master Repository Database Connection	✓		None	
rdbms.master.passwd	*****	Password of the ODI Master Repository Database Connection	✓	✓	None	
rdbms.master.port	1521	TCP port of the ODI Master Repository Database Connection			None	
rdbms.master.sid	orcl1107	SID of the ODI Master Repository Database Connection	✓		None	
rdbms.master.tns.connect.descriptor	(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=\${rdbms.master.host})(PORT=\${rdbms.master.port}))(CONNECT_DATA=(SERVER=DEDICATED)(SERVICE_NAME=\${rdbms.master.	TNS Connect Descriptor of the ODI Master Repository Database Connection			None	
rdbms.master.username	ALM_MASTER_11	User Name of the ODI Master Repository Database Connection	✓		None	
restore.odi.dependencies	false	Flag to indicate whether to also restore the dependencies of the ODI objects (listed in .shadow files) or not	✓		None	
topology.context.restore	true	Flag to indicate whether to restore the contexts or not	✓		None	
topology.logical.restore	true	Flag to indicate whether to restore the logical topology or not	✓		None	
topology.physical.restore	true	Flag to indicate whether to restore the physical topology or not	✓		None	

24 items found, displaying all

Required Information	Parameter Name	Your Value
ODI_HOME Location	odi.home	
ODI User Name	odi.user.name	
ODI User Password	odi.user.passwd	
The ODI Work Repositories you use for the different environments	odi.workRepository.name	
Password of the ODI Master Repository Database Connection	rdbms.master.passwd	

Required Information	Parameter Name	Your Value
SID of the ODI Master Repository Database Connection	rdbms.master.sid	
User Name of the ODI Master Repository Database Connection	rdbms.master.username	

3.4. Other Oracle Initialization Values

For the following parameters, we use the Oracle initialization values.

If your value is the same as the initialization value, no modification will be required. If not, you will have to modify their initialization value. How you do this will be explained later on, in the section [Specifying the Initialization Values for the Parameters \(Global Administration\)](#) (page 51).

Required Information	Parameter Name	Default Value	Your Value
Hostname of the ODI Master Repository Database Connection	rdbms.master.host	localhost	
TCP port of the ODI Master Repository Database Connection	rdbms.master.port	1521	

CHAPTER 4

Getting Started

You can either download the required files from our website or from our FTP Server.

4.1. Downloading the zip files from our website

You need to download the following files from our website:

- The LCM framework (IKAN ALM)
- LCM for ODI (for the ODI version you are using)

THE IKAN ALM Demo folder

This folder contains the installation files for the IKAN ALM 5.7 Demo version: *ikanalm_demo_x64.exe*.

The ODI_XX folder (XX being the version of ODI you are using (11gR1, 12cR1 or 12cR2))

This folder contains the IKAN ALM Solution Phases for ODI and the installation files for the VCR4ODI connector:

- SolutionPhases_ODI_XX folder
 - com.ikanalm.phases.database.odi.copyODIFiles-X.Y.Z.jar
X.Y.Z being the release ID of the Phase. This Phase is the same for all versions of ODI.
 - com.ikanalm.phases.database.odi.restoreODIxxObjects-X.Y.Z.jar
XX being the version of ODI and X.Y.Z being the release ID of the Phase.
- VRCR4ODI_XX.zip
- Documentation
 - LCM4ODI_GettingStartedGuide_4.2.pdf
 - VCR4ODI_InstallationGuide_4.0.pdf

4.2. Downloading the “LCM4ODI_trial” folder from the FTP Server

Download the following folders from our FTP Server:

THE IKAN ALM Demo folder

This folder contains the installation files for the IKAN ALM 5.7 Demo version: *ikanalm_demo_x64.exe*.

The ODI_XX folder (XX being the version of ODI you are using (11gR1, 12cR1 or 12cR2))

This folder contains the IKAN ALM Solution Phases for ODI and the installation files for the VCR4ODI connector:

- SolutionPhases_ODI_XX folder
 - com.ikanalm.phases.database.odi.copyODIFiles-X.Y.Z.jar
X.Y.Z being the release ID of the Phase. This Phase is the same for all versions of ODI.
 - com.ikanalm.phases.database.odi.restoreODIxxObjects-X.Y.Z.jar
XX being the version of ODI and X.Y.Z being the release ID of the Phase.
 - LCM4ODI_GettingStartedGuide.pdf
- VRCR4ODI_XX.zip
- VCR4ODI_InstallationGuide.pdf (full version)

4.3. Let's get started ...

INSTALLING IKANALM

Installing the IKAN ALM Demo Version

5.1. Installing the Software

Note: A Java Development Kit (JDK) version 1.8 or higher is required. Java Runtime Environment (JRE) will not be sufficient!

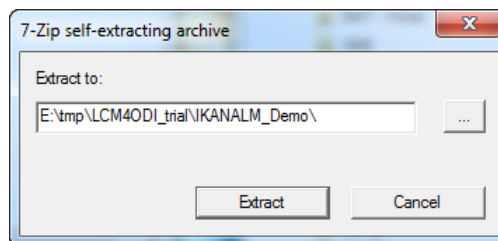
Step 1: Start the Installation

The IKANALM_Demo folder you downloaded from our FTP server contains a self-extractable archive (ikanalm_trial_x64).

Note: If you downloaded the "IKAN ALM Demo folder" from our website, you can skip steps 1 and 2 and immediately execute the installer.

1. Double-click or execute this archive.

A window similar to this one will pop up:



2. Choose a destination folder.

The files will be extracted in a subfolder named "ikanalm_trial_x64".

After extraction, the folder will contain:

- alm_license.lic: a Demo license (valid for 30 days) for IKAN ALM
- almdemo_install_5-7_x64.exe

The 64-bit installer image for IKAN ALM Demo

- IKAN ALM 5.7 Windows Demo Installation Guide.pdf: full installation guide for installing IKAN ALM Demo

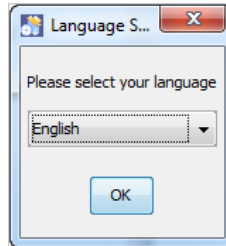
3. Execute the installer.

The graphical IKAN ALM installer will start. Refer to the section *Troubleshooting* in the full *IKAN ALM Windows Demo Installation Guide* if the IKAN ALM installer does not start.

Depending on the flavor of Windows OS used, there can be differences in the installer's behavior.

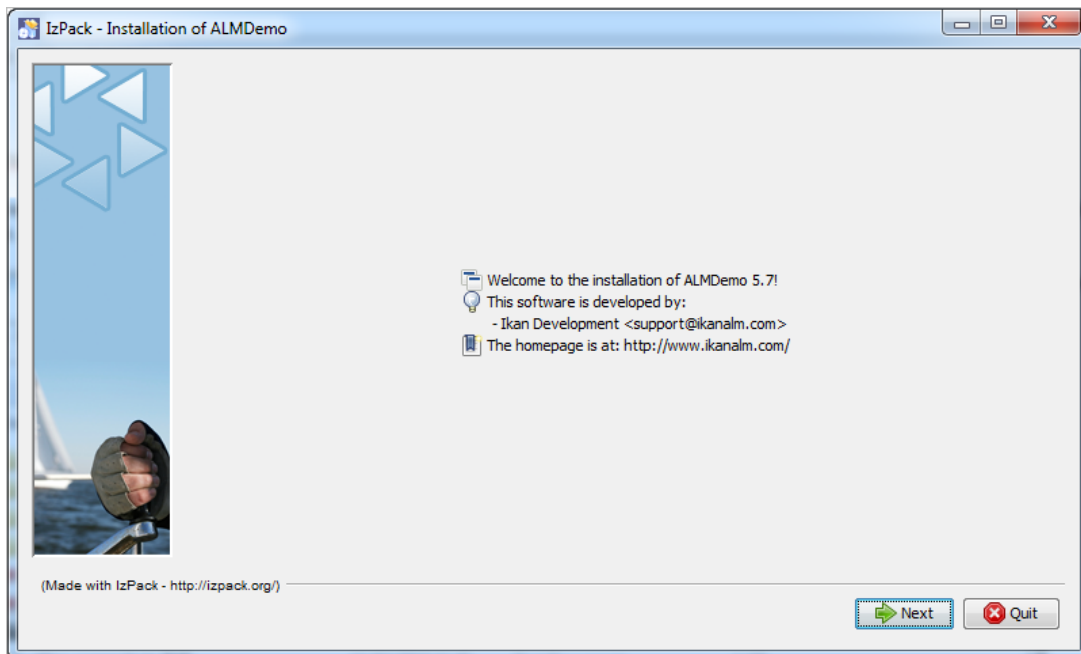
If UAC is enabled (Windows Vista and later), Windows will ask for a confirmation for the program to make changes to the computer (if your User account is an Administrator). If you are trying to install the application from an account other than Administrator, you might get the message "requested operation requires elevation."

Double-click the `almdemo_install_5.7_x64.exe` file. The following window will pop up:

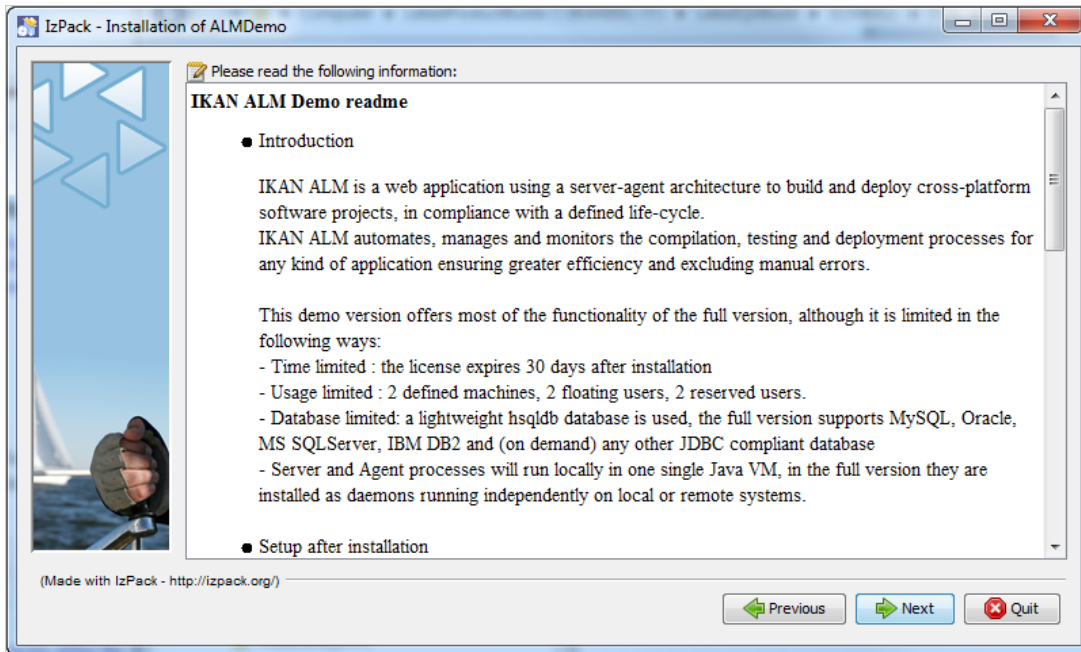


Select your preferred installation language and click *OK*.

The following welcome screen will be displayed:

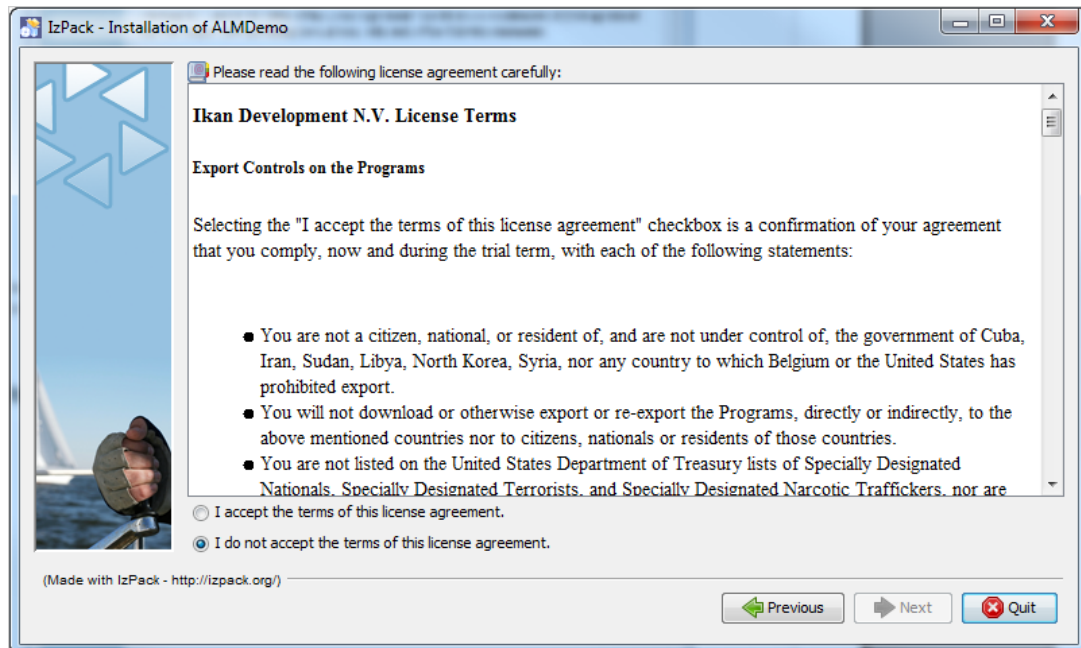


Click *Next* to continue.



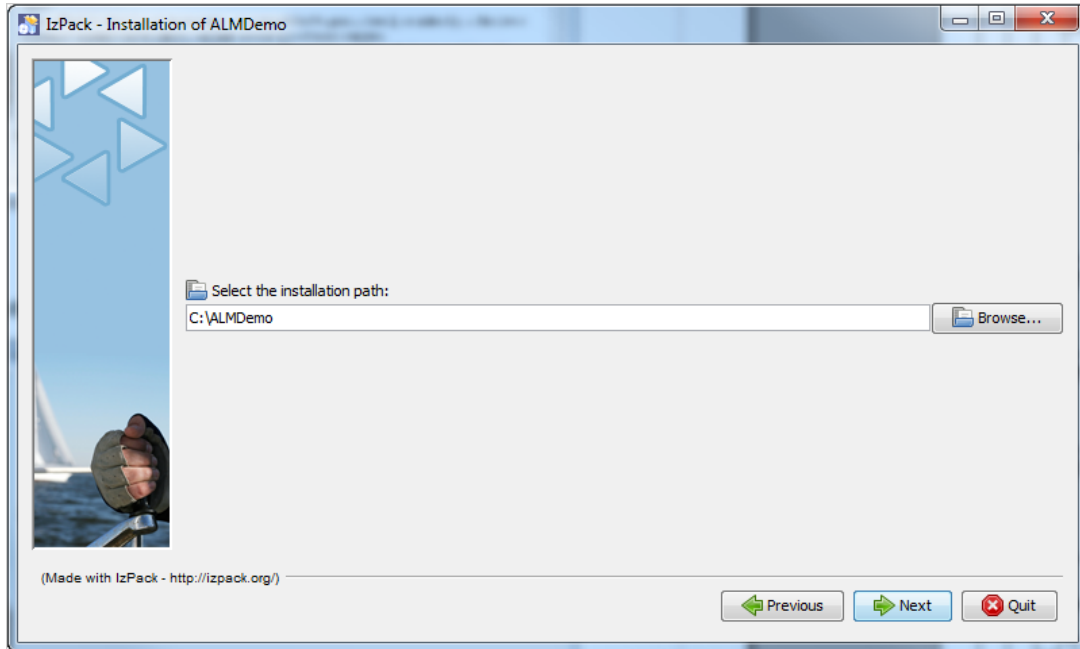
Carefully read the readme information. Click the *Next* button to continue.

Step 2: License Agreement



Read the license agreement carefully. Select the option *I accept the terms of this license agreement*. and click the *Next* button.

Step 3: Installation Location

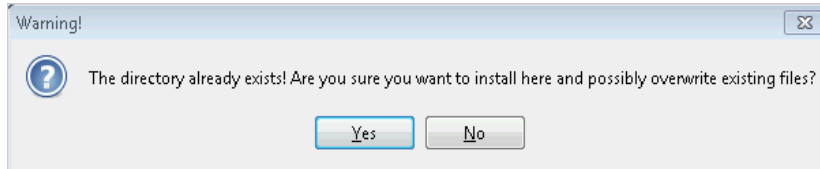


Select the home directory for installing the Demo Application on your file system. The default home directory is C:\ALMDemo. You can change this location using the *Browse* button.

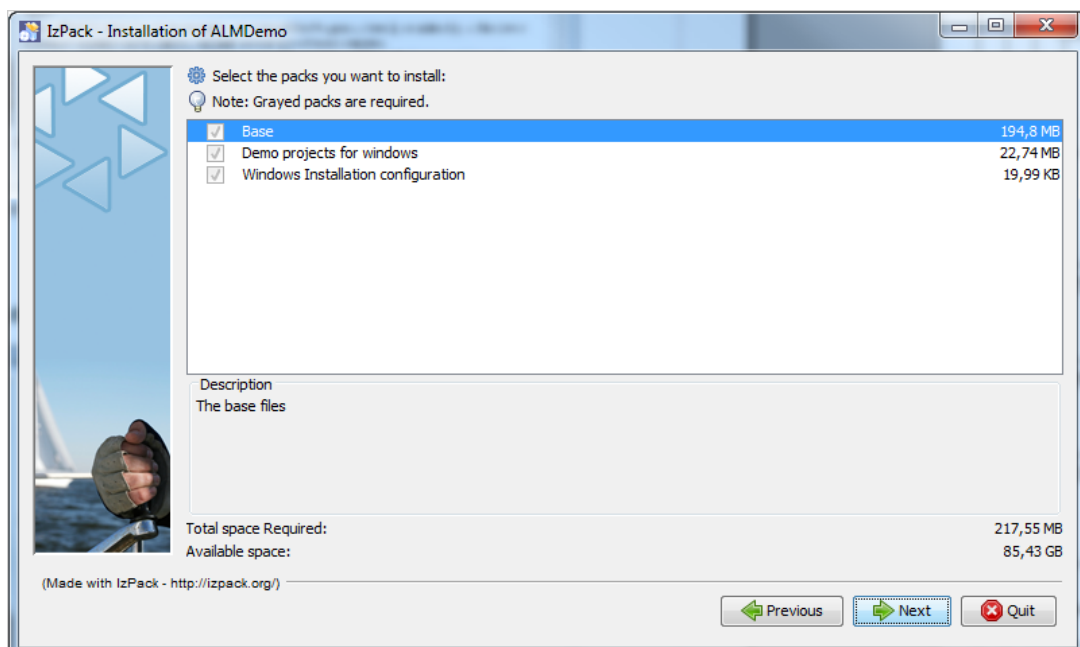
Further on in this document, this location will be referred to as *ALM_HOME*.

Click *Next* to continue.

If the directory already exists, a warning message will appear.

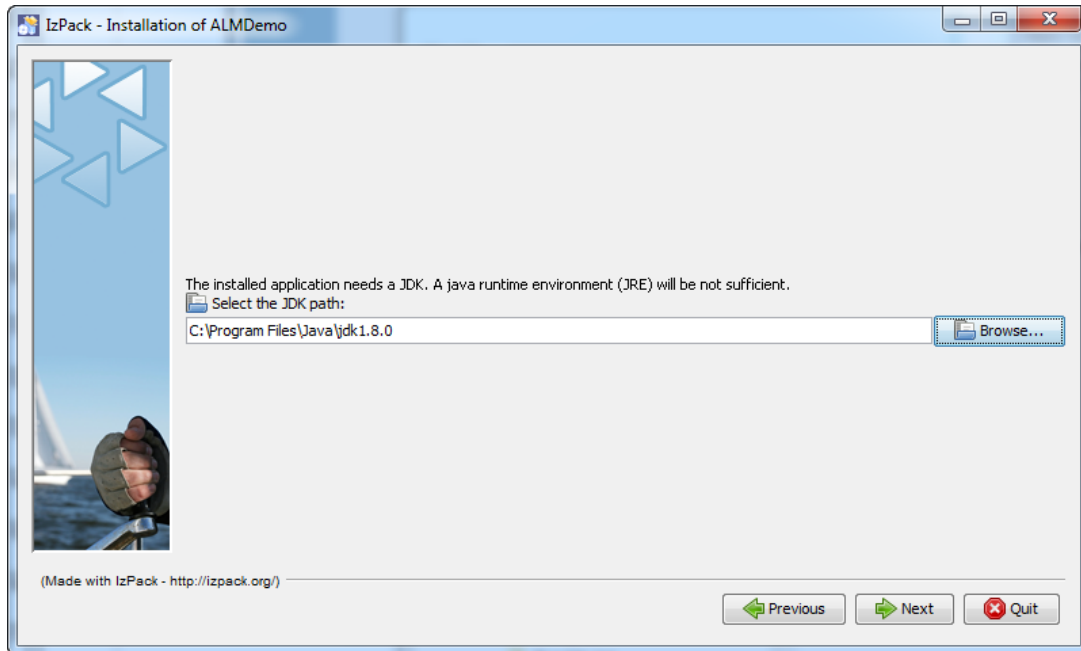


All installation packs must be installed.



Click *Next* to proceed to the selection of a Java Development Kit (JDK).

Step 4: Select a Java Development Kit

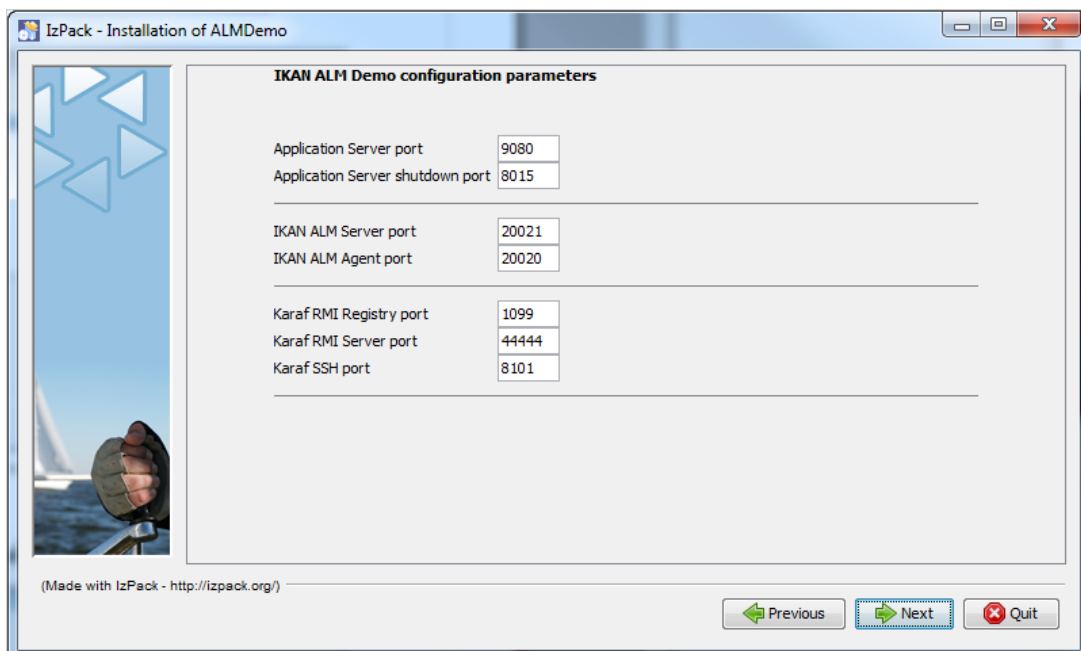


Note: A Java Development Kit (JDK) version 1.8 or higher is required. Java Runtime Environment (JRE) will not be sufficient!

If the selected directory does not contain a proper JDK, the following error message will be displayed:



Step 5: Configuration Settings

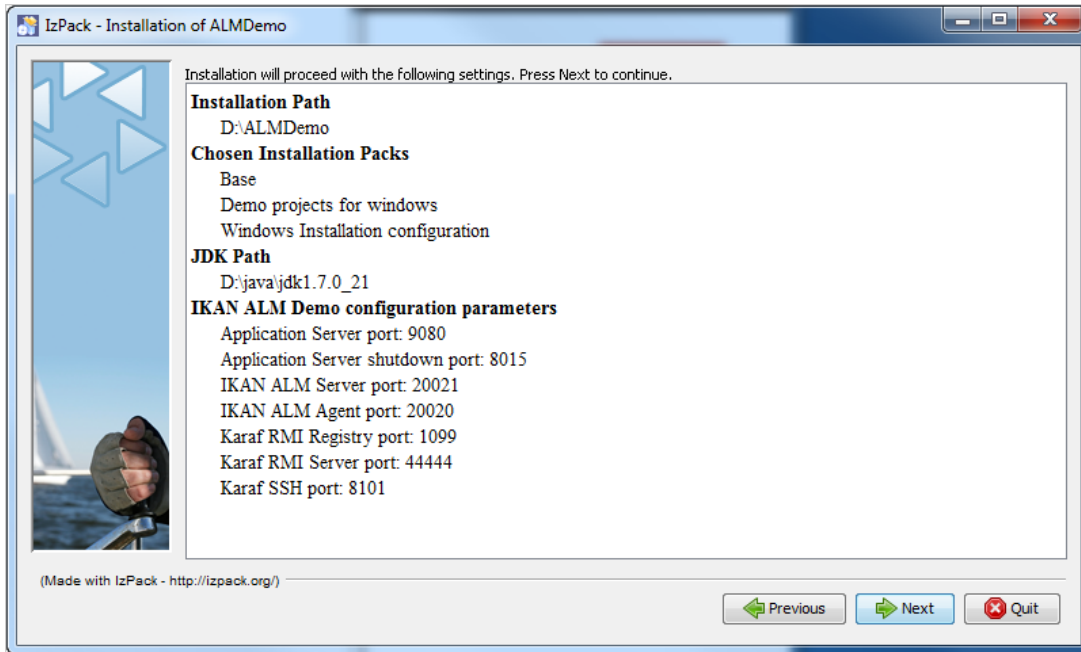


Configure the network ports used by IKAN ALM. Normally, the defaults presented should be OK, but it is important that these port numbers are not in use by other processes.

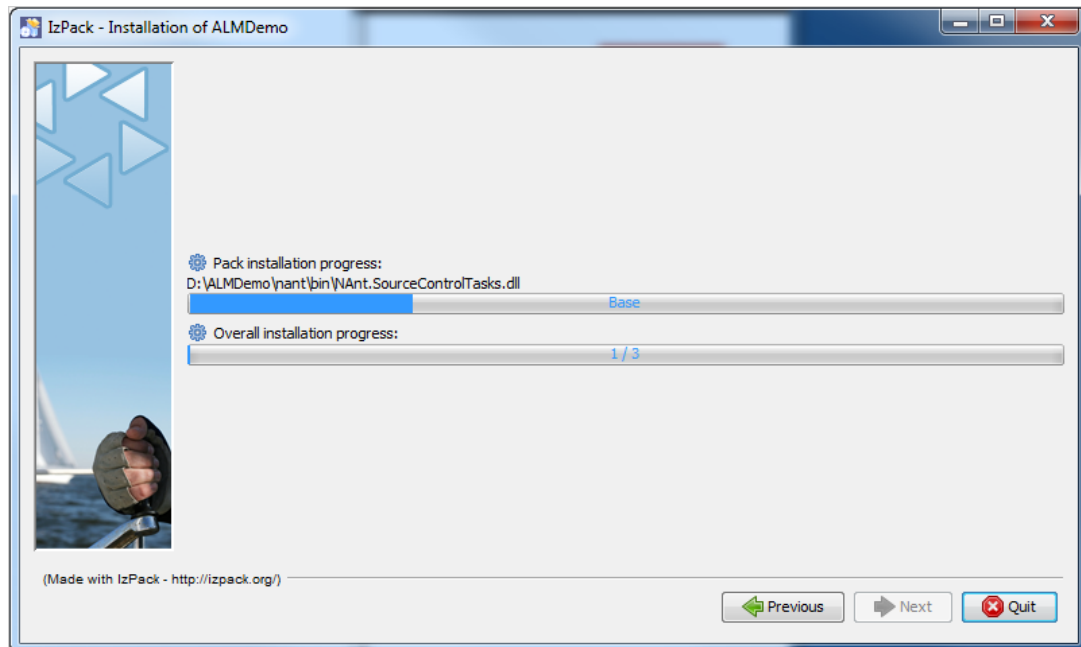
Get to a command prompt and type `netstat -an` to check which ports are currently in use.

In the remainder of this guide, the Application Server Port will be referred to as `ALM_APPSERVER_PORT`.

Click *Next* to obtain an installation summary.

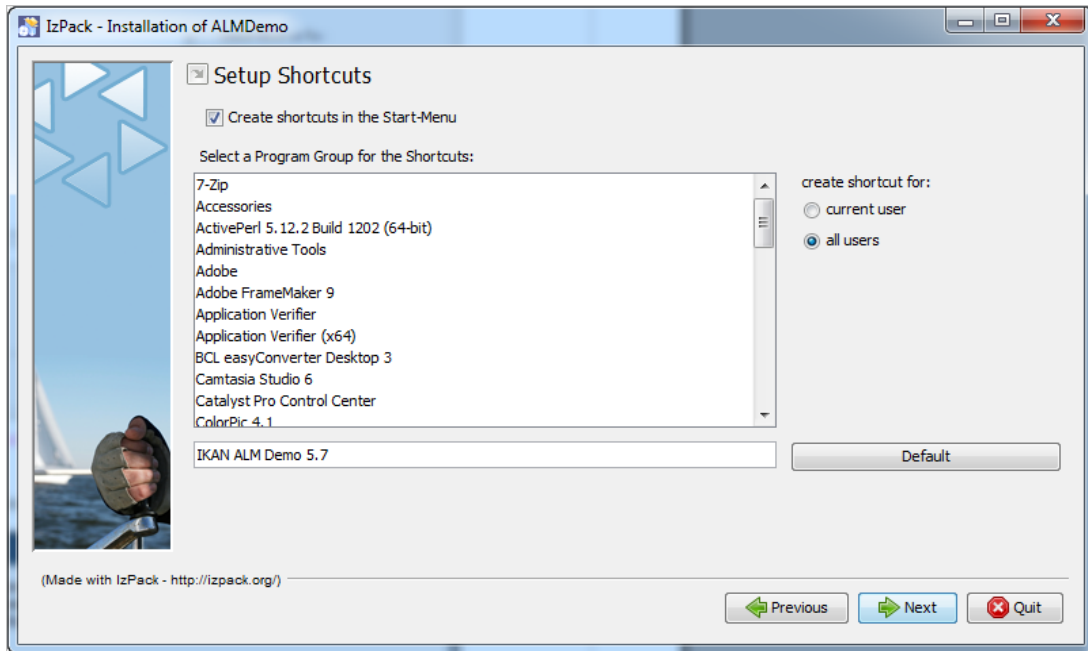


Click *Next* to start the installation with the given settings:



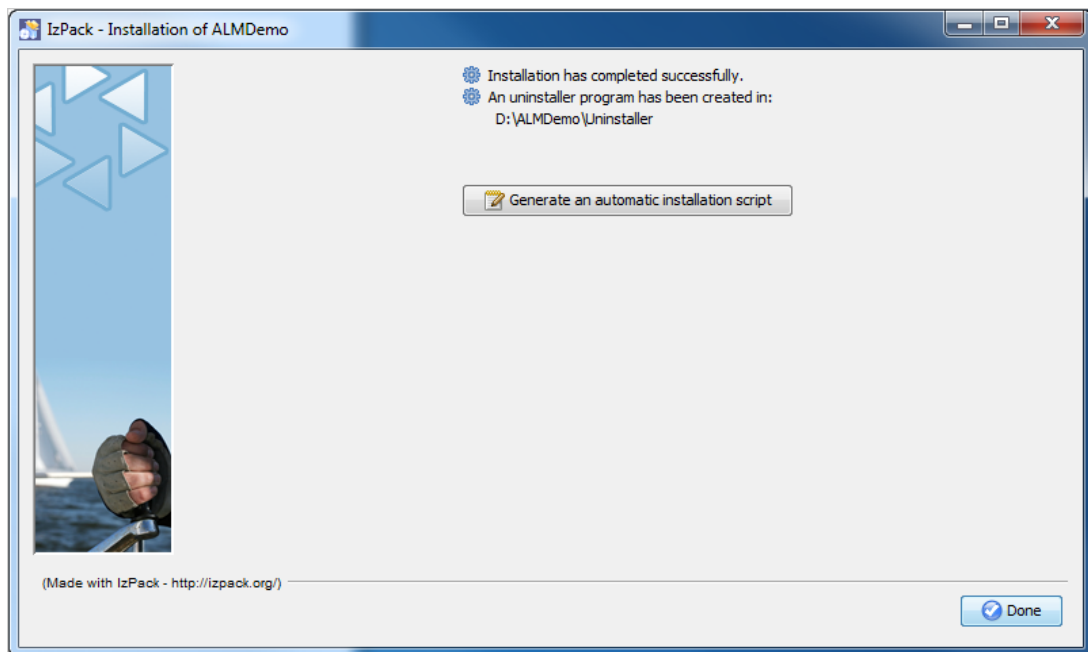
Once the installation is finished, click *Next* to proceed with setting up the Start Menu.

Step 6: Start Menu Shortcuts



Select the Program Group and Shortcut Name for the Start Menu. The default group and name is *IKAN ALM Demo 5.7*. You may change these names.

Click *Next* to complete the IKAN ALM Demo installation.



5.2. Install the IKAN ALM License

Together with this IKAN ALM Demo Installation, you have received a license file named *alm_license.lic*. This demo license will expire after 30 days (See [Step 2: License Agreement](#) on page 15.). Contact IKAN ALM support if you want to extend the license period.

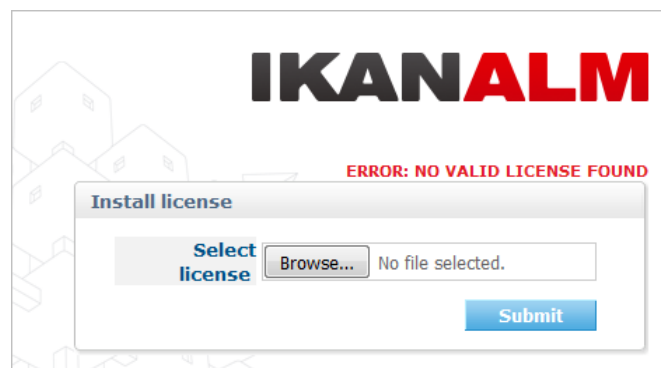
Start the IKAN ALM Demo Server:

- Via the Start Menu:
Applications > IKAN ALM Demo 5.7 > Start IKAN ALM Demo server
- Or via the batch file:
Double click the *ALM_HOME/startDemo.cmd* file.

Open the IKAN ALM web application:

- Via the start menu:
Start > Programs > IKAN ALM Demo 5.7 > Open IKAN ALM in browser
- Or via a web browser:
Provide the following URL: *http://MACHINE_NAME:ALM_APPSERVER_PORT/alm* (e.g., *http://ikan009:9080/alm*)

The *Install license* screen will appear.



Enter or browse to the file location of the *alm_license.lic* file you received and click *Submit*. Do not alter the file or filename. The following screen will appear when the license is installed:



5.3. Start

The IKAN ALM Demo Server must be running in order to work with IKAN ALM.

Start the IKAN ALM Demo Server:

- Via the shortcut menu:
Start > Programs > IKAN ALM Demo 5.7 > Start IKAN ALM Demo server
- Or via the batch file:
Double click the *ALM_HOME/startDemo.cmd* file.

This will launch a Tomcat webserver and the IKAN ALM Demo server (which runs both the IKAN ALM Server and Agent). When the IKAN ALM Demo server prompt window displays a message like: "Successfully connected with IKAN ALM server", the ALM Demo is fully started.

```

Select Start IKAN ALM Demo server
[exec] 2013-08-22 10:22:36,982 ; INFO ; date: pid=agent) ; AgentDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; IKAN ALM Agent started
[exec] 2013-08-22 10:22:36,983 ; INFO ; date: pid=agent) ; AgentDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; Release 5.5 - Build 30 - Ucr
Tag B_5-5_0_h30
[exec] 2013-08-22 10:22:36,985 ; DEBUG ; pool-10-thread-1 ; AgentDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; Testing connection to IKAN ALM
server, serverhost=localhost, serverport=20021
[exec] 2013-08-22 10:22:37,012 ; INFO ; pool-10-thread-1 ; AgentDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; Successfully connected with
IKAN ALM server
[exec] 2013-08-22 10:22:37,015 ; INFO ; LM BuilderDaemon ; BuilderDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; Agent Machine information :
hostname=IKAN026 ipaddress=192.168.253.49
[exec] 2013-08-22 10:22:37,016 ; INFO ; M DeployerDaemon ; DeployerDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; Agent Machine information :
hostname=IKAN026 ipaddress=192.168.253.49
[exec] 2013-08-22 10:22:37,016 ; DEBUG ; pool-10-thread-1 ; AgentDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; Cancelled the scheduled serv
er connect job
[exec] 2013-08-22 10:22:37,067 ; INFO ; LM BuilderDaemon ; BuilderDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; IKAN ALM BuilderDaemon start
ed
[exec] 2013-08-22 10:22:37,067 ; INFO ; M DeployerDaemon ; DeployerDaemon
; 79 - be.ikan.lib.util - 4.0.0 ; IKAN ALM DeployerDaemon star
  
```

If a message such as "INFO : Server startup in nnn ms", the Tomcat Server is completely up and running and you can use the Web application.

```

Tomcat
[Server@ee32bd0]: [Thread[IKAN ALM Hsqldb thread,5,main]]: checkRunning(false) e
xited
[Server@ee32bd0]: [Thread[IKAN ALM Hsqldb thread,5,main]]: setDatabasePath(C:\fil
e:D:\ALMDemo\database\hsqldb\almDemo)
[Server@ee32bd0]: [Thread[IKAN ALM Hsqldb thread,5,main]]: setNoSystemExit(true)
[Server@ee32bd0]: [Thread[IKAN ALM Hsqldb thread,5,main]]: setTrace(false)
aug 22, 2013 10:22:24 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory D:\ALMDemo\appServer\webapps\docs
aug 22, 2013 10:22:24 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory D:\ALMDemo\appServer\webapps\examples
aug 22, 2013 10:22:24 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory D:\ALMDemo\appServer\webapps\host-mana
ger
aug 22, 2013 10:22:24 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory D:\ALMDemo\appServer\webapps\manager
aug 22, 2013 10:22:24 AM org.apache.catalina.startup.HostConfig deployDirectory
INFO: Deploying web application directory D:\ALMDemo\appServer\webapps\ROOT
aug 22, 2013 10:22:24 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["http-bio-9080"]
aug 22, 2013 10:22:24 AM org.apache.coyote.AbstractProtocol start
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
aug 22, 2013 10:22:24 AM org.apache.catalina.startup.Catalina start
INFO: Server startup in 10481 ms
  
```

Note: After having terminated this *Getting Started Guide*, you should stop the IKAN ALM Demo Server: Via the shortcut menu: *Start > Programs > IKAN ALM Demo 5.7 > Stop IKAN ALM Demo server* or via the batch file: *ALM_HOME/stopDemo.cmd*

5.4. Open the IKAN ALM Web Application

If you installed the shortcuts:

Start > Programs > IKAN ALM Demo 5.7 > Open IKAN ALM in browser

If you did not install the Start Menu shortcuts do the following:

1. Start an internet browser

2. Provide the following URL: *http://MACHINE_NAME:ALM_APPSERVER_PORT/alm*

Note: *MACHINE_NAME* may be replaced by the machine name or IP address, e.g., 127.0.0.1, IKAN001, ...) of the system where you installed IKAN ALM demo. *ALM_APPSERVER_PORT* as configured in section [Step 5: Configuration Settings](#) (page 17). For example, an IKAN ALM start URL could be *http://ikan009:9080/alm*.

An internet browser window will appear, displaying the IKAN ALM *Log in* screen.

5.5. Log on to the IKAN ALM Web Application

You are ready to start working with the IKAN ALM web interface.



You may add this link to your internet browser's Favorites.

Log on to IKAN ALM with the following predefined User ID and password:

User ID	Password	Access Rights
global	global	Global Administration

Provide the User ID and Password, and select *Log in*. The IKAN ALM home page will be displayed and you can start working in IKAN ALM.

Next, we will set up the Subversion repository.

SETTING UP SUBVERSION

Setting Up Subversion

6.1. Setting up the Subversion Versioning System

The IKAN ALM Demo installation includes a Subversion installation as well as an SVN repository holding source files for some IKAN ALM demo projects.

As far as the Life Cycle Management solution for ODI is concerned, these demo projects are irrelevant. You can however use that repository for this LCM4ODI_TRIAL, provided you perform the following steps:?

1. Create a new folder that you have write access to.

For the purpose of this *Getting Started Guide*, we will call this folder *MyFolder*.?

2. Create the following structure in *MyFolder*:

MyProject

MyProject/trunk

MyProject/tags

MyProject/branches

IMPORTANT!

In this example, *MyProject* is the VCR Project name. This name can be replaced by a name of your choice.

When creating the Subversion definition in IKAN ALM, you will need to specify the VCR Project name you have chosen in this step!

3. Import the created *MyFolder* into Subversion

- 3.1. Open a command prompt and change the directory to *MyFolder*.

```
CD ...\MyFolder
```

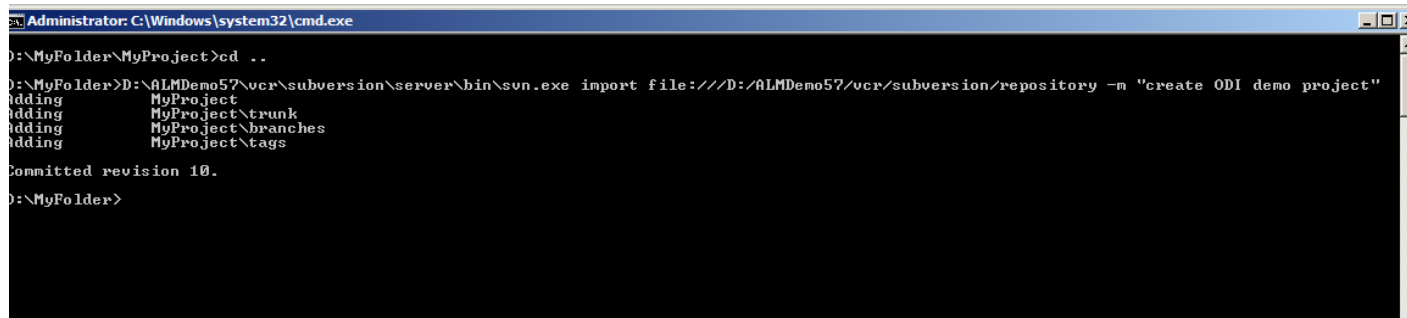
- 3.2. Execute the following command to import the folder into Subversion

```
<ALM_HOME>\vcr\subversion\server\bin\svn.exe import file:///<ALM_HOME>/vcr/subversion/repository -m "create ODI demo project"
```

(in which *<ALM_HOME>* stands for the IKAN ALM demo installation location)?

IMPORTANT!

Carefully copy the slashes exactly as mentioned above. The first part of the command is a Windows command, therefore backslashes should be used!



```
Administrator: C:\Windows\system32\cmd.exe
D:\MyFolder\MyProject>cd ..
D:\MyFolder>D:\ALMDemo57\usr\subversion\server\bin\svn.exe import file:///D:/ALMDemo57/usr/subversion/repository -m "create ODI demo project"
Adding              MyProject
Adding              MyProject\trunk
Adding              MyProject\branches
Adding              MyProject\tags
Committed revision 10.
D:\MyFolder>
```

Note: The Subversion repository included in the IKAN ALM Demo installation is set up as a so-called “project-oriented” repository, and these actions will add a new project for the ODI application to this repository.

INSTALLING THE VCR4ODI CONNECTOR

Installing VCR4ODI

Each ODI User needing *VCR4ODI*, must install the connector in his/her local environment. The installation process consists of extracting the zip file.

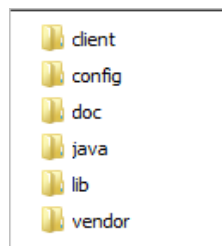
To complete the installation process, you will need the following information:

- the installation zip file
- the location of the ODI_HOME and ORACLE_HOME directories.

1. Copy the *VCR4ODI_xx.zip* file to the location where you want to install VCR4ODI.
2. Extract the zip file.

Note: *VCR4ODI* is shipped with a Java version so that the connector can be started immediately.

The following folders will be extracted:



Folder	Description
client	The <i>bin</i> folder inside this folder contains the <i>vcr4odi.bat</i> file, which launches the connector.
config	Contains the Log file, the configuration files (to connect ODI to the VCR) and the License file.
doc	Contains the documentation.
java	The Java installation shipped with VCR4ODI.
lib	Contains the core Java application.
vendor	Contains the external jars.

- All applications depend on the setting of the `%ODI_HOME%` variable in the `setEnv.bat` file. Therefore, you must first of all specify the ODI home directory.

3.1. Go to the `config` folder and open the `setEnv.bat` file using a text editor.

3.2. Specify the ODI and Oracle Home directories.

Note: Specifying `ORACLE_HOME` is only needed if the Oracle DB that is being used for the ODI repositories requires connections via TNS. In that case, Oracle client software must be installed on the developer machine running VCR4ODI, and the `ORACLE_HOME` variable must point to this installation. If the method for connecting to the ODI repository is the standard JDBC user/pwd method, the Oracle client software is not required and the `ORACLE_HOME` variable can be left unset.

```

16 REM # VARIABLES DESCRIPTION
17 REM #
18 REM # The following variables can be changed to tune VCR4ODI configuration.
19 REM #
20 REM #   VCR4ODI_HOME
21 REM #       Installation directory of VCR4ODI software
22 REM #   VCR4ODI_INIT_HEAP
23 REM #       Initial java machine heap size used by VCR4ODI.
24 REM #   VCR4ODI_MAX_HEAP
25 REM #       Maximum java machine heap size used by VCR4ODI.
26 REM #   VCR4ODI_JAVA_HOME
27 REM #       Installation directory of the java virtual machine. Java 1.7 and above is necessary
28 REM #   ODI_HOME
29 REM #       Installation directory of ODI software
30 REM #
31
32 REM #
33 REM # VARIABLES
34 REM #
35
36 set ODI_HOME="C:/oracle/product/ODI_11117"
37 set ORACLE_HOME="C:/oracle/product/11.1.0/db_1"
38 set JPS_CONFIG="%ODI_HOME%/oracledi/client/odi/bin/jps-config.xml"
39
40

```

Specify the ODI and Oracle Home Directories


- Save and close the file.
- Go to the `client/bin` folder and execute the `vcr4odi.bat` file to start VCR4ODI. The VCR4ODI startup window will be displayed.



6. Start the VCR4ODI Connector

- If this is the first time you use the VCR4ODI connector, you first need to define a Profile.
In that case, continue to the section [Creating a VCR4ODI Profile](#) (page 30).
- If a Profile has already been defined, select it from the drop-down list and click the *Start* button.
In that case continue to the section [Starting VCR4ODI](#) (page 34).

Creating a VCR4ODI 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
- the ODI parameters: all parameter necessary to connect to the master and to the repository,
- the Subversion connection information

2. Provide a name for the new Profile.

3. Update the “ODI Parameters” panel as required.

Oracle Data Integrator Connection

Field	Meaning
User	The name of the ODI User.
Password	The password of the ODI 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 “SVN Connection parameters” panel as required.

Field	Meaning
Repository URL	The URL path of the Subversion Repository. This path is composed as follows: <code>file:/// <ALM_HOME>/vcr/subversion/repository/<MyProject>/trunk</code> In which <code><ALM_HOME></code> stands for the IKAN ALM demo installation location) and <code><MyProject></code> for the VCR Project name you created earlier (Setting Up Subversion (page 24))? The path to the repository can also be found in the Version Control Repository definition in the Global Administration section in IKAN ALM (Subversion Definition (page 36)).
User ID	As we are using the Subversion file protocol to establish the connection, a User ID is not required.
Password	As we are using the Subversion file protocol to establish the connection, a Password is not required.
Working Copy	The path to a 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. Important: each user should have his/her own local working repository.

5. The completed form will look similar to the one below.

VCR4ODI Configuration Details

VCR4ODI

Profile: LCM4ODI_trial

User: vcr4odi

Password: ●●●●●●

Oracle Data Integrator Connection

User: SUPERVISOR

Password: ●●●●●●

Database Connection (Master Repository)

User: ALM_MASTER_11

Password: ●●●●●●●●

Driver Name: oracle.jdbc.OracleDriver

URL: jdbc:oracle:thin:@localhost:1521:orcl1107

Work Repository

Work Repository: ALM_WORK_11_DEV

Subversion Connection

Repository URL: file:///D:/ALMDemo57/vcr/subversion/repository/MyProject/trunk

User:


Password:

Working Copy: E:/LCM4ODI_WorkingCopy/

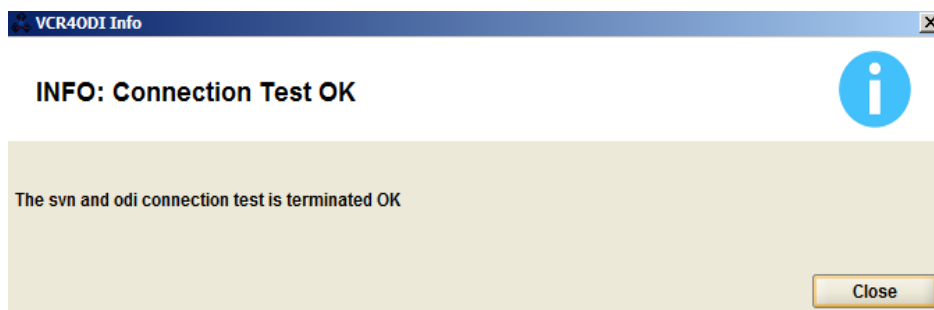
LCM4ODI_trial [gear icon] OK Cancel

This Repository URL can be found in the Subversion definition in IKAN ALM.

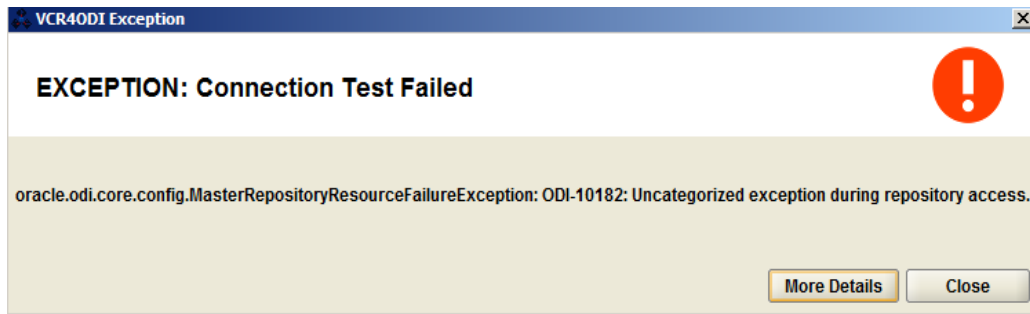
6. Test the connection.

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.

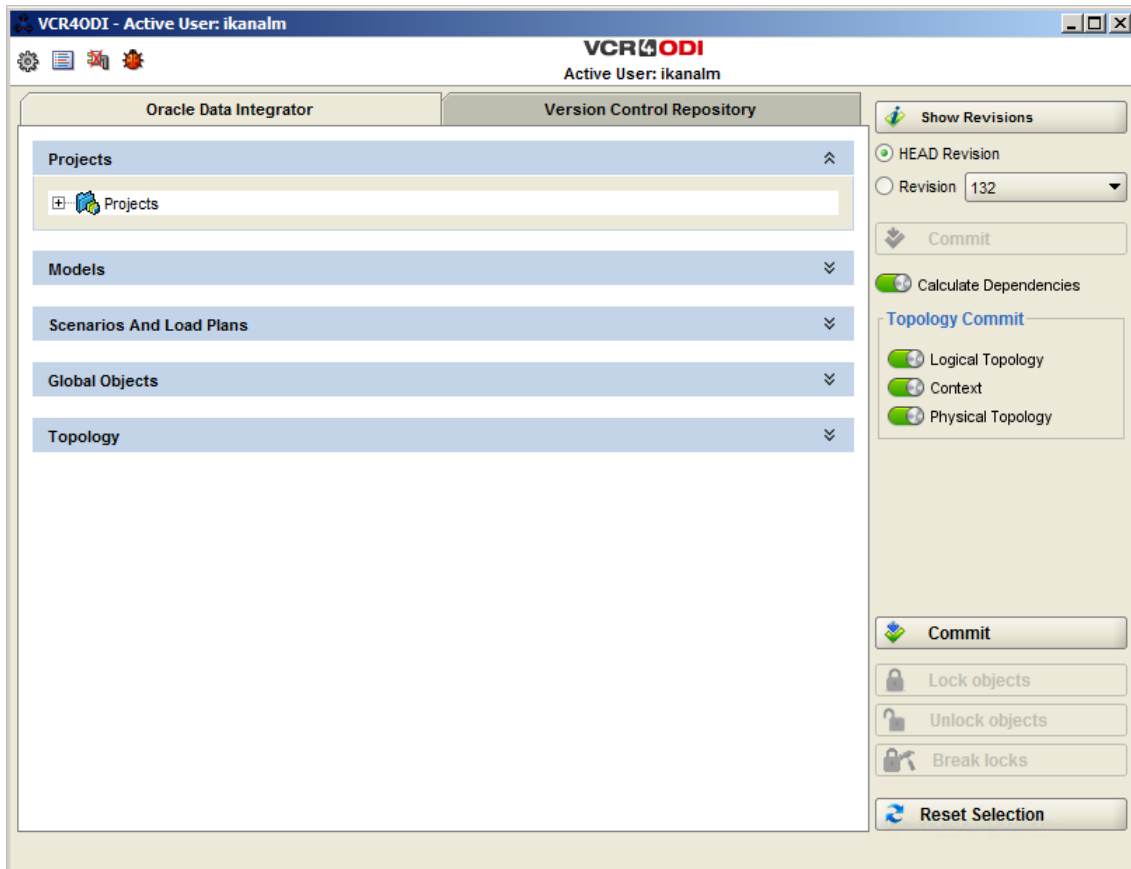


You can click the *More Details* button for more information concerning the connection problems.

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

Starting VCR4ODI

After having clicked the *Start* button, the following screen is displayed.



This screen contains two tab pages:

- The *Oracle Data Integrator* page
This tab page is used to commit modified ODI objects from a selected ODI Repository into Subversion.
- The *Version Control Repository* page
This page is used to restore objects from Subversion into the selected ODI Repository.

Next, we will start using IKAN ALM and create a first project.

USING THE LCM4ODI SOLUTION

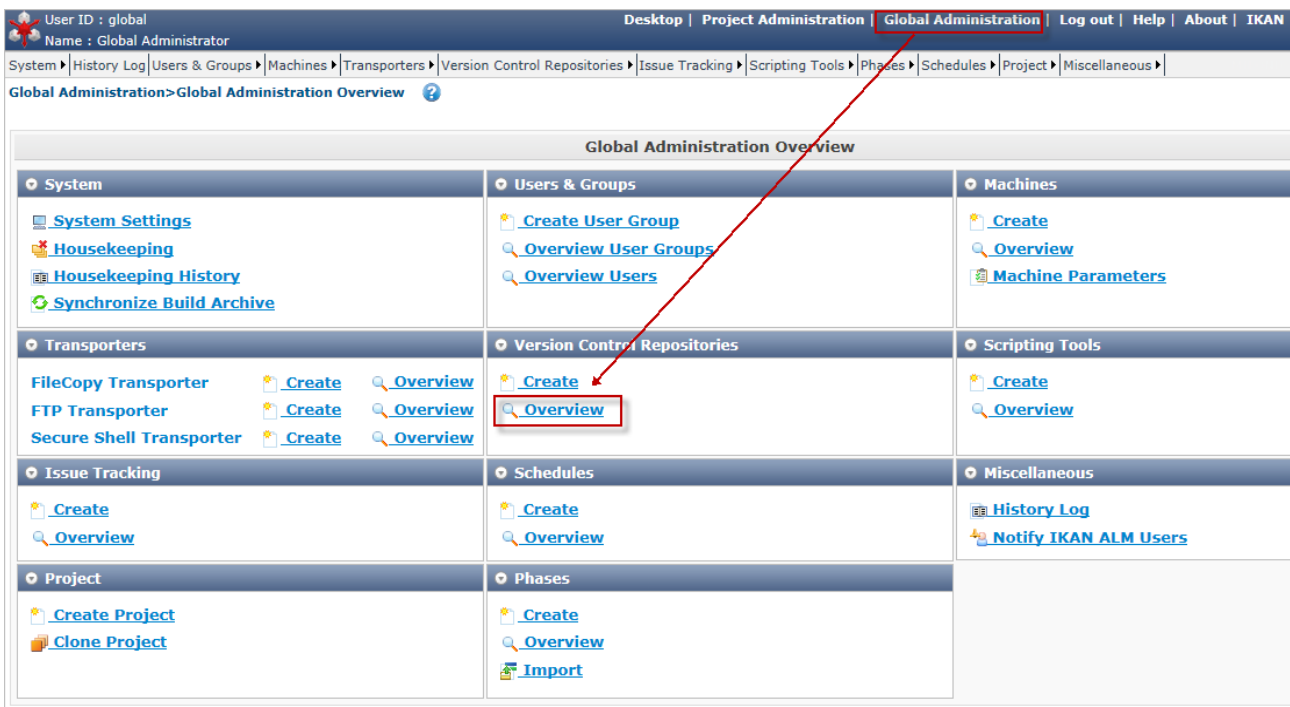
Creating a Project

First, we will:

- verify the definition of the Subversion repository
- and, next, create our first project.

10.1. Subversion Definition

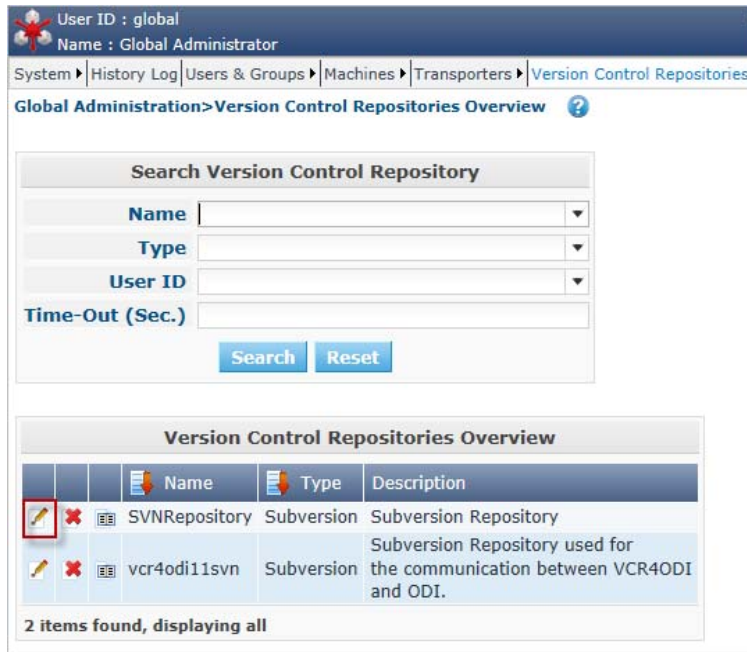
1. Go to the *Global Administration* section.
2. Select *Version Control Repositories > Overview*



The screenshot shows the IKAN Global Administration interface. The top navigation bar includes 'User ID : global', 'Name : Global Administrator', and tabs for 'Desktop', 'Project Administration', 'Global Administration', 'Log out', 'Help', 'About', and 'IKAN'. Below the navigation bar, a breadcrumb trail reads 'System > History Log > Users & Groups > Machines > Transporters > Version Control Repositories > Issue Tracking > Scripting Tools > Phases > Schedules > Project > Miscellaneous'. The main content area is titled 'Global Administration Overview' and contains a grid of sections. The 'Version Control Repositories' section is expanded, showing a 'Create' link and an 'Overview' link. The 'Overview' link is highlighted with a red box, and a red arrow points from the 'Global Administration' tab to it.

The Version Control Repositories Overview will be displayed.

3. Click the  *Edit* icon in front of the *SVNRepository*.



User ID : global
Name : Global Administrator

System | History Log | Users & Groups | Machines | Transporters | Version Control Repositories

Global Administration > Version Control Repositories Overview

Search Version Control Repository



Name

Type

User ID

Time-Out (Sec.)

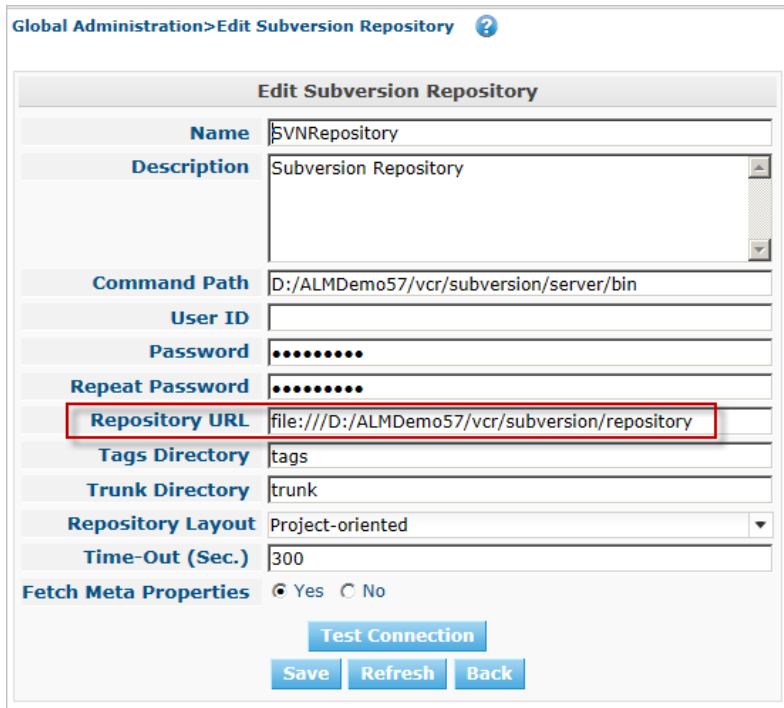
Version Control Repositories Overview

	Name	Type	Description
	SVNRepository	Subversion	Subversion Repository
	vcr4odi11svn	Subversion	Subversion Repository used for the communication between VCR4ODI and ODI.

2 items found, displaying all

The *Edit Subversion Repository* window will be displayed.

On this screen, you will find the URL to the Repository containing your VCR Project (in our example: MyProject).



Global Administration > Edit Subversion Repository

Edit Subversion Repository

Name

Description

Command Path

User ID

Password

Repeat Password

Repository URL

Tags Directory

Trunk Directory

Repository Layout

Time-Out (Sec.)

Fetch Meta Properties Yes No

Note: This Repository URL is the same as the one used in the VRC4ODI Profile.

10.2. Create your First Project

Now we can start with the actual creation of our Project.

1. In the *Global Administration* section, select *Project > Create Project* and fill out the fields as required.

Global Administration > Create Project ?

Create Project

Project Settings		Head Project Stream Settings	
Project Type	Release-based	Build Prefix	REL
Name	MyProject	Status	Under construction
Description	Demo Project for the LCM4ODI Solution	Description	
VCR	SVNRepository	Locked	<input type="radio"/> Yes <input checked="" type="radio"/> No
VCR Project Name	MyProject	Hidden	No
Issue Tracking System		Tag-Based	<input type="radio"/> Yes <input checked="" type="radio"/> No
Build Tool Type	ANT	Build Type	Full Build
Deploy Tool Type	ANT	Accept Forced Build	<input checked="" type="radio"/> Yes <input type="radio"/> No
Build Script	build.xml	Tag Template	\${streamType}_\${prefix}_b\${buildNumber}
Deploy Script	deploy.xml	VCR Branch ID	
Locked	Yes		
Hidden	No		

Project Security Settings (optional)

Check Project Name in the VCR

- 1.1. Specify the Subversion repository containing your project and its project name in the VCR.

Important: The name of the project in the SVN repository is case-sensitive!

- 1.2. Choose *ANT* for the Build and Deploy Tool Type. Together with the Project, a Head Project Stream is automatically created that points to the trunk in your project in Subversion.

- 1.3. Provide a Build Prefix.

2. Click the *Create* button to create the project.

Note: Before creating the Project, you can use the *Check Project Name in the VCR* button to check whether the Project Name exists in the VCR. Do not forget to click the *Create* button afterwards.

The BASE Life Cycle

At the moment you created the project, the BASE Life Cycle, which is linked to the Subversion trunk, has been created automatically. This Life Cycle will be used for ongoing development.

Note: In IKAN ALM, a Life Cycle represents the path/stages that software developments go through before reaching the production stage. A typical Life Cycle is DEV-TEST-UAT-PROD, but IKAN ALM also supports other Life Cycles, e.g., HOTFIX-PROD.

1. On the Main Menu, select *Project Administration*.
The *Projects Overview* screen is displayed.

Project Administration > Projects Overview ?

Search Project


Base Criteria Extended Criteria


Name

VCR Project Name

Show Hidden Projects Yes No All

Projects Overview														
			Name	Description	VCR	VCR Project Name	Project Type	Build Tool Type	Deploy Tool Type	Build Script	Deploy Script	Locked	User Access	Admin Access
			Customers	A simple web application for managing a customers ...	SVNRepository	customers	Release-based	ANT	ANT				ALM User	ALM Project
			Demo	Demo Project for the Getting Started Guide.	vcr4odi11svn	vcr4odi11svn	Release-based	ANT	ANT	build.xml	deploy.xml			
			Monocalendar	Demo project for ALM stored in Subversion	SVNRepository	monocalendar	Release-based	NANT	NANT	default.build			ALM User	ALM Project
			MyProject	Demo Project for the LCM4ODI Solution	SVNRepository	MyProject	Release-based	ANT	ANT	build.xml	deploy.xml			

- Click the  *Edit* icon in front of the name of your project (in our example *MyProject*).
The *Project Info* screen is displayed.

Project Administration>Project Info 

Project Info

Info

Name MyProject

Description Demo Project for the LCM4ODI Solution

Project Type Release-based

Locked Yes

Hidden No

VCR SVNRepository

VCR Project Name MyProject

[Check Project Name in the VCR](#)

Issue Tracking System

Build Tool Type ANT

Deploy Tool Type ANT

Build Script build.xml

Deploy Script deploy.xml

Security Settings (optional)

User Access

Admin Access

[History](#)

[Unlock](#) [Edit](#) [Refresh](#) [Back](#)

Administration

[History Log](#)

[Audit Project](#)

Project Streams

[Create Branch](#)

[Overview](#)

Life Cycles

[Create](#)

[Overview](#)

Levels

[Create Build Level](#)

[Create Test Level](#)

[Create Production Level](#)

[Overview](#)

Build Environments

[Create](#)

[Overview](#)

[Build Parameters](#)

Deploy Environments


[Create](#)

[Overview](#)

[Deploy Parameters](#)

- On the submenu, select *Project Stream > Overview*.

The *Project Streams Overview* screen is displayed. You will notice that when you created the Project, a Project Stream linked to the Subversion trunk has been created automatically.

Project Administration>Project Streams Overview 

Search Branch Project Stream

Base Criteria Extended Criteria

Build Prefix

VCR Branch ID





Show Hidden Project Streams Yes No All

Build Suffix

Life-Cycle

[Search](#) [Reset](#)

Project Streams Overview

	Description	Project Stream Type	Status	Build Prefix	Locked	Tag-Based	Build Type	Partial Build VCR Tag	Highest Build Number	Accept Forced Build	VCR Branch ID	Build Suffix	Life-Cycle	Tag Template
   		Head	Under construction	REL			Full Build	0		<input checked="" type="checkbox"/>			BASE	\${streamType}_\${prefix}_b\${buildNumber}

Next, we will define the required Levels.

Setting up the Build and Deploy Levels

1. On the submenu, select *Life-Cycles > Overview*.
2. On the Life-Cycle: BASE panel, click the *Create Build Level*.
The *Create BUILD Level* screen is displayed.
Fill out the fields as required and click the *Create* button.

The screenshot shows the 'Create BUILD Level' form with the following fields and values:

- Name:** CREATE_ARCHIVE *
- Description:** This Level will create a Release based on the current Subversion trunk's content and will store the result in the IKAN ALM Archive.
- Type:** Build
- Locked:** Yes
- Debug:** Yes No
- Notification Type:** No notification *
- Notification Criteria:** Never *
- Schedule:** (empty dropdown)
- Requester User Group:** (empty dropdown)
- Life-Cycle:** BASE

Buttons at the bottom: Create, Reset, Back


Most fields speak for themselves (let's neglect the Notification, Schedule and Requester fields for now).













Note: Activating the *Debug* option makes it easier to track stuff in the beginning, especially when a Build fails. Once everything runs smoothly, set it to *No*.

The BUILD Level is used for creating the Development Archive.

3. Repeat the previous action to create the Test Levels (Test and Acceptance) using the *Create Test Level* link and the Production Level using the *Create Production Level* link.
TEST, ACCEPTANCE and PRODUCTION Levels are used to restore the Deploy to the respective Test, Acceptance and Production ODI Repositories.

The result will look similar to the following overview:

Project Administration > Lifecycles Overview 

Lifecycle : BASE												
Description											Base	
	BASE Lifecycle											
Defined Levels												
		Name	Description	Type	Locked	Optional	Notification Type (Criteria)	Requester	Pre-Notify	Pre-Approve	Post-Approve	Post-Notify (Criteria)
		CREATE_ARCHIVE	This Level will create a Release based on the current Subversion trunk's content and will store the result in the IKAN ALM Archive	Build			No notification (Never)					
			DEPLOY_11_TEST	Deploy to WORK_11_TEST	Test		No notification (Never)					
			DEPLOY_11_UAT	Deploy to WORK_11_UAT	Test		No notification (Never)					
		DEPLOY_11_PROD	Deploy to WORK_11_PROD	Production			No notification (Never)					
Create Test Level Create Production Level												

Next, we will define the required Environments.

Creating the Build and Deploy Environments

A Level is a conceptual step in the Lifecycle. As we need a concrete Machine to execute our Build and Deploys on, we need to link a Build and Deploy Environment(s) to the Level.

The sources we want to Build will be transferred from the Version Control Repository to a subdirectory of the Source Location. The results will be placed in the Target Location. Note that, normally, those locations will be cleaned up after the Build, unless you activate the *Debug* option, which we do for the same reason as explained in the section about setting up the Build and Deploy Levels.

The following steps need to be repeated for every Level and for both Environments.

Note: For this *LCM4ODI_trial*, we **only** use the machine on which you installed IKAN ALM!

1. On the submenu, select *Build Environments > Create* and fill out the fields as required.

The screenshot shows a web-based form titled "Create Build Environment". The form is divided into two columns. The left column contains text input fields for "Name" (filled with "build"), "Build Suffix", "Source Location" (filled with ".env\MyProject\CREATE_ARCHIVE\create_archive\source"), "Target Location" (filled with ".env\MyProject\CREATE_ARCHIVE\create_archive\target"), and "Build Script". The right column contains radio buttons for "Downloadable Build" (set to "No") and "Debug" (set to "No"), a dropdown menu for "Level" (set to "CREATE_ARCHIVE"), another dropdown menu for "Machine" (set to "ikandemo004v"), and a dropdown menu for "Build Tool" (set to "ANT1.9.3"). At the bottom of the form are two buttons: "Create" and "Reset".

Note: The source and target location are preferably located in the ALM_DATA folder. You are free to specify the Source and Targets. We recommend to store them in a separate ALM_DATA folder to prevent those folders from being deleted during an eventual deinstallation. The structure entered in the Source and Target Location fields will be created automatically.

In our example we use `D:\ALM_DATA\env\MyProject\BUILD\build\source`.

`D:\ALM_DATA` is the ALM_DATA directory, followed by the `env` directory containing our project, `MyProject` (the project directory), `BUILD` (the level), `build` (the environment) and finally the `source` or `target` directory.

In order to distinguish Levels from Environments, we use uppercase for the level and lowercase for the environment directories.

- Repeat the same process for the Deploy Environment selecting *Deploy Environments > Create* on the submenu.

As the Source and Target locations differ for the three Levels, you need to create three environments.

The result will look similar to the following overview:

Deploy Environments Overview													
		Name	Machine	Source Location	Target Location	Deploy Script	Partial Deploy	Level	Deploy Tool	Build Environment	Debug		
				deploy_11_prod	ikandemo004	D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod/target			DEPLOY_11_PROD	ANT1.9.3	create_archive	
				deploy_11_test	ikandemo004	D:/ALM_DATA/env/MyProject/DEPLOY_11_TEST/deploy_11_test/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_TEST/deploy_11_test/target			DEPLOY_11_TEST	ANT1.9.3	create_archive	
				deploy_11_uat	ikandemo004	D:/ALM_DATA/env/MyProject/DEPLOY_11_UAT/deploy_11_uat/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_UAT/deploy_11_uat/target			DEPLOY_11_UAT	ANT1.9.3	create_archive	

Using Phases to define the Build and Deploy processes

Next, we need to define our Build (Create Archives) and Deploy (Restore) processes. The Build and Deploy processes are unique to every Solution you want to implement.

The behavior of the Build and Deploy processes is defined in the IKAN ALM Phases.

14.1. Core IKAN ALM Phases

Below you will find an overview of the default Phases generated by IKAN ALM at the moment you create your Levels and Build and Deploy environments.

To display them, select the [Edit Phases](#) link in front of the required Level, Build or Deploy Environment on the respective Overviews (In the *Project Administration* section, select *Levels > Overview*, *Build Environments > Overview* or *Deploy Environments > Overview*).

Default Level Phases

Project Administration > Level Phases Overview [?](#)


Level Info	
Name CONTBUILD	Notification Type No notification
Description Continuous Build Level	Notification Criteria Never
Type Build	Requester User Group ALM User
Locked Yes	Debug No

Phases Overview					
		Phase Name	Phase Version	Fail on Error	Next Phase on Error
		Retrieve Code	5.7.0	Yes	Cleanup Work Copy
		Build	5.7.0	Yes	Cleanup Work Copy
		Tag Code	5.7.0	Yes	Cleanup Work Copy
		Deploy	5.7.0	Yes	Cleanup Work Copy
		Cleanup Work Copy	5.7.0	Yes	

[Insert Phase](#) [History](#)

[Back](#)









































Default Build Environment Phases

Project Administration > Build Environment Phases Overview 

Build Environment


Name create_archive	Downloadable Build No
Build Suffix	Debug No
Source Location D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/source	Level CREATE_ARCHIVE
Target Location D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/target	Machine ikandemo004v
Build Script	Build Tool ANT1.9.3

Phases Overview

					Phase Name	Phase Version	Fail On Error	Next Phase On Error
					Transport Source	5.7.0	Yes	Cleanup Source
					Verify Build Script	5.7.0	Yes	Cleanup Source
					Execute Script	5.7.0	Yes	Cleanup Source
					Transport Deploy Script	5.7.0	Yes	Cleanup Source
					Compress Build	5.7.0	Yes	Cleanup Source
					Archive Result	5.7.0	Yes	Cleanup Source
					Cleanup Source	5.7.0	No	Cleanup Result
					Cleanup Result	5.7.0	No	

[Insert Phase](#) [History](#)


























Default Deploy Environment Phases

Project Administration > Deploy Environment Phases Overview 

Deploy Environment

Name deploy_11_prod	Debug Yes
Source Location D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod/source	Level DEPLOY_11_PROD
Target Location D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod//target	Machine ikandemo004v
Deploy Script	Deploy Tool ANT1.9.3
Partial Deploy No	Build Environment create_archive

Phases Overview

					Phase Name	Phase Version	Fail On Error	Next Phase On Error
					Transport Build Result	5.7.0	Yes	Cleanup Build Result
					Decompress Build Result	5.7.0	Yes	Cleanup Build Result
					Verify Deploy Script	5.7.0	Yes	Cleanup Build Result
					Execute Script	5.7.0	Yes	Cleanup Build Result
					Cleanup Build Result	5.7.0	No	

[Insert Phase](#) [History](#)

14.2. Specific Solution Phases

IKAN ALM provides specific Solution Phases for ODI, OWB and Mainframe.

In the next section, we will guide you through the installation of the Solution Phases for ODI, explaining how to import and configure them, and how to adjust the parameters.

ODI Solution Phases

The IKAN ALM ODI Solution uses two Solution Phases.

Solution Phase	Usage
Copy ODI Files	This phase is used to collect the objects from Subversion, which will be part of your ODI Release.
Restore Oracle DI Objects	This phase restores/imports Oracle DI Objects to a specified Work Repository.

15.1. Copying the ODI Solution Phases

Before you can install and use the Phases in IKAN ALM, you need to copy the *SolutionPhases_ODI_xx* folder to the right location.

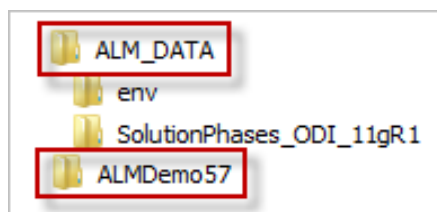
1. Create a new folder, called *ALM_DATA*, at the same level as your *IKANALM_HOME* folder.

Note: When installing the IKAN ALM Demo version, by default, the *IKANALM_HOME* directory is set to *ALMDemo*.

We recommend creating a separate folder for your ALM data at the same level as the *IKANALM_HOME* directory. This will prevent those data from being deleted during an eventual deinstallation. This *ALM_DATA* folder will be used to store the build and deploy information (in the *env* folder) and all data related to the Phases.

2. In the *ALM_DATA* folder, create the subfolder *env*.
This folder will contain all Build and Deploy information.
3. Copy the *SolutionPhases_ODI_xx* subfolder from your temporary location to the *ALM_DATA* folder.

Your directory structure should now look as follows:



15.2. Implementing the Phases in IKAN ALM

The procedure for implementing the Phases consist of the following steps:

1. [Importing the ODI Solution Phases in IKAN ALM](#) (page 48)
2. [Specifying the Initialization Values for the Parameters \(Global Administration\)](#) (page 51)
3. [Adding the “Copy ODI Files” Phase to the Build Environment](#) (page 54)
4. [Adding the “Restore Oracle DI Objects” Phase to the Deploy Environments](#) (page 56)
5. [Modifying the Phase Parameters in the Deploy Environments](#) (page 56)

Importing the ODI Solution Phases in IKAN ALM

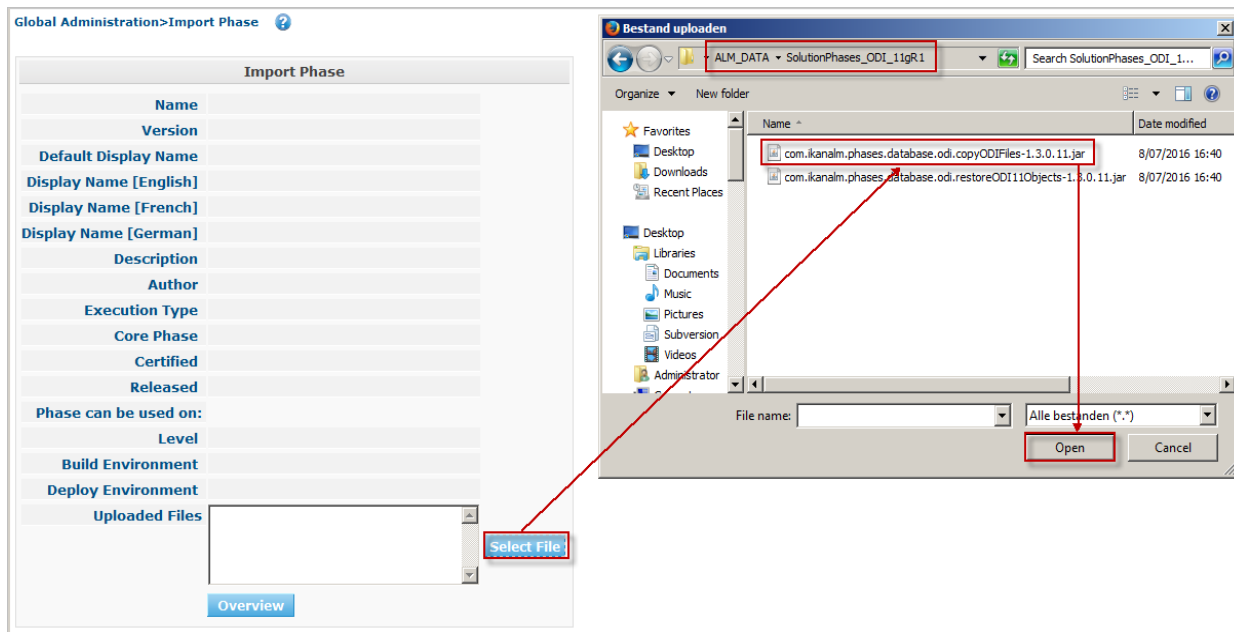
First of all, you need to import the Solution Phases in IKAN ALM.

In this example, we will import the following Phases (X.Y.Z being the release ID of the Phase):

Solution Phase	Corresponding JAR File
Copy ODI Files	com.ikanalm.phases.database.odi.copyODIFiles-X.Y.Z
Restore Oracle DI Objects	Import the Solution Phases corresponding to the version of ODI you are using: For ODI 11gR1: com.ikanalm.phases.database.odi.restoreODI11Objects-X.Y.Z For ODI 12cR1: com.ikanalm.phases.database.odi.restoreODI12R1Objects-X.Y.Z For ODI 12cR2: com.ikanalm.phases.database.odi.restoreODI12R2Objects-X.Y.Z

1. Go to *Global Administration > Phases > Import*.
2. On the Import Phase panel, click the *Select File* button.
Select the JAR file associated with the *Copy ODI Files* phase and click *Open*.

Note: In our example, the ODI Solution Phases are located in ALM_DATA/
SolutionPhases_ODI_XX.



3. On the *Import Phase* panel, click the *Import* button.

As a result, the *Import Phase* screen will display the information concerning the Phase, the files it contains (*PHASE_NAME.xml* being the main Ant script) and the Phase Parameters.

Global Administration > Import Phase

INFO: ACTION SUCCESSFUL

Import Phase

Name com.ikanalm.phases.database.odi.copyODIFiles

Version 1.3.0.11

Default Display Name Copy ODI Files

Display Name [English] Copy ODI Files

Display Name [French] Copie des fichiers ODI

Display Name [German] Copy ODI Files

Description Copy ODI-related files from a source dir to a target dir. By default copies all .xml, .shadow, and .sql files. Can be customized to include or exclude other file patterns.

Author IKAN

Execution Type ANT

Core Phase No

Certified No

Released No

Phase can be used on:

Level No

Build Environment Yes

Deploy Environment Yes

Uploaded Files

- copyOdiFiles.xml
- antext/lib/ant-contrib-1.0b3.jar
- antext/lib/be.ikan.scm4all.dient.ant.se
- antext/lib/com.springsource.org.apach
- antext/lib/com.springsource.org.apach

[Select File](#)

[Import](#) [Overview](#)

Import Phase Parameters

Name	Default value	Description	Mandatory	Secure	Integration Type
alm.phase.mainScript	copyOdiFiles.xml	Main Script	✓		None
alm.phase.extractBundle	true		✓		None
alm.phase.builder			✓		ANT
customExcludes		Custom patterns of files to exclude in the copy. Patterns should follow the syntax of Ant Patterns. Multiple patterns can be separated by ",".			None
customIncludes	**/*.*	Custom patterns of files to include in the copy. Patterns should follow the syntax of Ant Patterns. Multiple patterns can be separated by ",".			None
includeEmptyDirs	true	Flag to indicate whether to copy empty directories to the target dir. Valid values are : true, false.	✓		None

4. Repeat this procedure for the *Restore Oracle DI Objects* phase.

The *Phases Overview* displays the imported Phases.

Phases Overview

Name	Version	Display Name	Author	Execution Type	Core	Certified	Released	L	B	D
com.ikanalm.phases.database.odi.restoreODI11Objects	1.3.0.11	Restore Oracle DI 11 Objects	IKAN	ANT						✓
com.ikanalm.phases.database.odi.copyODIFiles	1.3.0.11	Copy ODI Files	IKAN	ANT				✓		✓

2 items found, displaying all

5. Next, you need to adapt the default parameter values.

Note: Parameters can be defined on Phase, Environment or Machine level. The Phase parameter takes precedence over the Environment Parameter which takes precedence over the Machine parameter.

There is no general rule to determine on which level you want to define the parameters. It depends on your company standards, on how unique the parameter is (whether it is only used in one single phase or in multiple phases), whether the value of the parameter is the same for all Levels and Environments on a Machine (in which case it might be interesting to define the parameter on Machine level) or whether it is different for each Environment (in which case it would probably be best to define it on Environment level), etc.

For a more detailed explanation on the definition of parameters, please refer to the chapter *Phases* in the *IKAN ALM User Guide* or contact IKAN ALM Support if you need advice on a particular situation.

As this *Getting Started Guide* is using a very basic example, we will keep the definition of the parameters as simple as possible. Only for a few parameters we will need to specify a different value for each of the Environments.

Specifying the Initialization Values for the Parameters (Global Administration)

For the ODI Solution to work for you, you need to verify and, if required, modify the initialization values of the Phase Parameters for each Phase. This is done in the Global Administration section of IKAN ALM.

The following table lists the parameters of the Deploy (Restore) Phase which you will need to modify in the Global Administration section.

Parameter	Phase	Description	Mandatory/Optional
odi.home	Restore Oracle DI Objects	ODI_HOME Location	Mandatory
odi.user.name	Restore Oracle DI Objects	ODI User Name	Mandatory
odi.user.passwd	Restore Oracle DI Objects	ODI User Password	Mandatory
oracle.home	Restore Oracle DI Objects	ORACLE_HOME Location. Required when rdbms.master.tns.connect.descriptor has been set	Mandatory
rdbms.master.passwd	Restore Oracle DI Objects	Password of the ODI Master Repository Database Connection	Mandatory
rdbms.master.sid	Restore Oracle DI Objects	SID of the ODI Master Repository Database Connection	Mandatory
rdbms.master.username	Restore Oracle DI Objects	User Name of the ODI Master Repository Database Connection	Mandatory

As you can see, the parameters are set to mandatory by default. In this context, *mandatory* means that the parameters will be automatically created when adding the Phase to a Level or Build/Deploy Environment.

1. Go to *Global Administration > Phases > Overview*.
2. On the *Phases Overview* panel, click the  *Edit* icon in front of the *Restore Oracle DI Objects* phase you imported.

Global Administration > Phases Overview ?

Search Phase

Name: Show Core Phases: Yes No All

Version: 1.3.0.11 Show Certified Phases: Yes No All

Display Name: Show Released Phases: Yes No All









Description: Show if Phase can be used on:

Author: Level: Yes No All

Execution Type: Build Environment: Yes No All

Deploy Environment: Yes No All

Phases Overview

Name	Version	Display Name	Author	Execution Type	Core	Certified	Released	L	B	D
    com.ikanalm.phases.database.odi.restoreODI11Objects	1.3.0.11	Restore Oracle DI 11 Objects	IKAN	ANT						<input checked="" type="checkbox"/>
    com.ikanalm.phases.database.odi.copyODIFiles	1.3.0.11	Copy ODI Files	IKAN	ANT						<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>

2 items found, displaying all

The *Edit Phase* screen will be displayed. Underneath the *Phase Info* panel, you will find the list of all defined Phase Parameters.

Note: You can sort this list alphabetically by clicking the *Name* column heading.

Global Administration > Edit Phase ?

Phase Info

Name: com.ikanalm.phases.database.odi.restoreODI11Objects Execution Type: ANT

Version: 1.3.0.11 Core Phase: No

Default Display Name: Restore Oracle DI 11 Objects Certified: No
















Author: IKAN Released: No

Description: Restore Oracle DI 11 Objects to a Work Repository. Compatible with ODI version 11.

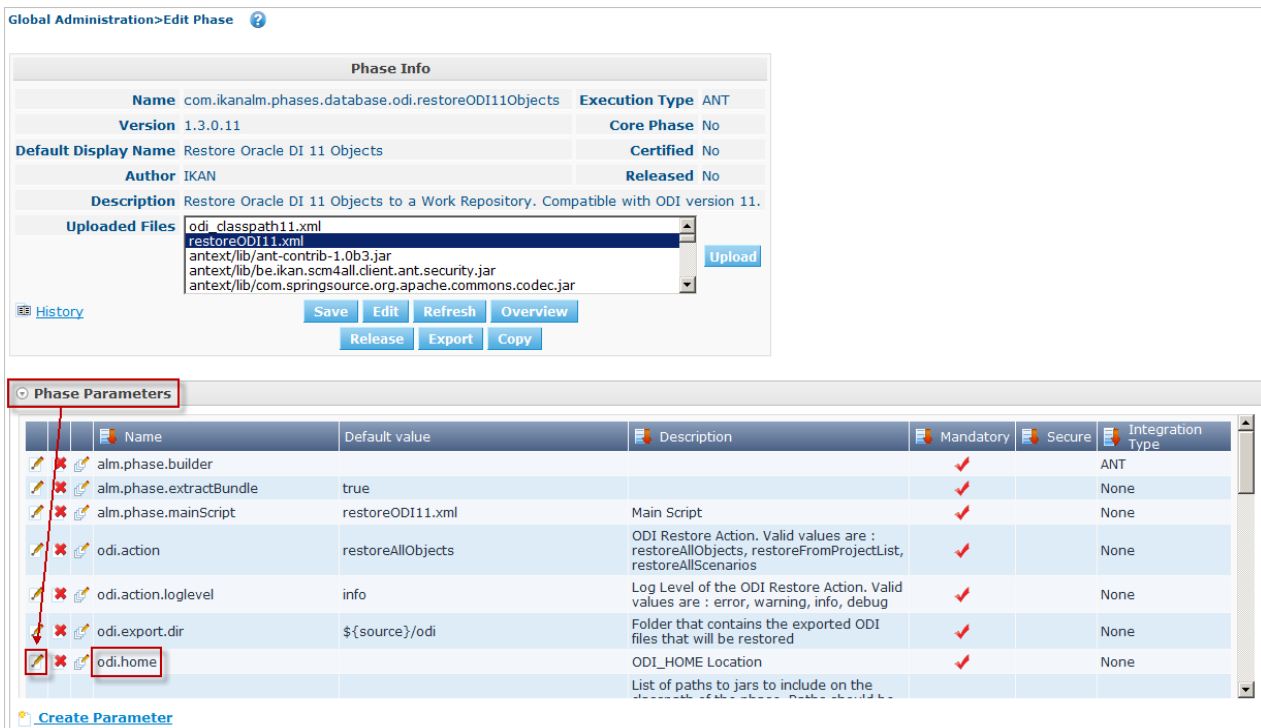
Uploaded Files:

[History](#)

Phase Parameters

Name	Default value	Description	Mandatory	Secure	Integration Type
  alm.phase.builder			<input checked="" type="checkbox"/>		ANT
  alm.phase.extractBundle	true		<input checked="" type="checkbox"/>		None
  alm.phase.mainScript	restoreODI11.xml	Main Script	<input checked="" type="checkbox"/>		None
  odi.action	restoreAllObjects	ODI Restore Action. Valid values are : restoreAllObjects, restoreFromProjectList, restoreAllScenarios	<input checked="" type="checkbox"/>		None
  odi.action.loglevel	info	Log Level of the ODI Restore Action. Valid values are : error, warning, info, debug	<input checked="" type="checkbox"/>		None
  odi.export.dir	\${source}/odi	Folder that contains the exported ODI files that will be restored	<input checked="" type="checkbox"/>		None
  odi.home		ODI_HOME Location	<input checked="" type="checkbox"/>		None
 		List of paths to jars to include on the			

- Click the  *Edit* icon in front of the *odi.home* parameter.



The screenshot shows the 'Global Administration > Edit Phase' window. The 'Phase Info' section displays the following details:

- Name:** com.ikanalm.phases.database.odi.restoreODI11Objects
- Version:** 1.3.0.11
- Default Display Name:** Restore Oracle DI 11 Objects
- Author:** IKAN
- Description:** Restore Oracle DI 11 Objects to a Work Repository. Compatible with ODI version 11.
- Uploaded Files:**
 - odi_classpath11.xml
 - restoreODI11.xml
 - antext/lib/ant-contrib-1.0b3.jar
 - antext/lib/be.ikan.scm4all.client.ant.security.jar
 - antext/lib/com.springsource.org.apache.commons.codec.jar

The 'Phase Parameters' table is shown below:

Name	Default value	Description	Mandatory	Secure	Integration Type
alm.phase.builder			✓		ANT
alm.phase.extractBundle	true		✓		None
alm.phase.mainScript	restoreODI11.xml	Main Script	✓		None
odi.action	restoreAllObjects	ODI Restore Action. Valid values are : restoreAllObjects, restoreFromProjectList, restoreAllScenarios	✓		None
odi.action.loglevel	info	Log Level of the ODI Restore Action. Valid values are : error, warning, info, debug	✓		None
odi.export.dir	\${source}/odi	Folder that contains the exported ODI files that will be restored	✓		None
odi.home		ODI_HOME Location	✓		None


The *Edit Phase Parameter* pop-up window is displayed.


- Specify the value in the *Default Value* field and click *Save*.
- Repeat this procedure for each of the global mandatory parameters that need to be adapted.

Parameter	Description
oracle.home	Oracle Home Directory Note: this is only required if you are using a TNS connection, in which case you should also set the <code>rdbms.master.tns.connect.descriptor</code> parameter.
odi.user.name	ODI User Name
odi.user.passwd	ODI User Password
rdbms.master.passwd	Password of the ODI Master Repository Database Connection
rdbms.master.sid	SID of the ODI Master Repository Database Connection
rdbms.master.username	User Name of the ODI Master Repository Database Connection
restore.odi.dependencies	It is recommended to set the default value of this parameter to <i>true</i> .

Note: As mentioned before, if you do not use the Oracle defaults for `rdbms.master.host` and `rdbms.master.port` (see [Other Oracle Initialization Values](#) (page 9)), you will have to modify those parameters as well (set them to mandatory and specify your value).

Adding the “Copy ODI Files” Phase to the Build Environment

1. Go to *Project Administration > Projects Overview* and click the  *Edit* icon in front of your Project.

Project Administration > Projects Overview 

Search Project
















Base Criteria [Extended Criteria](#)

Name

VCR Project Name


Show Hidden Projects Yes No All

Projects Overview

		Name	Description	VCR	VCR Project Name	Project Type	Build Tool Type	Deploy Tool Type	Build Script	Deploy Script	Locked	User Access	Admin Access
			Customers	A simple web application for managing a customers ...	SVNRepository	customers	Release-based	ANT				ALM User	ALM Project
			Demo	Demo Project for the Getting Started Guide.	vcr4odi11svn	vcr4odi11svn	Release-based	ANT	build.xml	deploy.xml			
			Monocalendar	Demo project for ALM stored in Subversion	SVNRepository	monocalendar	Release-based	NANT	default.build			ALM User	ALM Project
			MyProject	Demo Project for the LCM4ODI Solution	SVNRepository	MyProject	Release-based	ANT	build.xml	deploy.xml			

2. On the submenu, select *Build Environments > Overview* and click the  *Edit Phases* icon in front of your Build Environment.

























The *Build Environment Phases Overview* screen is displayed. It lists the default IKAN ALM Phases. Some of those phases need to be removed and replaced with the specific ODI Solution Phases.

Project Administration > Build Environment Phases Overview 

Build Environment

Name create_archive	Downloadable Build No
Build Suffix	Debug No
Source Location D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/source	Level CREATE_ARCHIVE
Target Location D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/target	Machine ikandemo004v
Build Script	Build Tool ANT1.9.3

Phases Overview

			Phase Name	Phase Version	Fail On Error	Next Phase On Error
			Transport Source	5.7.0	Yes	Cleanup Source
			Verify Build Script	5.7.0	Yes	Cleanup Source
			Execute Script	5.7.0	Yes	Cleanup Source
			Transport Deploy Script	5.7.0	Yes	Cleanup Source
			Compress Build	5.7.0	Yes	Cleanup Source
			Archive Result	5.7.0	Yes	Cleanup Source
			Cleanup Source	5.7.0	No	Cleanup Result
			Cleanup Result	5.7.0	No	

[Insert Phase](#) [History](#)

3. Delete the *Verify Build Script*, the *Execute Script* and the *Transport Deploy Script* using the  *Delete* icon.

Next, we will insert the appropriate ODI Solution Phases.

4. Click the *Insert Phase* link underneath the *Phases Overview*.
The *Insert Phase* screen is displayed.

Project Administration > Insert Phase ?

Build Environment

Name create_archive	Downloadable Build No
Build Suffix	Debug No
Source Location D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/source	Level CREATE_ARCHIVE
Target Location D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/target	Machine ikandemo004v
Build Script	Build Tool ANT1.9.3

Phase to insert

Phase Copy ODI Files - 1.3.0.12

Fail on Error Yes No

Insert at position 2

Next Phase on Error 4. Cleanup Source - 5.

Label

Phases Overview

Position	Phase
1	Transport Source - 5.7.0
2	Compress Build - 5.7.0
3	Archive Result - 5.7.0
4	Cleanup Source - 5.7.0
5	Cleanup Result - 5.7.0

Available Phases

Phase Name	Phase Version	Execution Type	Author
<input checked="" type="radio"/> Copy ODI Files	1.3.0.12	ANT	IKAN
<input type="radio"/> Tomcat Deploy	1.0.0	ANT	IKAN
<input type="radio"/> Update Database	1.0.0	ANT	IKAN
<input type="radio"/> Transport Package Results	5.7.0	CORE	IKAN
<input type="radio"/> Execute Script	5.7.0	CORE	IKAN
<input type="radio"/> Cleanup Result	5.7.0	CORE	IKAN
<input type="radio"/> Cleanup Source	5.7.0	CORE	IKAN

4.1. On the *Available Phases* panel, select the *Copy ODI Files* phase.

4.2. Specify its position in the *Insert at position* field, by selecting 2 from the drop-down menu.
While going through the drop-down menu, the position will be indicated with a dotted line on the *Phases Overview* panel at the left.

4.3. Select the *Next Phase on Error*, which will be the *Cleanup Source* phase.

4.4. Finally, click *Insert*.

The Phase will be added to the *Phases Overview*.

Project Administration > Build Environment Phases Overview ?

INFO: ACTION SUCCESSFUL


Build Environment	
Name create_archive	Downloadable Build No
Build Suffix	Debug No
Source Location D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/source	Level CREATE_ARCHIVE
Target Location D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/target	Machine ikandemo004v
Build Script	Build Tool ANT1.9.3

Phases Overview							
				Phase Name	Phase Version	Fail On Error	Next Phase On Error
↓	↕	✖	✖	Transport Source	5.7.0	Yes	Cleanup Source
↑	↓	↕	✖	Copy ODI Files	1.3.0.12	Yes	Cleanup Source
↑	↓	↕	✖	Compress Build	5.7.0	Yes	Cleanup Source
↑	↓	↕	✖	Archive Result	5.7.0	Yes	Cleanup Source
↑	↓	↕	✖	Cleanup Source	5.7.0	No	Cleanup Result
↑	↓	↕	✖	Cleanup Result	5.7.0	No	

[Insert Phase](#) [History](#)

- Before continuing, check the order of the Phases. If needed, you can modify the order using the *Up* and *Down* arrows.
- Next, we will use the same procedure to add the required phases to the Deploy Environments.

Adding the “Restore Oracle DI Objects” Phase to the Deploy Environments

- Select *Deploy Environments > Overview* and click the  *Edit Phases* icon in front of your Deploy Environment.
- Delete the *Verify Deploy Script* and the *Execute Script* phases.
- Insert the *Restore Oracle DI Objects* phase after the *Decompress Build Result* phase.
- Select the *Next Phase on Error*, which will be the *Cleanup Build Result* phase.
- Repeat the previous steps for each of your Deploy Environments (in our example: Test, Acceptance and Production).


Modifying the Phase Parameters in the Deploy Environments


At the beginning of this section, we have modified the default values for all mandatory parameters.

Before we can really use our Phases, we still need to make a few modifications to some parameters which have different values for the Deploy Environments. For example: the *odi.workRepository.name* parameter specifies the location of the ODI Work Repository, which is different for each Deploy environment.

These changes are made in the Project Administration section of IKAN ALM.

Environment	Parameter	Explanation	Mandatory/Optional
Deploy	odi.workRepository.name	ODI Work Repository	Mandatory
Deploy	odi.action	Repository actions. Use the following values: <ul style="list-style-type: none"> For the TEST environment: restoreAllObjects For the ACCEPTANCE and PRODUCTION environments: restoreAllScenarios 	Mandatory

1. Go to *Project Administration > Projects Overview* and click the  *Edit* icon in front of your Project.
2. On the submenu, select *Deploy Environments > Overview*.
The *Deploy Environments Overview* screen is displayed.

Project Administration > Deploy Environments Overview 

Search Deploy Environment

Base Criteria
Extended Criteria

Name















Build Environment

Level

Machine

Deploy Tool


Deploy Environments Overview

	Name	Machine	Source Location	Target Location	Deploy Script	Partial Deploy	Level	Deploy Tool	Build Environment	Debug
   	deploy_11_prod	ikandemo004v	D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod//target			DEPLOY_11_PROD	ANT1.9.3	create_archive	
   	deploy_11_test	ikandemo004v	D:/ALM_DATA/env/MyProject/DEPLOY_11_TEST/deploy_11_test/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_TEST/deploy_11_test/target			DEPLOY_11_TEST	ANT1.9.3	create_archive	
   	deploy_11_uat	ikandemo004v	D:/ALM_DATA/env/MyProject/DEPLOY_11_UAT/deploy_11_uat/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_UAT/deploy_11_uat/target			DEPLOY_11_UAT	ANT1.9.3	create_archive	

3. Modify the *odi.workRepository.name* parameter.
As the *odi.workRepository.name* parameter is Mandatory, the parameter has already been created in our Deploy Environments.
Therefore, we only need to specify the appropriate value for the selected Deploy Environment.

3.1. Click the  *Edit Phases* icon in front of your Deploy Environment.





















The *Deploy Environments Phases Overview* is displayed.

Project Administration > Deploy Environment Phases Overview 

Deploy Environment

Name deploy_11_prod	Debug Yes
Source Location D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod/source	Level DEPLOY_11_PROD
Target Location D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod//target	Machine ikandemo004v
Deploy Script	Deploy Tool ANT1.9.3
Partial Deploy No	Build Environment create_archive

Phases Overview

				Phase Name	Phase Version	Fail On Error	Next Phase On Error
				 Transport Build Result	5.7.0	Yes	Cleanup Build Result
				 Decompress Build Result	5.7.0	Yes	Cleanup Build Result
				 Restore Oracle DI 11 Objects	1.3.0.12	Yes	Cleanup Build Result
				 Cleanup Build Result	5.7.0	No	

































[Insert Phase](#) [History](#)

3.2. Click the  *View Parameters* icon in front of the *Restore Oracle DI Objects* phase.

3.3. Click the  *Edit* icon in front of the *odi.workRepository.name* parameter.

3.4. Enter the correct value and click *Save*.

Phase Parameters

		Name	Value	Integration Type	Mandatory	Secure
		alm.phase.builder		ANT	<input checked="" type="checkbox"/>	
		alm.phase.extractBundle	true	None	<input checked="" type="checkbox"/>	
		alm.phase.mainScript	restoreODI11.xml	None	<input checked="" type="checkbox"/>	
		odi.action	restoreAllObjects	None	<input checked="" type="checkbox"/>	
		odi.action.loglevel	info	None	<input checked="" type="checkbox"/>	
		odi.export.dir	\${source}/odi	None	<input checked="" type="checkbox"/>	
		odi.home		None	<input checked="" type="checkbox"/>	
		odi.home.classpath.includes		None	<input checked="" type="checkbox"/>	
		odi.home.jps-config		None	<input checked="" type="checkbox"/>	
		odi.user.name		None	<input checked="" type="checkbox"/>	
		odi.user.passwd		None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
		odi.workRepository.name		None	<input checked="" type="checkbox"/>	
		oracle.home		None		
		project.codes		None		
		rdbms.master.host	localhost	None	<input checked="" type="checkbox"/>	
		rdbms.master.passwd	*****	None	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Edit Parameter Value ✖

Name odi.workRepository.name

Value

3.5. Return to *Deploy Environments > Overview* and do the same for the other Deploy Environments.

4. The last parameter to modify is the *odi.action* parameter which specifies the repository action to be executed.

The default value (`restoreAllObjects`) is correct for the TEST environment, so we only need to modify the value for the ACCEPTANCE and PRODUCTION environment.

Follow the procedure described in the previous step and set the value to `restoreAllScenarios` for ACCEPTANCE and PRODUCTION.

5. All parameters have now been set.

Auditing the Project

Everything is ready to start a Build, except that the Level was locked when we created it: first we need to verify if our definitions are consistent. Do so by selecting *Audit Project* on the submenu.

On the overview, you will see most of the different objects we created.

The information screen for the *MyProject* Project displays the Build Archive of the Head Project Stream (where our future Builds will be stored) and the Build Level containing one Build Environment on the IKAN ALM Agent.

Click the *Unlock* button, and we are ready to build!

Project Administration > Audit Project

INFO: THE PROJECT CONFIGURATION IS CONSISTENT. CLICK THE UNLOCK BUTTON TO ENABLE THE PROJECT.

Project Info: MyProject

Name MyProject	Build Script build.xml	Build Tool Type ANT
Description Demo Project for the LCM4ODI Solution	Deploy Script deploy.xml	Deploy Tool Type ANT
Locked No	VCR SVNRepository	User Access
Hidden No	VCR Project Name MyProject	Admin Access

Unlock

Project Stream Build Archive Locations

Description	Locked	Head	Build Prefix	Build Suffix	Build Archive Location	Message
		✓	REL		D:/ALMDemo_57/system/buildArchive/MyProject/REL	

Build and Deploy Environment Overview

Environment Name	Environment Type	Level Name	Level Locked	Level Type	Machine	Source	Target	Message
create_archive	Build	CREATE_ARCHIVE	🔒	Build	ikandemo004v	D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/source	D:/ALM_DATA/env/MyProject/CREATE_ARCHIVE/create_archive/target	
deploy_11_prod	Deploy	DEPLOY_11_PROD	🔒	Production	ikandemo004v	D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_PROD/deploy_11_prod/target	
deploy_11_test	Deploy	DEPLOY_11_TEST	🔒	Test	ikandemo004v	D:/ALM_DATA/env/MyProject/DEPLOY_11_TEST/deploy_11_test/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_TEST/deploy_11_test/target	
deploy_11_uat	Deploy	DEPLOY_11_UAT	🔒	Test	ikandemo004v	D:/ALM_DATA/env/MyProject/DEPLOY_11_UAT/deploy_11_uat/source	D:/ALM_DATA/env/MyProject/DEPLOY_11_UAT/deploy_11_uat/target	

Committing a Change Using VCR4ODI

Before we execute a Build in IKAN ALM, we have to make sure that changes we made in ODI have been committed to the Subversion repository.

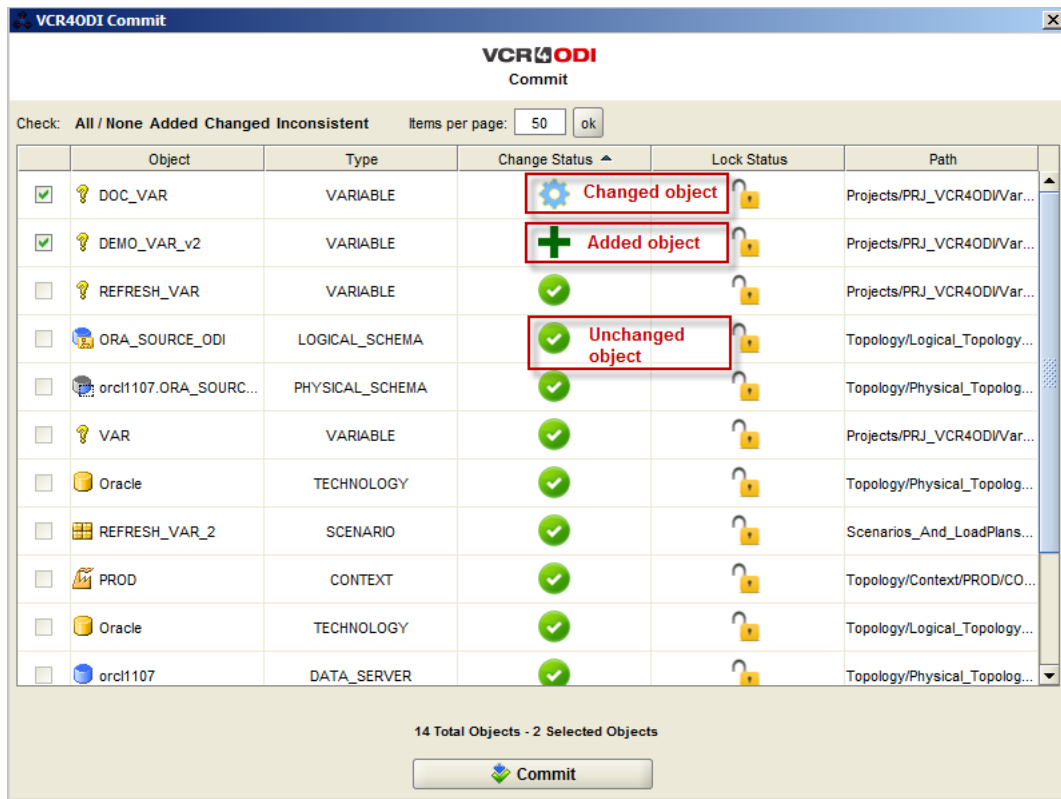
Here is where VCR4ODI comes into the picture.

The VCR4ODI connector enables the interaction between ODI Studio (the ODI Repositories) and the Version Control Repository (Subversion). It is used to commit ODI objects to the VCR and, vice versa, to restore objects from the VCR into ODI.







1. Start the *VCR4ODI* connector using your User Profile and select the *Oracle Data Integrator* tab to display the objects in the ODI Repository.
2. Select the object(s) you want to commit.
3. 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 selected objects and their dependencies.

The list of all the selected objects, and the dependencies that were calculated by the commit check-in operation, will be displayed in a new window. For each of the objects, the status and the locks are indicated.



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. Note: 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.

Status	Description
	The object is locked by the current user.
	The object is locked by another user. Note: If the object has been modified by that other user, you will not be able to commit.

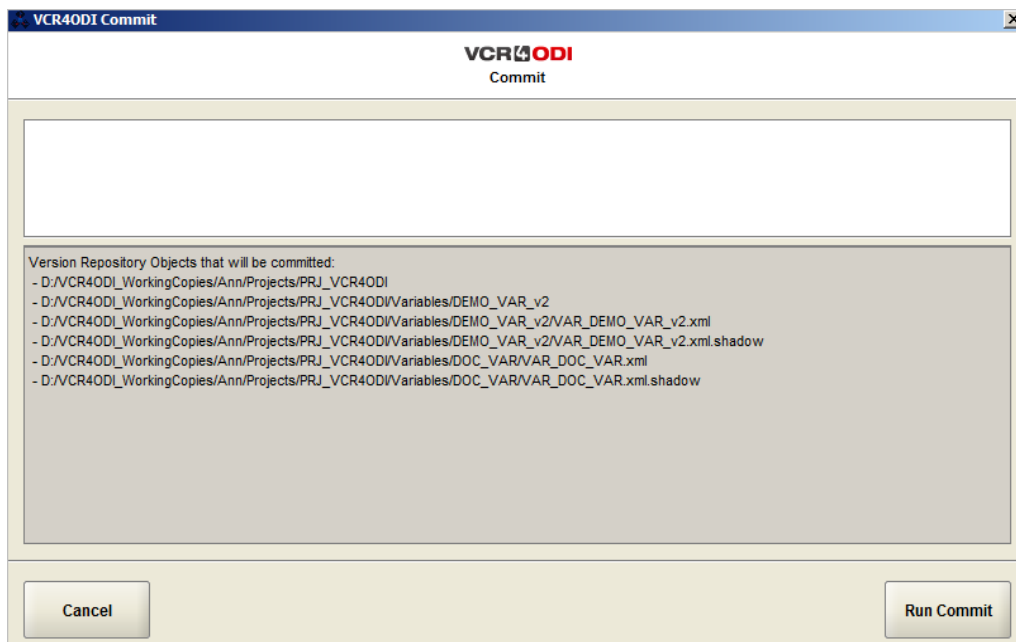
- If required, you can still refine the selection of objects that will be committed using the selection options above the displayed list.

Several additional selection options are available:

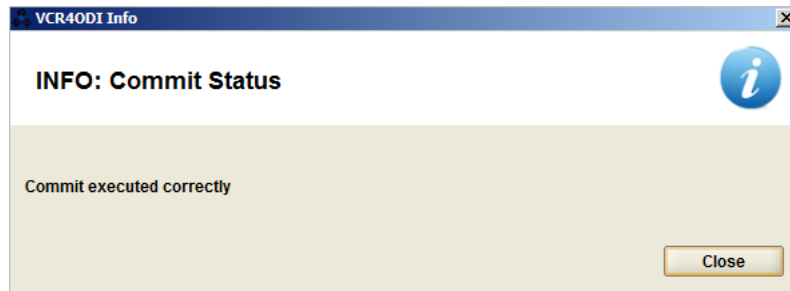
All/None	Clicking the <i>All/None</i> option will select/deselect all objects.
Added	Clicking the <i>Added</i> option will select/deselect the new objects.
Changed	Clicking the <i>Changed</i> option will select/deselect the modified objects.
Inconsistent	Clicking the <i>Inconsistent</i> option will select/deselect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

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

- Confirm the selection by clicking the *Commit* button.
- The list of all objects that will be committed is displayed. 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.



7. Click the *Run Commit* button to finalize the commit procedure.
You can also click *Cancel* to stop the commit operation and return to the *Oracle Data Integrator* tab.
8. If the commit procedure executes correctly, the following message will be displayed:



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

9. Your changes have been committed to the Repository and, next, IKAN ALM will be able to restore the committed ODI objects and use them for the Build and Deploy processes.

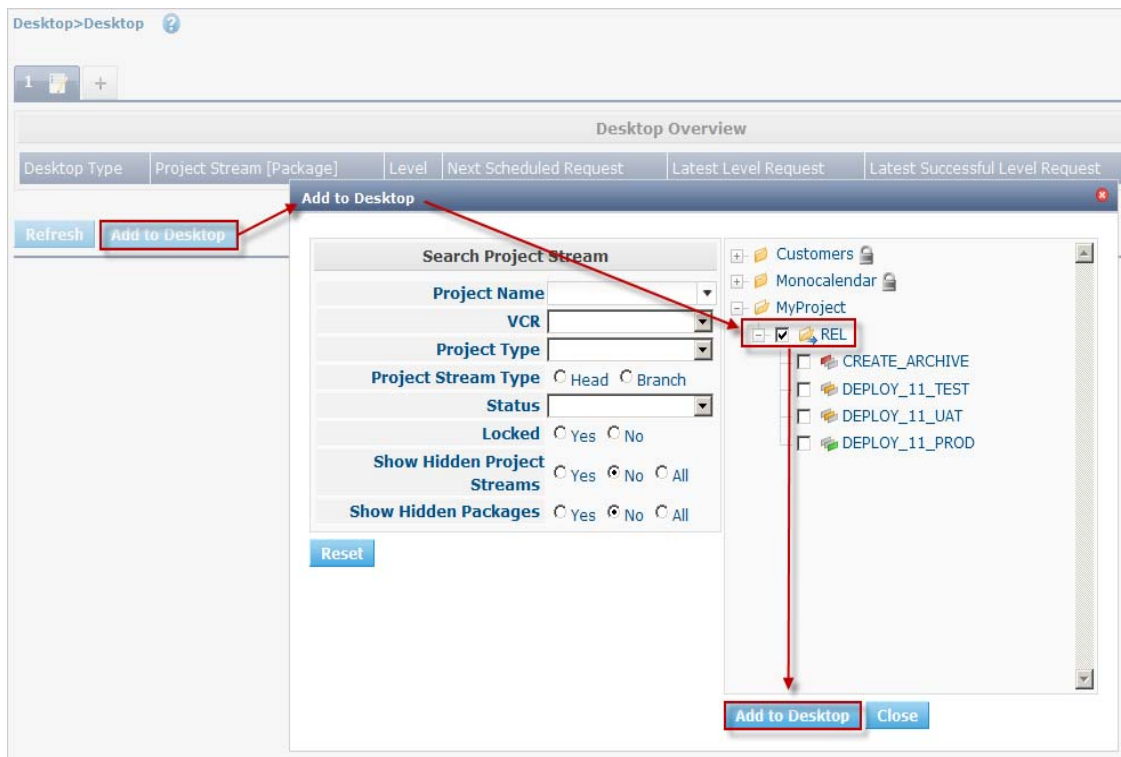
Creating the Build Level Request

What have we done so far?

- We have defined a Subversion repository in IKAN ALM
- Next, we have created an IKAN ALM Project and defined the necessary Lifecycles, Levels and Environments.
- After that, we have imported and configured our Solution Phases.
- Finally, we have committed our development changes to the Subversion repository.

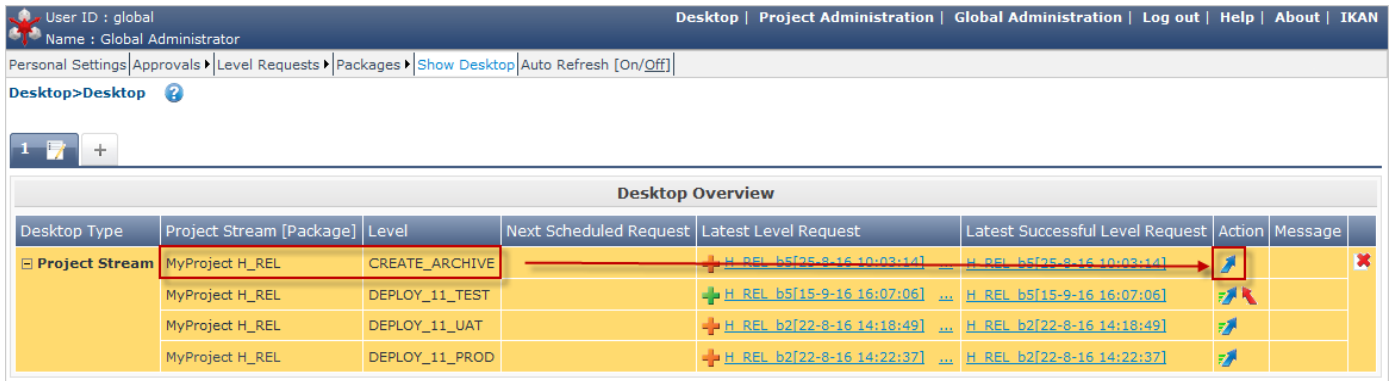
The next step will be to create a Build Level Request using the Sources in the Subversion repository.





1. First, we will add the Head Project Stream of the *MyProject* Project to our Desktop. Go to your Desktop and click the *Add to Desktop* button at the bottom of the screen.
2. In the pop-up window, find the Head Project Stream (REL) of the *MyProject* project, select it and click *Add to Desktop*.



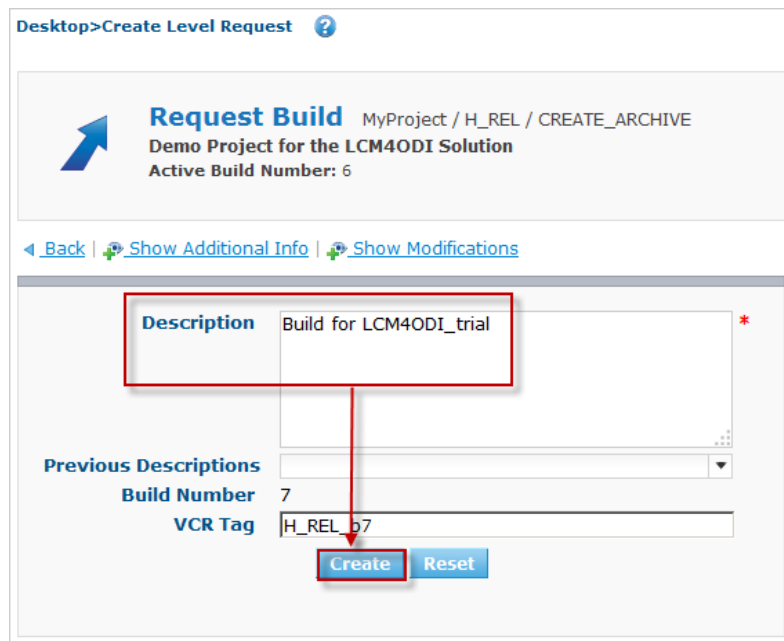
The Project stream will be added to your Desktop. Click the *Close* button to close the pop-up window.

3. Click the  *Request* icon in the *Action* column of our Project Stream.



Desktop Type	Project Stream [Package]	Level	Next Scheduled Request	Latest Level Request	Latest Successful Level Request	Action	Message
Project Stream	MyProject_H_REL	CREATE_ARCHIVE		H_REL_b5[25-8-16 10:03:14]	H_REL_b5[25-8-16 10:03:14]		
	MyProject_H_REL	DEPLOY_11_TEST		H_REL_b5[15-9-16 16:07:06]	H_REL_b5[15-9-16 16:07:06]		
	MyProject_H_REL	DEPLOY_11_UAT		H_REL_b2[22-8-16 14:18:49]	H_REL_b2[22-8-16 14:18:49]		
	MyProject_H_REL	DEPLOY_11_PROD		H_REL_b2[22-8-16 14:22:37]	H_REL_b2[22-8-16 14:22:37]		

The *Create Level Request* screen will be displayed.



Request Build MyProject / H_REL / CREATE_ARCHIVE
 Demo Project for the LCM4ODI Solution
 Active Build Number: 6

[Back](#) | [Show Additional Info](#) | [Show Modifications](#)

Description Build for LCM4ODI_trial *

Previous Descriptions

Build Number 7

VCR Tag H_REL_b7

[Create](#) [Reset](#)

Provide a meaningful description, do not modify the Indicative VCR Tag which will be created in Subversion when the Build is successful.

4. Click the *Create* button.

You are forwarded to the *Level Requests Overview* screen.

Desktop>Level Requests Overview ?

Search Level Request

Project

Name

Type

Issue Tracking System

Package

Project Stream

Level

Level Request

User ID User Group

Status Type

VCR Tag Action Type

Level Request Dates

Level Requests Overview

OID	Project Stream	Level Name	Level Type	Action Type	User ID	Status	Build Number	VCR Tag	Start	Duration
22	MyProject H_REL	CREATE_ARCHIVE	Build		global		7	H_REL_b7		
21	MyProject H_REL	CREATE_ARCHIVE	Build		global		6	H_REL_b6	16-9-16 10:52:25	00:00:14
20	MyProject H_REL	DEPLOY_11_TEST	Test		global		5	H_REL_b5	15-9-16 16:05:24	00:01:42
19	MyProject H_REL	DEPLOY_11_TEST	Test		global		5	H_REL_b5	15-9-16 16:02:32	00:01:12
18	MyProject H_REL	DEPLOY_11_TEST	Test		global		5	H_REL_b5	15-9-16 15:58:45	00:01:10
17	MyProject H_REL	DEPLOY_11_TEST	Test		global		5	H_REL_b5	15-9-16 15:56:09	00:01:22
16	MyProject H_REL	DEPLOY_11_TEST	Test		global		5	H_REL_b5	25-8-16 10:06:23	00:01:15
15	MyProject H_REL	CREATE_ARCHIVE	Build		global		5	H_REL_b5	25-8-16 10:03:05	00:00:09
14	MyProject H_REL	CREATE_ARCHIVE	Build		global		4	H_REL_b4	25-8-16 9:51:00	00:00:18
13	MyProject H_REL	CREATE_ARCHIVE	Build		global		3	H_REL_b3	22-8-16 14:41:20	00:00:12

22 items found, displaying 1 to 10

5. Click the *OID* number in front of the Project Stream (22 in this example).

You are forwarded to the *Level Requests Detail* screen.

The header of the *Level Request Detail* screen displays information on the status of the Level Request. The different tab pages underneath the header display additional information such as the status of each of the Phases (on the *Phase Logs* tab).

This is how it should look when the Request is finished:

Desktop>Level Request Detail ?

Warning MyProject / H_REL / CREATE_ARCHIVE / Build# 7
 22: Build for LCM4ODI_trial
 Requested by: global on: 16-9-16 11:01:35

Use the tab pages to display additional information.

Summary Phase Logs Results Approvals Issues Sources Modifications Dependencies

Back Refresh Build History

Actions
No actions available

Info
Build Number 7
VCR Tag H_REL_b7
Action Request Build
Type Builds based on latest code
Start 16-9-16 11:01:35
Duration 00:00:10
[Show more...](#)

Builds & Deploys

	OID	Environment	Machine	Start	D
+	7	create_archive	ikandemo004v	16-9-16 11:01:41	0

Warning log

Phase Cleanup Work Copy - 5.7.0
Start 16-9-16 11:01:46
Duration < 1 sec.
Status Warning

Message
 WorkCopy Location D:\ALMDemo_57\system/workCopy/22 not cleaned up, because of enabled debugging on the level
[Top](#)

Note: The Deploy ended in warning. That is due to the fact that the *Debug* option is still set for the Level.

6. Select the *Results* tab page.

This page shows the result of the Build executed by the IKAN ALM Agent on the Build Environment.

Desktop>Level Request Detail ?

Warning MyProject / H_REL / CREATE_ARCHIVE / Build# 7
 22: Build for LCM4ODI_trial
 Requested by: global on: 16-9-16 11:01:35

Summary Phase Logs **Results** Approvals Issues Sources Modifications Dependencies

Back Refresh Build History

Results

Build: 7

Build File Name MyProject_H_REL_b7_CREATE_ARCHIVE.zip
File Size 394 KB
Archive Status Present


Environment create_archive
Machine ikandemo004v
Status +

[Download Build Result](#)

MyProject_H_REL_b7_CREATE_ARCHIVE.zip

7. Click the *Download Build Result* link to download and check the Build Result.
8. Select the *Phase Logs* tab page. The gray lines on the overview represent the different Phases, the white lines represent the Build or Deploy actions.

Desktop > Level Request Detail ?

 **Warning** [MyProject / H_REL / CREATE_ARCHIVE / Build# 7](#)
 22: Build for LCM4ODI_trial
 Requested by: global on: 16-9-16 11:01:35

Summary **Phase Logs** Results Approvals Issues Sources Modifications Dependencies

[_Back](#) | [Refresh](#) | [Build History](#)

Phase Logs

> [Level Parameters](#)

Phase Name	Start Date/Time	Duration
> + Retrieve Code	16-9-16 11:01:35	00:00:05
> + Build	16-9-16 11:01:41	00:00:04
> + Build 7 on machine ikandemo004v	16-9-16 11:01:41	00:00:04
> + Tag Code	16-9-16 11:01:46	< 1 sec.
> + Deploy	16-9-16 11:01:46	< 1 sec.
> + Cleanup Work Copy	16-9-16 11:01:46	< 1 sec.

9. Click the Build name (in our example: *Build 7 on machine ikandemo004v*) to expand the information panel displaying the Phase details.

 **Build 7 on machine ikandemo004v** 16-9-16 11:01:41 00:00:04

OID 7 Start Date/Time 16-9-16 11:01:41
 Environment create_archive Duration 00:00:04
 Machine ikandemo004v Status **Success**

> [Build Parameters](#)

> + Transport Source	16-9-16 11:01:42	< 1 sec.
> + Copy ODI Files	16-9-16 11:01:42	00:00:03
> + Compress Build	16-9-16 11:01:45	< 1 sec.
> + Archive Result	16-9-16 11:01:45	< 1 sec.
> + Cleanup Source	16-9-16 11:01:45	< 1 sec.
> + Cleanup Result	16-9-16 11:01:45	< 1 sec.

10. Next, you can click one of the Phases to immediately jump to its *Phase Log*.

The *Copy ODI Files* Phase Log shows the result of the *copyODIFiles.xml* script executed by the Ant scripting tool.

The screenshot displays the following information:

- Build Header:** Build 7 on machine ikandemo004v, Start Date/Time 16-9-16 11:01:41, Duration 00:00:04, Status Success.
- Environment:** create_archive, Machine ikandemo004v.
- Build Parameters:** Transport Source (16-9-16 11:01:42, < 1 sec.), Copy ODI Files (16-9-16 11:01:42, 00:00:03).
- Phase Details (Copy ODI Files - 1.3.0.12):** Start Date/Time 16-9-16 11:01:42, Duration 00:00:03, Status Success.
- Log Section:** Includes a 'Download Log' link and a list of messages such as:
 - CopyALM.properties:
 - init: [mkdir] Created dir: D:\ALM_DATA\env\MyProject\CREATE_ARCHIVE\create_archive\target\7\odi
 - executeCopy: [copy] Copying 193 files to D:\ALM_DATA\env\MyProject\CREATE_ARCHIVE\create_archive\target\7\odi
 - Multiple [copy] messages for copying files from various source paths to target directories.

11. The Build has been successful. The next step will be to deliver the Build to the TEST environment.

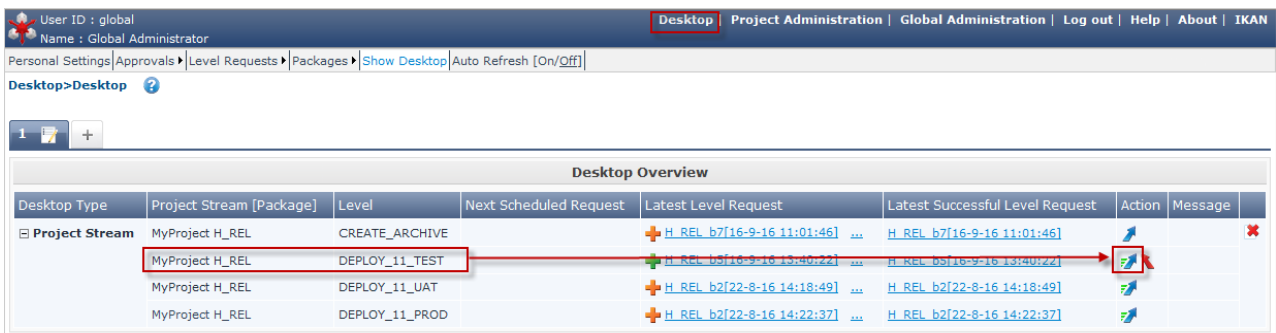
Creating the Deliver Level Request(s)

Now, let's deliver the build result to the TEST environment.






Just like when we created the Build Level Request, we create the Deploy Level Request via the Desktop.

Note: Afterwards, using the same procedure, you can create the Level Requests to deliver the results to the ACCEPTANCE and, finally, to the PRODUCTION environment.

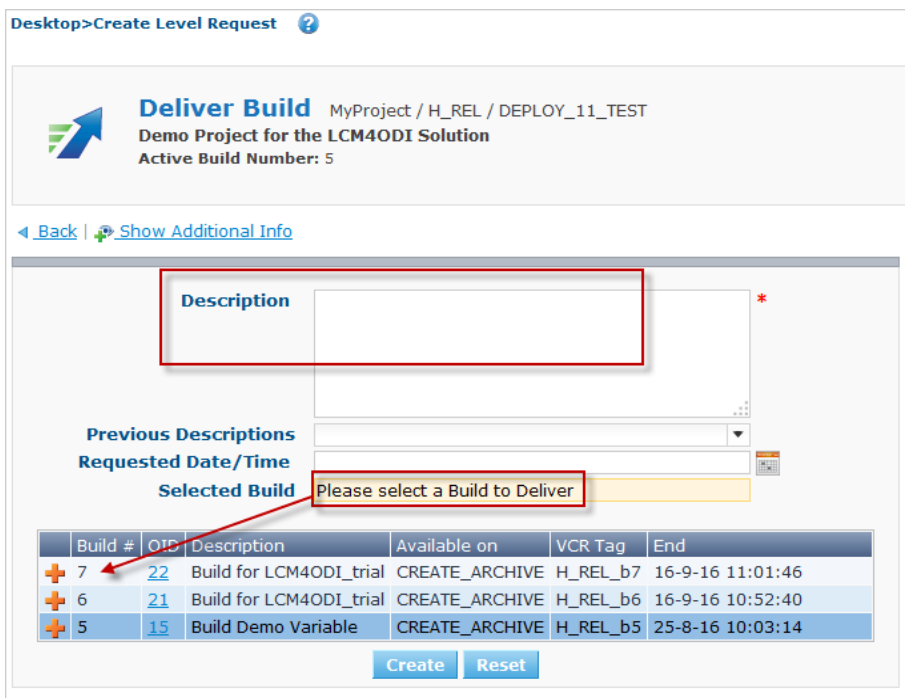
1. On the main menu, select *Desktop*.
2. Select the  *Deliver* icon in the *Action* Column of the TEST Level to create the Deliver Level Request.



The screenshot shows the 'Desktop Overview' table with the following data:

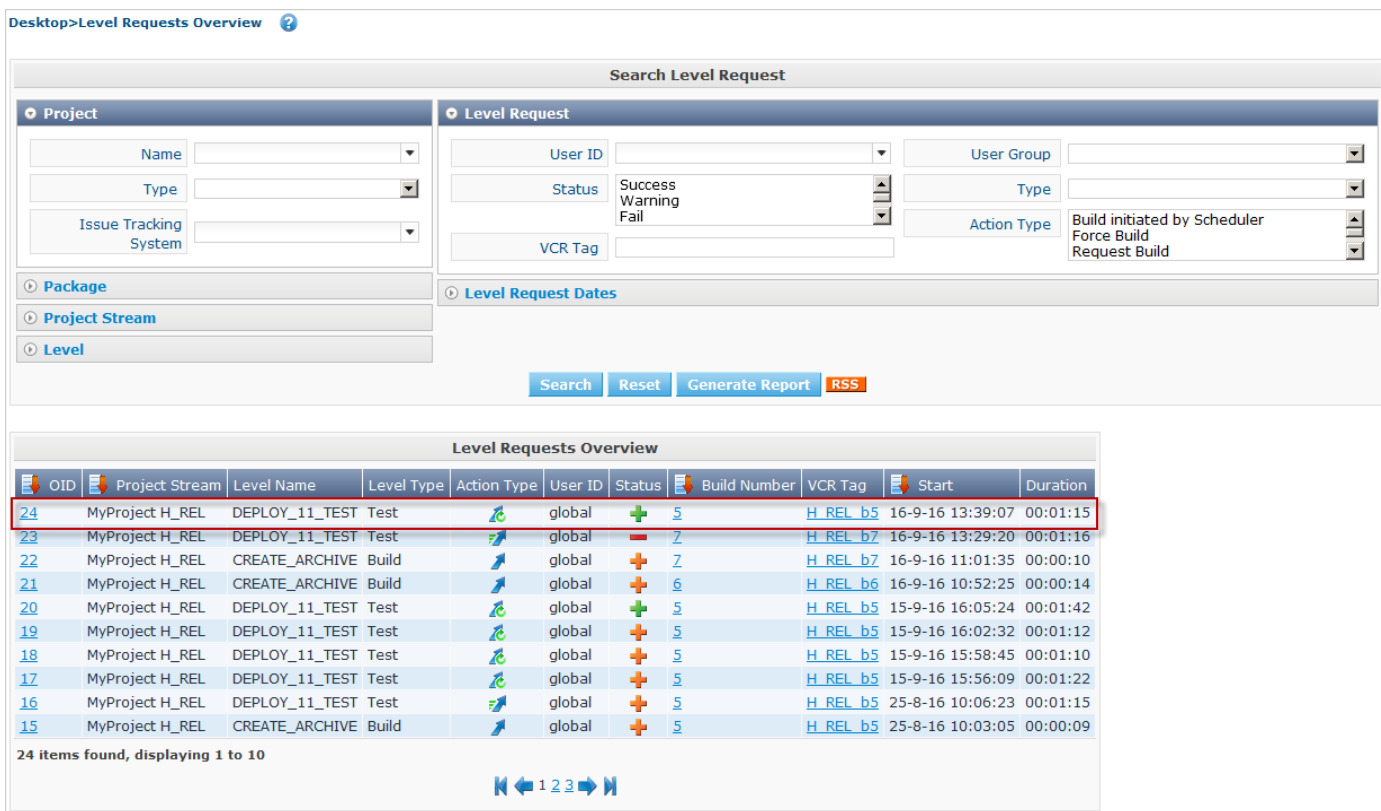
Desktop Type	Project Stream [Package]	Level	Next Scheduled Request	Latest Level Request	Latest Successful Level Request	Action	Message
Project Stream	MyProject H_REL	CREATE_ARCHIVE		H_REL_b7f16-9-16 11:01:46 ...	H_REL_b7f16-9-16 11:01:46]		
	MyProject H_REL	DEPLOY_11_TEST		H_REL_b2f22-8-16 14:22:37 ...	H_REL_b2f22-8-16 14:22:37]		
	MyProject H_REL	DEPLOY_11_UAT		H_REL_b2f22-8-16 14:18:49 ...	H_REL_b2f22-8-16 14:18:49]		
	MyProject H_REL	DEPLOY_11_PROD		H_REL_b2f22-8-16 14:22:37 ...	H_REL_b2f22-8-16 14:22:37]		

The *Create Level Request* screen will be displayed.



3. Provide a meaningful description, select the Build that was created earlier on our Build Level, and click the *Create* button.

You are forwarded to the *Level Requests Overview* screen.



- Click the *OID* number in front of the Project Stream.
You are forwarded to the *Level Requests Detail* screen.
The Detailed Overview for the Deploy Level Request is similar to the one for the Build Level Request.

Note: The execution of the Level Request may take several minutes dependent on the number of files that need to be processed. You can use the *Refresh* button to refresh the window.

Desktop>Level Request Detail ?

Success [MyProject / H_REL / DEPLOY_11_TEST / Build# 5](#)
24: BUID Demo Variable
Requested by: global on: 16-9-16 13:39:06

Summary Phase Logs Results Approvals Issues Sources Modifications Dependencies

[Back](#) [Refresh](#) [Build History](#)

Actions

No actions available

Info

Build Number 5
VCR Tag H_REL_b5
Action Redeliver Build
Type Deploys of archived Build
Start 16-9-16 13:39:07
Duration 00:01:15
[Show more...](#)

Builds & Deploys

	OID	Environment	Machine	Start	Duration
+	17	deploy_11_test	ikandemo004v	16-9-16 13:39:07	00:01:15

- Select the *Phase Logs* tab page. The gray lines on the overview represent the different Phases, the white lines represent the Build or Deploy actions.

Desktop>Level Request Detail ?

Success [MyProject / H_REL / DEPLOY_11_TEST / Build# 5](#)
24: BUID Demo Variable
Requested by: global on: 16-9-16 13:39:06

Summary **Phase Logs** Results Approvals Issues Sources Modifications Dependencies

[Back](#) [Refresh](#) [Build History](#)

Phase Logs

> Level Parameters

Phase Name	Start Date/Time	Duration
> + Retrieve Code	16-9-16 13:39:07	< 1 sec.
> + Build	16-9-16 13:39:07	< 1 sec.
> + Tag Code	16-9-16 13:39:07	< 1 sec.
> + Deploy	16-9-16 13:39:07	00:01:15
> + Deploy 17 on machine ikandemo004v	16-9-16 13:39:07	00:01:15
> + Cleanup Work Copy	16-9-16 13:40:22	< 1 sec.

- Click the Deploy name (in our example: *Deploy 17 on machine ikandemo004v*) to expand the information panel displaying the Phase details.

Deploy 17 on machine ikandemo004v		
OID 17	Start Date/Time	16-9-16 13:39:07
Environment deploy_11_test	Duration	00:01:15
Machine ikandemo004v	Status	Success
Deploy Parameters		
Transport Build Result	16-9-16 13:39:07	< 1 sec.
Decompress Build Result	16-9-16 13:39:07	< 1 sec.
Restore Oracle DI 11 Objects	16-9-16 13:39:07	00:01:13
Cleanup Build Result	16-9-16 13:40:21	< 1 sec.

The most important Phase is the *Restore Oracle DI Objects* Phase log where we can find the result of the *restoreODI.xml* script executed by Ant.

Click the *Restore Oracle DI Objects* Phase to open its *Phase Log*.

Phase Restore Oracle DI 11 Objects - 1.3.0.12 Duration 00:01:13
Start Date/Time 16-9-16 13:39:07 Status Success

Phase Parameters

Message

Log

[Download Log](#)

```

init:
[mkdir] Created dir: D:\ALM_DATA\env\MyProject\DEPLOY_11_TEST\deploy_11_test\source\17\com.ikanalm.phases.database.odi.restoreODI11Objects_2\log4j
[copy] Copying 1 file to D:\ALM_DATA\env\MyProject\DEPLOY_11_TEST\deploy_11_test\source\17\com.ikanalm.phases.database.odi.restoreODI11Objects_2\log4j
restoreODI11:
[echo] INFO: odi.action=restoreAllObjects
create.vcr4odi.configFile:
[mkdir] Created dir: D:\ALM_DATA\env\MyProject\DEPLOY_11_TEST\deploy_11_test\source\17\com.ikanalm.phases.database.odi.restoreODI11Objects_2\config
[copy] Copying 1 file to D:\ALM_DATA\env\MyProject\DEPLOY_11_TEST\deploy_11_test\source\17\com.ikanalm.phases.database.odi.restoreODI11Objects_2\config
applyPermission:
validate.vcr4odi.configFile:
restore.all.objects:
restore.action:
[java] sep 16, 2016 1:39:12 PM org.eclipse.persistence.default
[java] INFO: EclipseLink, version: Eclipse Persistence Services - 2.3.1.v20111018-r10243
[java] sep 16, 2016 1:39:12 PM org.eclipse.persistence.default
[java] INFO: master-session login successful
[java] sep 16, 2016 1:39:12 PM org.eclipse.persistence.default
[java] INFO: work-session login successful
[java] sep 16, 2016 1:39:12 PM org.eclipse.persistence.default

```

- The next steps will be to deliver the results to the ACCEPTANCE and/or PRODUCTION environments.

The procedure for doing this is exactly the same as the one described for delivering the results to the TEST environment.

- You have reached the end of this *Getting Started Guide*.

You have learned to perform the following basic actions:

- define a Subversion repository,
- create a Project,
- set up the different Build and Test Levels within a Life Cycle,
- create different Environments,

- set up the different Build and Test Levels within a Life Cycle,
- import the Solutions Phases and get them to work by adapting the different parameters,
- and, finally, create Build and Deploy Level Requests and follow up on the results.

If you still have unresolved questions or comments, do not hesitate to contact us at info@redbridgesoftware.com.

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

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

All other trademarks are property of their respective owners.